LIGHT FIELD RECOVERY USING CONVOLUTIONAL SPARSE CODING

A Project Report

submitted by

SUSMITHA A V

in partial fulfilment of requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

JUNE 2017

THESIS CERTIFICATE

This is to certify that the thesis titled **Light Field Recovery Using Convolutional Sparse Coding**, submitted by **Susmitha A V**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by her under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Kaushik Mitra
Project Guide
Dept. of Electrical Engineering
IIT Madras, 600 036

Place: Chennai

Date: 2nd June 2017

ACKNOWLEDGEMENTS

I would like to thank Prof. Kaushik Mitra for his wonderful supervision. I am very grateful to Anil Kumar Vadathya for his inspiration and guidance. Thank you Chinni Chaitanya for all the support. Thanks to all those who helped me to complete this project.

ABSTRACT

KEYWORDS: Convolutional Sparse Coding; Light Field; ADMM

Light field imaging is extensively used in many areas giving state of the art results. Most existing approaches either multiplex a low-resolution light field into 2D sensor image or require multiple photographs to be taken for acquiring a high resolution image. A few approaches also include traditional Sparse Coding to model the light field. A slight variation of Sparse Coding which uses convolutions to model the image is called Convolutional Sparse Coding (CSC). In this project, we extend the idea of CSC to light fields. The filters/kernels learned are used to reconstruct the light field views and view interpolation.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Conventional cameras don't capture most of the information about light distribution entering the camera, it only captures the spatial variation of the scene. But, Light Field Imaging captures both angular and spatial information. Higher Dimensionality feature of light field is extensively used in many areas now-a-days, giving state-of-art results. Light field Imaging offers post capture refocus of the scene, perspective shifts, recovering depth information. there are many existing approaches to learn and reconstruct light fields using Sparse Coding.

Sparse Coding Algorithm is a representation learning method which aims at finding sparse representation of input data in form of linear combination of basis elements. Over-complete dictionaries $(K > n)$ allow multiple representations of the signal improving the sparsity.Using an over-complete dictionary, $D \in \mathbb{R}^{n \times K}$, a signal $y \in \mathbb{R}^n$ can be represented by sparse linear combination of these atoms,The representation of y may either be exact $y = Dx$ or approximate $y \approx Dx$ satisfying the $\|y - Dx\|_p \leq \epsilon$ and sparsity constraints.The sparse vector x contains the representation coefficients of the signal, the sparsity of the signal x depends on the number of non zero coefficients in vector x, less the number of non-zero components more the sparsity.

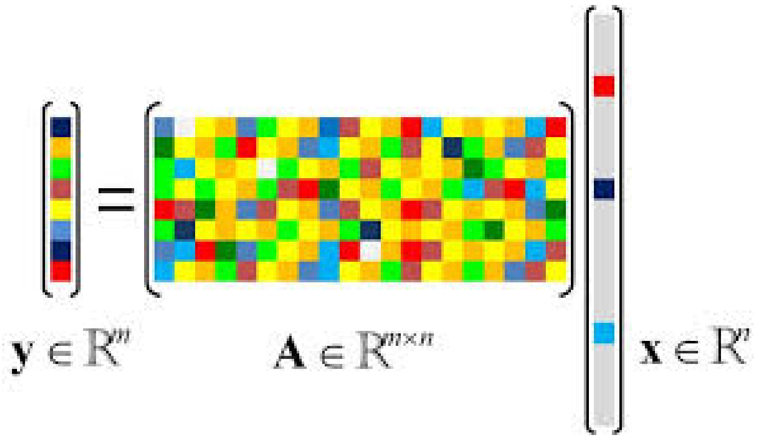
$\mathbf{y} \in \mathbb{R}^m$
$\mathbf{A} \in \mathbb{R}^{m \times n}$
$\mathbf{x} \in \mathbb{R}^n$

Figure 1.1: Sparse coding using Dictionary; Source:home.iitk.ac.in

However, the traditional sparse coding technique has a fundamental drawback it assumes the input signals $y \in \mathbb{R}^n$ are independent of one another.In general, natural images have repetitive features, this leads to many basis elements.This drawback is solved by convolutional sparse coding using convolution operation to model shift invariance property of the images. CSC takes input vectors unlike traditional sparse coding which learns inputs as patches, thus CSC is better than sparse coding in handling inputs during training and reconstruction.The optimization problem is considerably difficult when compared Sparse dictionary learning, since the signal in Dictionary learning was linear combination of dictionary atoms but CSC is the convolution of kernels with sparse code, where code update is higher dimensional that sparse vector of Dictionary learning and computationally expensive. Here in CSC the size of sparse code and input signal are same, but in dictionary learning the size of sparse vector can be different from that of input signal which leads to more sparse solution.

