Restoration of blurred noisy images

Anastasia Belozerova
Ekaterina Chuikova
Daria Fokina
Fedor Streltsov

Skolkovo Institute of Science and Technology

Courses: Optimization Methods, NLA

Project Proposal

Task

Given the observed blurry noisy image and blur matrix restore the initial image

Applications

- Micro science
- Security
- Photography industry
- Computer vision

Related Works

- A. Danielyan, V. Katkovnik, K. Egiazarian, BM3D frames and variational image deblurring, 2011.
- N. P. Galatsanos, A.K. Katsaggelos, Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation, 1992.
- Lu Yuan, Jian Sun, Long Quan, Heung-Yeung Shum, Progressive Inter-scale and Intra-scale Non-blind Image Deconvolution.

Problem Statement

Notation

Let $z : 3 \times N^2$ be vector-reshaped observed image,
A - blur matrix, σ - derivation of additional noise,
$y : 3 \times N^2$ - initial image we are trying to restore,
$p(y)$ - penalty ('l1' and 'l2' are considered), τ - penalty parameter.

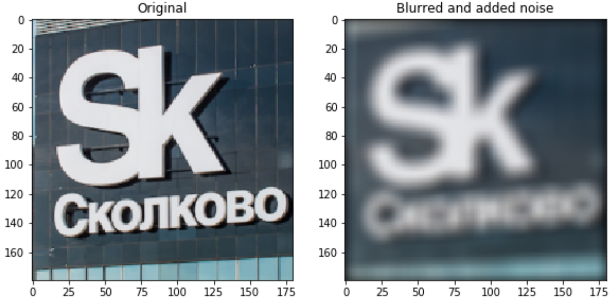
Formulation

$$\hat{y} = \arg\min_{y \in \mathbb{R}^{n^2}} \left(\frac{1}{2\sigma} ||z - Ay||_2^2 + \tau p(y) \right), \tag{1}$$

$$0 \leq \hat{y}_{ij} \leq 1, \forall i, j \tag{2}$$

Preparing Data

- Google photo of Skolkovo
- Take gaussian filter
- Create kernel
- Reduce convolution to matvec operation
- Add some noise



Implemented Methods

- cvxpy, lstsq, gurobi
- Projected Gradient Descent, Fast Gradient
- Proxi-Gradient
- Newton method

Variations

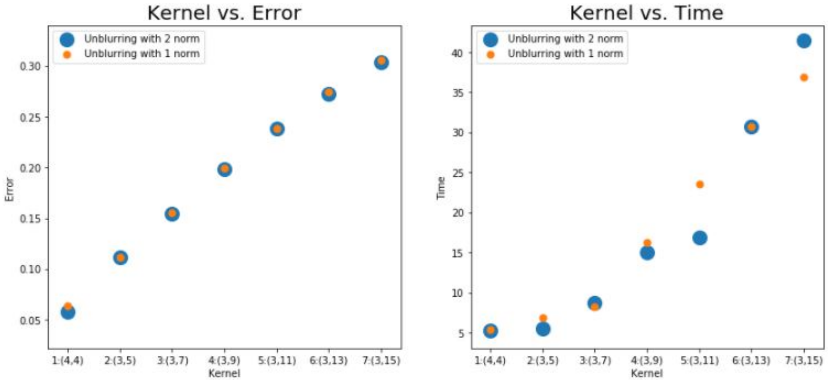
- Regularizations
- Kernels
- Initializations