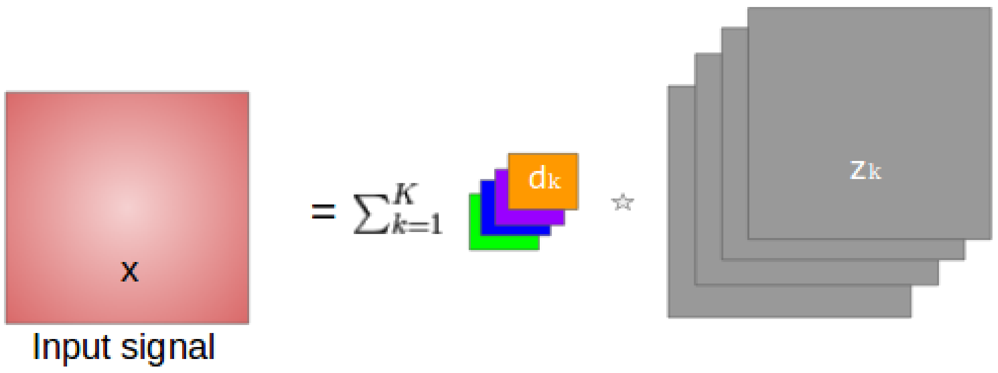

Figure 1.2: Describing Convolution between kernels and sparse code resulting modelling the input signal in CSC, x is the input signal, d_k,z_k are the kth kernel and sparse code.

CHAPTER 2

Back Ground

2.1 LightField

Light field views are array of images can be captured using a simple camera positioned in different consecutive view points. The position of each pixel in light can be described by plenoptic function (u, v, x, y). The angular and spatial dimensions of light field are (u, v), (x, y) repectively. Figure 2.1 is an example of lightfield

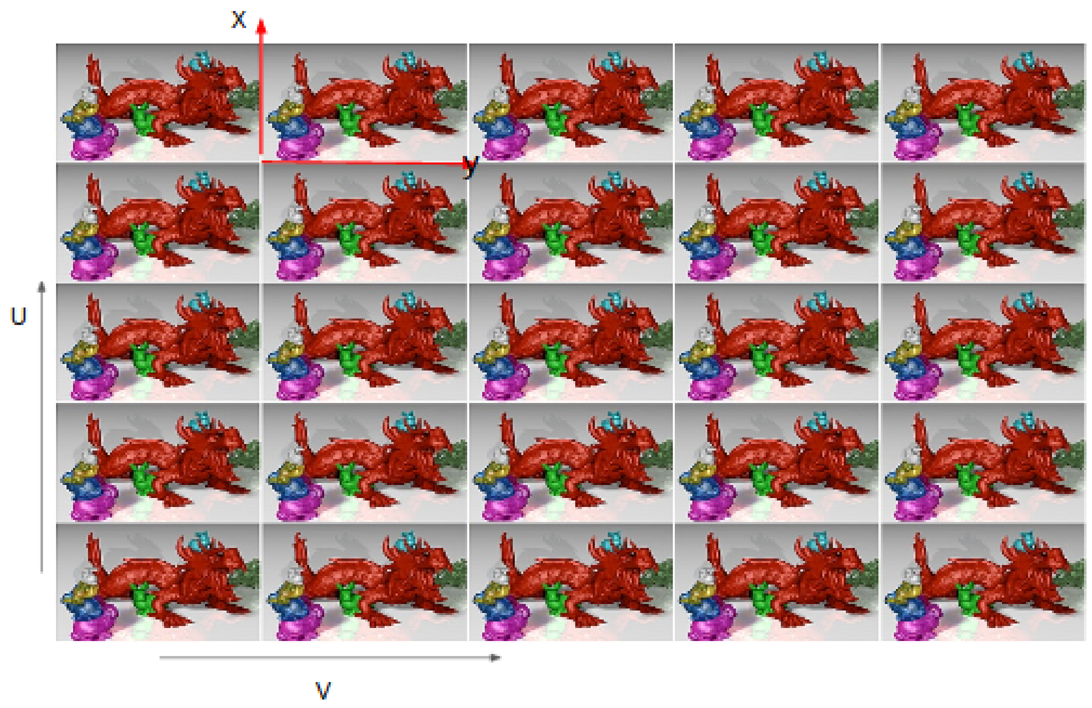

Figure 2.1: Light Field array of views depicting angular and spatial dimensions

2.2 Convolutional Sparse Coding

Convolutional sparse coding models local interactions of the signal using convolution operator on sparse vector. It is spatially invariant which is most relaxing feature of traditional sparse coding techniques. The kernels are constant which models the edges in the signal and relevant image features. The sparse code is decided by the weightage of particular kernel in the input signal. Finally input signal is modelled by the sum of the convolution of kernels with the corresponding sparse codes. The sparse constraint is also added to objective function, to minimize sparsity of the code. Objective function of convolutional sparse coding can be modelled as equation 2.1

$$\underset{d,z}{argmin} \left\| x - \sum_{k=1}^{K} d_k * z_k \right\|_2^2 + \beta \sum_{k=1}^{K} \|z_k\|_1 \tag{2.1}$$

$$\text{subject to } \|d_k\|_2^2 \leq 1 \quad \forall k \in \{1, 2...K\}$$

Equation 2.1 : x is the signal, d_k is the kth kernel and z_k is the k^{th} sparse code, $*$ is the convolution operator, K total number of kernels, β is the sparsity constant

Convolution in frequency domain is multiplication in time domain, therefore the objective function can also be written as equation 2.2

$$\underset{d,z}{argmin} \left\| \widehat{x} - \sum_{k=1}^{K} \widehat{d_k} \odot \widehat{z_k} \right\|_2^2 + \beta \sum_{k=1}^{K} \|z_k\|_1 \tag{2.2}$$

Equation 2.2 : expresses the computationally expensive convolution operations as more efficient multiplications in the Fourier domain. Here,$\widehat{}$denotes the frequency representation the component-wise product in frequency domain.

CHAPTER 3

Extension to light field

3.1 Objective function

The objective function for whole light field is taken as the sum of individual objective functions. Due to the following assumptions.

- In this case the kernels are 4D, the fourth dimension is number of views of light field, this add more complexity to optimization.
- Sparse code corresponding to each is taken constant across all the views, assuming that there will only be a small perspective shift in views, otherwise the whole image of each views is constant
- The small change in the views is learned by the kernels corresponding to the views

$$\underset{d^v,z}{argmin} \sum_{v=1}^{V} \left(\left\| x^v - \sum_{k=1}^{K} d_k^v * z_k \right\|_2^2 \right) + \beta \sum_{k=1}^{K} \|z_k\|_1 \tag{3.1}$$

$$subject\ to\ \|d_k^v\|_2^2 \leq 1$$

- V is the total number of light fields, d_k^v is the kernel corresponding to k^{th} kernel of v^{th} view of light field

3.2 Training

Training on light field is done iterative manner following the kernel update for each view and sparse code update, till no more improvement in both types updates

3.2.1 Kernel update for each view

- The Kernel update for each view is independent of others, only dependant on the corresponding view and sparse code which is constant across all the views.
- The proximal function of kernel and view are calculated using kernels of respective view and sparse code in each iteration

$$\frac{1}{2}\left(\|Zd^v - xi_1^v\|_2^2\right) + \frac{\rho}{2}\|d - xi_2\|_1 \tag{3.2}$$

- xi_1 and xi_2 are quadratic and Projection proximal funtions from Flexi Heide et al, Z is the stacking of z_k along columns $Z = \mathcal{F}_2([z_1^T, ...z_k^T])$, d^v is the stacked matrix of kernels corresponding to each view v. ρ is a constant to regulate the sparsity of the code.

$$z_{opt} = \left(Z_j^\dagger Z_j + 2I\right)^{-1}\left(Z_j^\dagger \times xi_1 + xi_2\right)$$