CVXPY. Various Kernels

CVXPY. Various Kernels

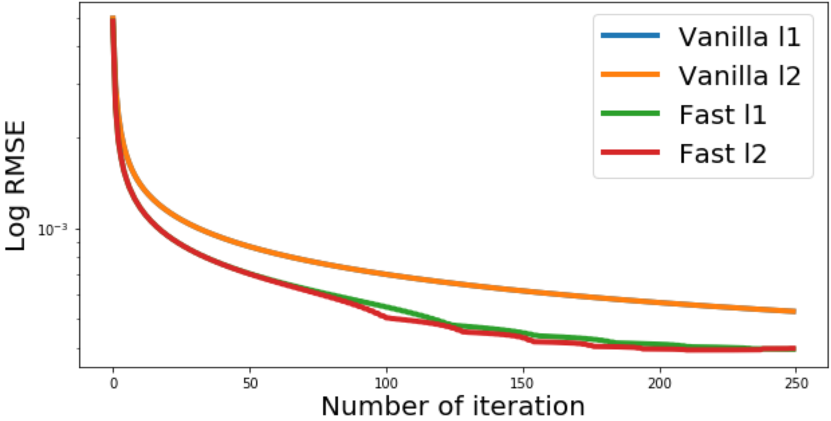
CVXPY. Various Kernels

CVXPY. Comparison



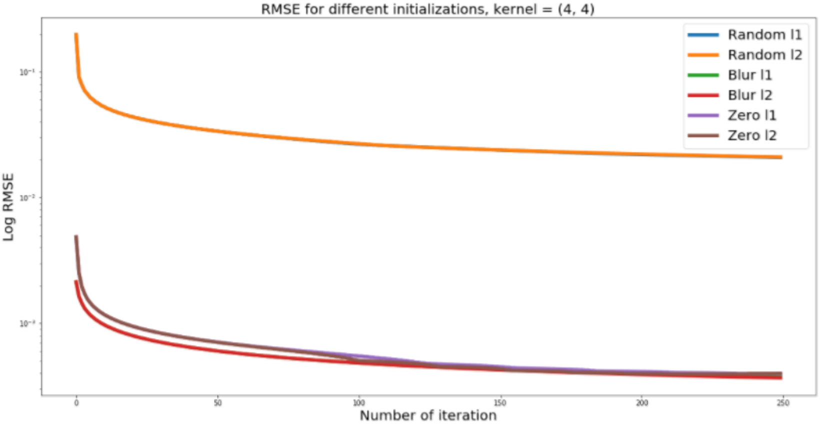
.

Gradient Descent. Vanilla and Fast



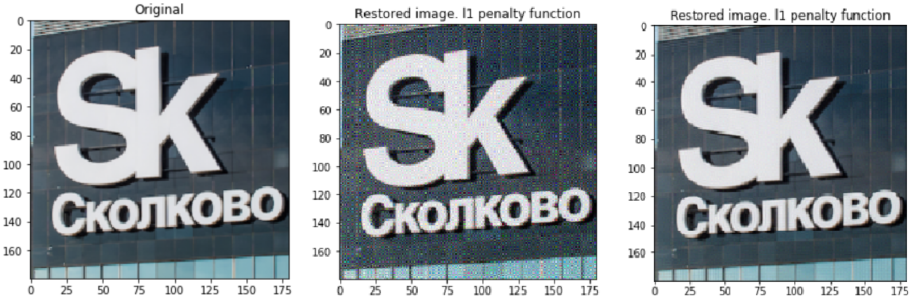
.

Fast Gradient. Different initializations



.

Fast Gradient. Different initializations

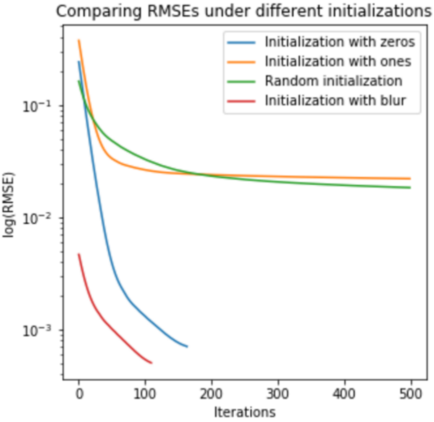


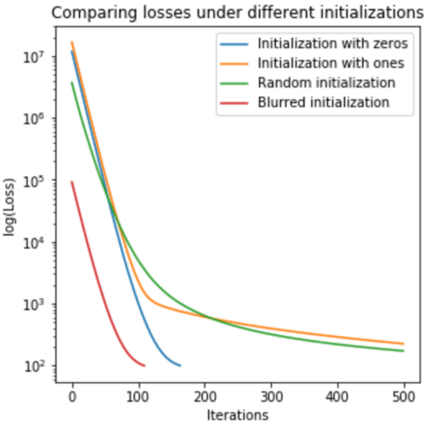
Comparison of two kernel's results

Proxi Gradient. Kernel = (4, 4)

Proxi Gradient. Kernel = (3, 15)

Newton Method. Different initializations



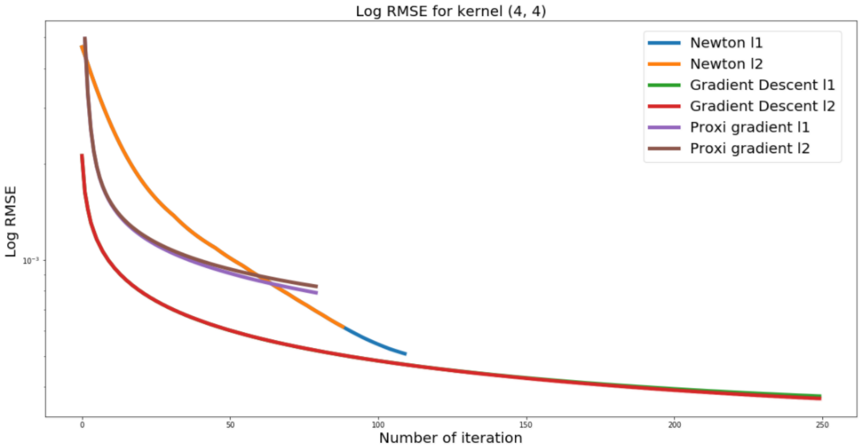


.