- Each block Z_j is the sparse code corresponding to the j^{th} kernel, since Z is the toeplitz matrix explaining the convolution as element-wise multiplication of stacked matrices since each block is a diagonal matrix.
- The inverse can be computed efficiently from Woodbury formula.

$$z_{opt} = \left(\left(\frac{1}{2}\right)I - \frac{1}{2}Z_j^\dagger\left(2I + Z_jZ_j^\dagger\right)^{-1}Z_j\right)\left(Z_j^\dagger \times xi_1 + xi_2\right) \tag{3.3}$$

- z_{opt} is given equation 3.3, where $\dagger$ is the conjugate transpose operator.

3.2.2 Code update

- Sparse code is different from that of single image code because here we take constant code across all the views.

$$\frac{1}{2}\sum_{v=1}^{V}\left(\|D^v z - xi_1^v\|_2^2\right) + \frac{\rho}{2}\|z - xi_2\|_1 \tag{3.4}$$

- where $D^v = \begin{bmatrix} D_1^k & \dots & D_k^v \end{bmatrix}$ is the concatenation of Toeplitz matrices each one representing a convolution with respective filter d_k^v

$$\left(D_j^\dagger D_j + 2I\right)^{-1}\left(D_j^\dagger \times xi_1 + xi_2\right)$$

- xi_1 and xi_2 are the Quadratic and shrinkage proximal operators from Flexi Heide et al

$$d_{opt} = \left(\left(\frac{1}{2}\right)I - \frac{1}{2}D_j^\dagger\left(2I + D_jD_j^\dagger\right)^{-1}D_j\right)\left(D_j^\dagger \times xi_1 + xi_2\right) \tag{3.5}$$

- where $D_j = \mathcal{F}_2\begin{bmatrix} d_j^1 & \dots & d_j^V \end{bmatrix}$

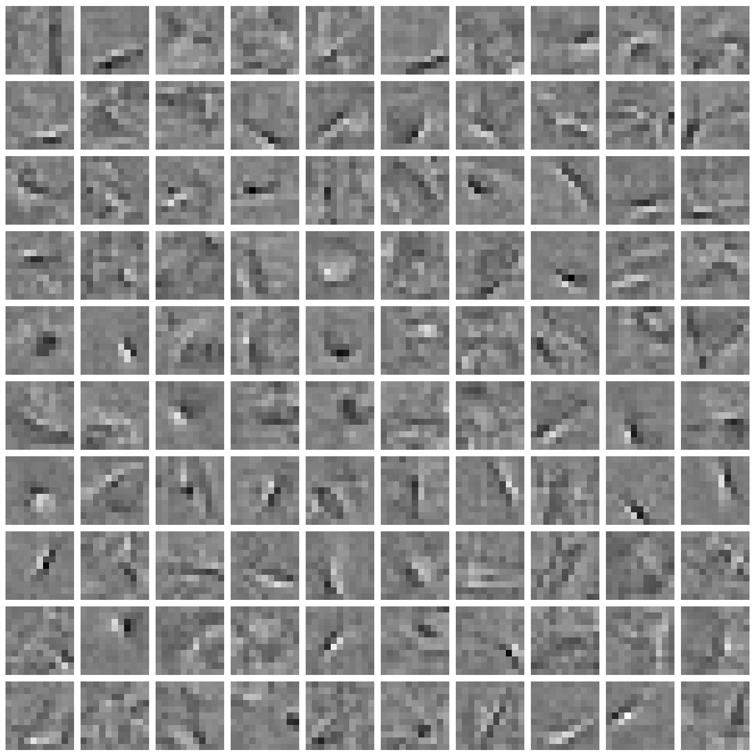

Figure 3.1: Kernel corresponding to one view of the light field

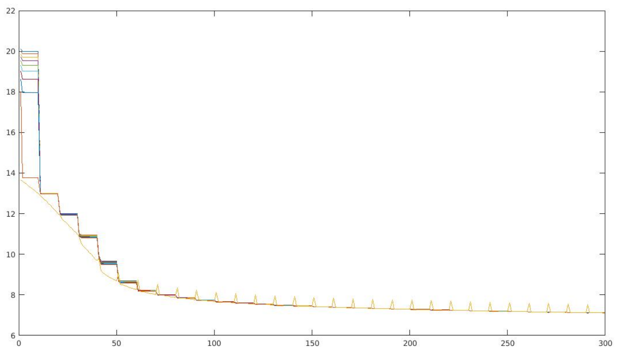

Figure 3.2: Plot covergence during training, $x - axis$: Number of iterations, $y - axis$: Objective function value

CHAPTER 4

Results

4.1 Denoising

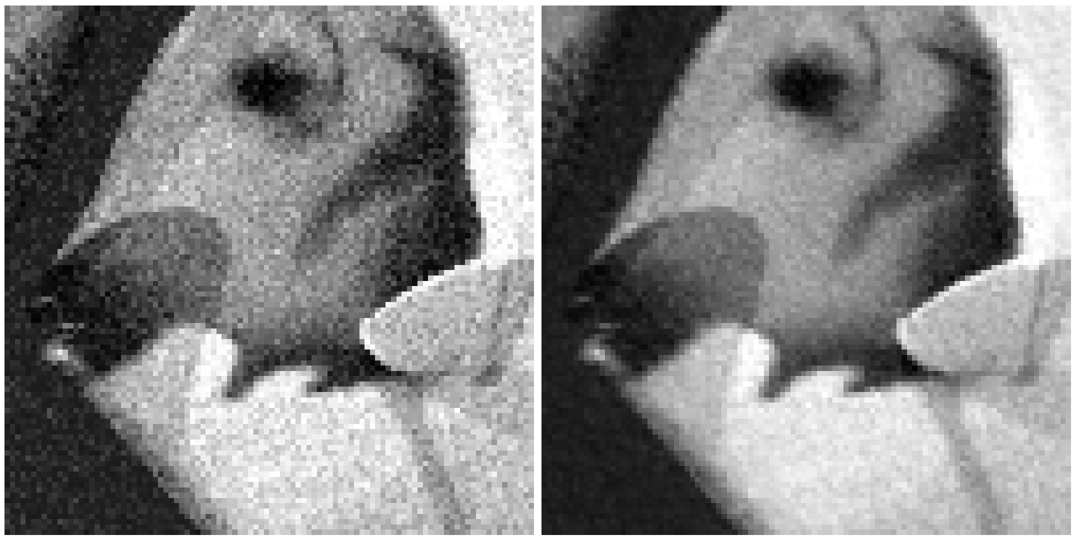
Figure 4.1
Denoising Light Field views
Noisy image PSNR: 23.9915 dB
Denoised Image PSNR: 28.8708 dB

4.2 View interpolation

Missing views are reconstructed from the input views, such as four corner views of the light field. The input views should have required information on perspective shift of edges.

4.2.1 Four corner views

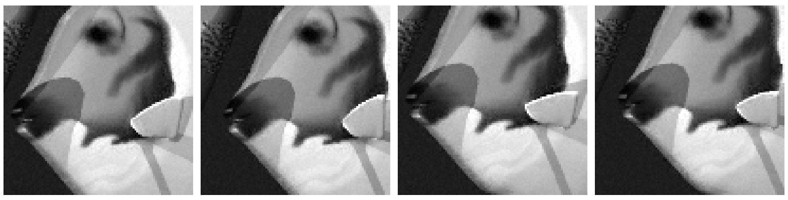

Figure 4.2: input views [(1,1),(1,5),(5,1),(5,5)