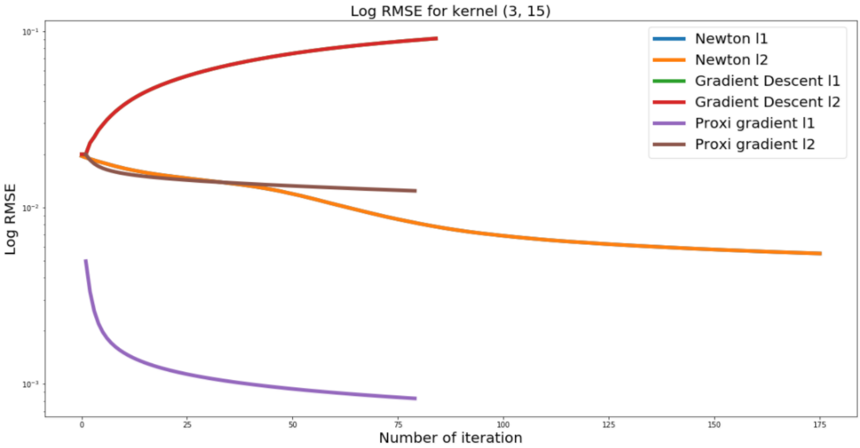
Newton Method. Kernel = (4, 4)

Newton Method. Kernel = (3, 15)

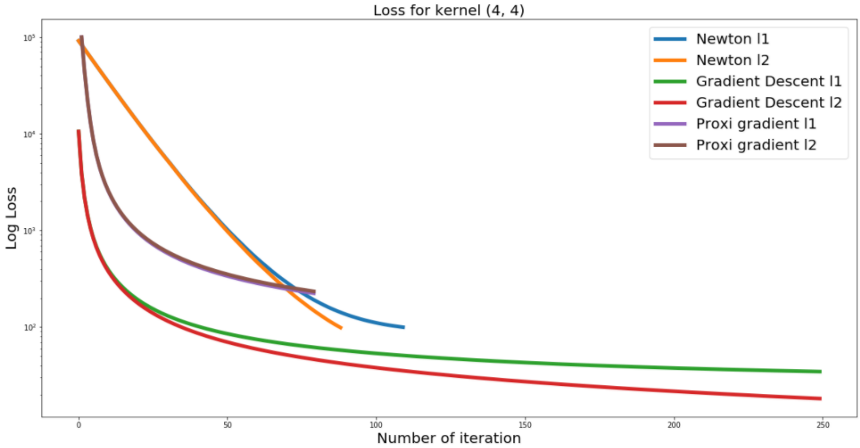
RMSE Rate. Kernel = (4, 4)



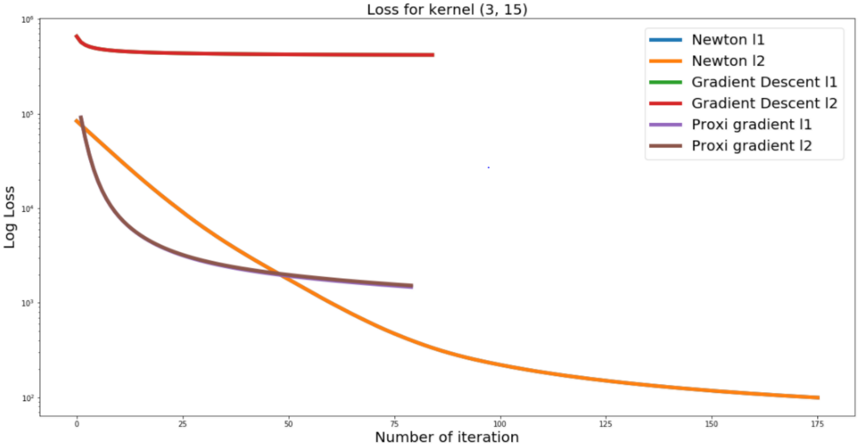
RMSE Rate. Kernel = (3, 15)



Loss Rate, Kernel = (4, 4)



Loss Rate, Kernel = (3, 15)



Final comparison

	Kernel(4, 4)			
	RMSE, l1	RMSE, l2	Loss, l1	Loss, l2
CVXPY	0.063	0.058	48.094	5.302
Fast Gradient	0.0007	0.0007	159.129	143.742
Proxi Gradient	0.0004	0.0005	88.271	94.285
Newton method	0.0005	0.0006	99.680	98.794

	Kernel(3, 15)			
	RMSE, l1	RMSE, l2	Loss, l1	Loss, l2
CVXPY	0.305	0.304	2392	1906
Fast Gradient	0.126	0.126	412988	412949
Proxi Gradient	0.008	0.009	373.271	510.558
Newton method	0.005	0.005	99.957	99.654

Conclusions

- Problem formulated
- Different approaches for solving were studied
- Data prepared
- Software developed, variations considered
- Results compared
- Team work appreciated

Our Team

Ekaterina Chuikova
data preparing
cvxpy

Daria Fokina
data preparing
proxi gradient

Anastasia Belozerova
projected and fast
gradient
presentation

Fedor Streltsov
Newton method