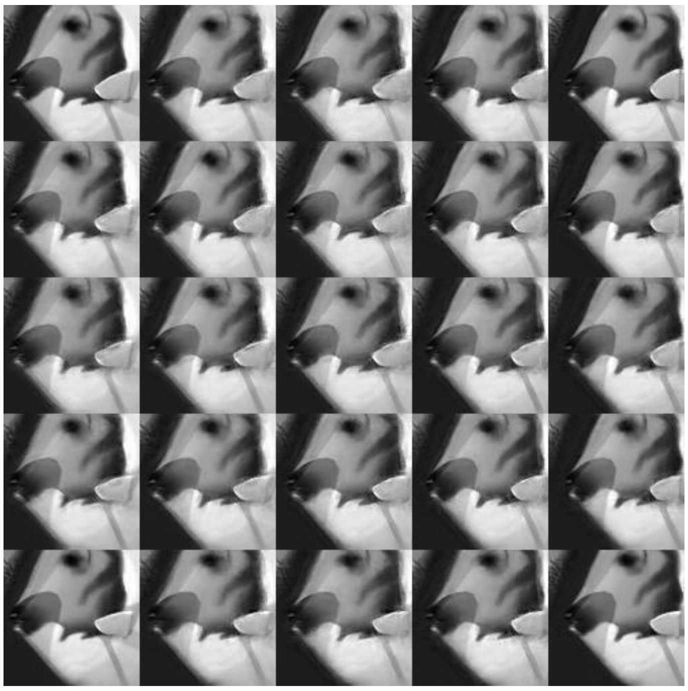

Figure 4.3
Output views from four view-interpolation
PSNR(output views): 20.7160 dB
PSNR(input views): 25.80 dB

4.2.2 Two Corner views $[(1,1),(5,5)]$

The two corner views (1,25) are given as input views, The reconstruction is kind of overfitting to the input views, hence the interpolated views are lesser quality than the former case (four corner views).

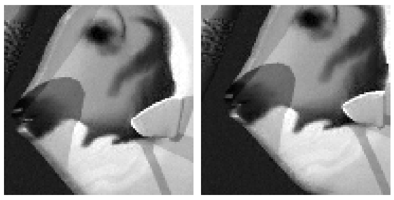

Figure 4.4: Input views $[(1,1),(5,5)]$

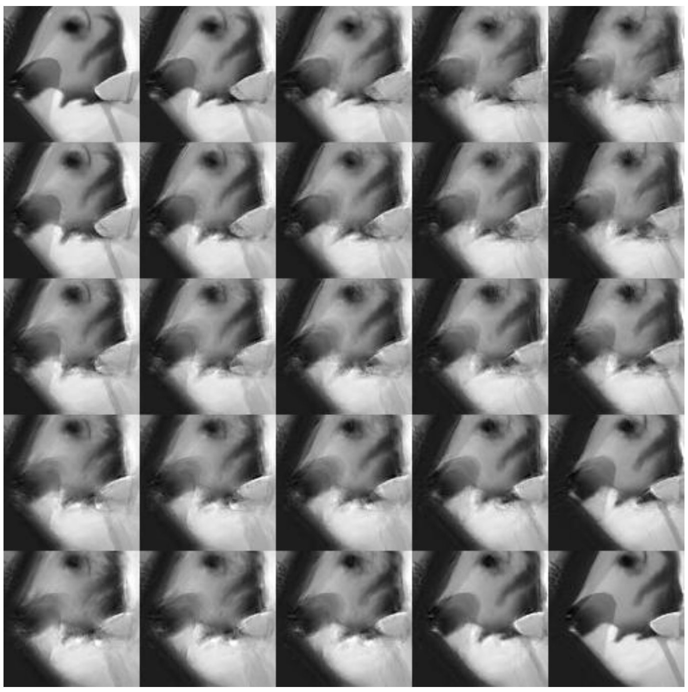

Figure 4.5
Output views from two-view interpolation
PSNR(output views): 20.00 dB
PSNR(input views): 22.46dB

CHAPTER 5

Conclusion

Convolutional Sparse coding is a powerful tool in learning image features, in this project it is used to learn Light field views.The implementation is an extension of Fast and Flexible Convolutional Sparse Coding Paper. The kernels learned were used to reconstruct views from noisy lightfield and View Interpolation. The results in de-noising are not as good as other approaches, but it can be improved by learning better Filter kernels. It can also be extended to reconstruction from Coded Image from Compressive light field paper.

REFERENCES

1. **Flexi Heide, W. H.** and **G. Wetzstein**, Fast and flexible convolutional sparse coding. *In Computer Vision and Pattern Recognition (CVPR)*. IEEE Conference on, 2015.

2. **Hilton Bristow, A. E.** and **S. Lucey**, Fast convolutional sparse coding. *In Computer Vision and Pattern Recognition (CVPR)*. IEEE Conference on, 2013.

3. **Stephen Boyd, E. C. B. P., Neal Parikh** and **J. Eckstein** (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, **3**(1), 1–122.