A two-component model for compressive light field reconstruction

Yuxin Hu
Stanford University
Stanford, CA
yuxinh@stanford.edu

Minda Deng
Stanford University
Stanford, CA
mindad@stanford.edu

Abstract

Light field photography, which records both spatial and angular information of a scene, embraces many novel applications. However, the intrinsic trade-off between spatial and angular resolution has limited its potential. In this work, we explored the structure of the light field data, and proposed a two-component model for its reconstruction inspired by compressed sensing, which is a promising tool to solve underdetermined problems. We applied this model to a view-combined camera, and successfully recovered the angular information while reserving high spatial resolution.

1. Introduction & Motivation

Our visual system can collect and process incredibly large amount of data and help us better perceive the world around us. The plenoptic function [1] serves as a simple model for all the information our visual systems deal with, including spatial, angular, wavelength and time dimensions. However, conventional cameras only capture images with spatial part of the information in the plenoptic function. To overcome this limitation, the idea of light field photography has been developed and recently been implemented and brought to the consumer market. Usually by using a microlens or camera array, light field photography records images while preserving both spatial and angular information of a scene. This technology has been on the focus of research and it opens up the possibilities for various applications. People now have been exploring and improving the light field cameras for digital refocusing, depth estimation, occlusion removal [2-4], etc.

While light field photography enables many novel applications, it suffers from an intrinsic tradeoff between angular resolution and spatial resolution. For a commercialized light field camera using microlens arrays, it has a raw sensor resolution comparable to that of a conventional camera. However, most of the pixels are now dedicated to capture the angular information. This unavoidably sacrifices spatial resolution and largely limits the applications of light field photography.

To overcome this intrinsic problem, people have been developing different physical and computational techniques. Linear optimization based super resolution algorithms can reconstruct images using images from different angles to achieve better spatial resolution [5]. Also dictionary based methods are applied in the context of light field imaging and reconstruction of images with better spatial resolution. Basic properties of light fields could be learned from the dictionary built from a large data base [6].

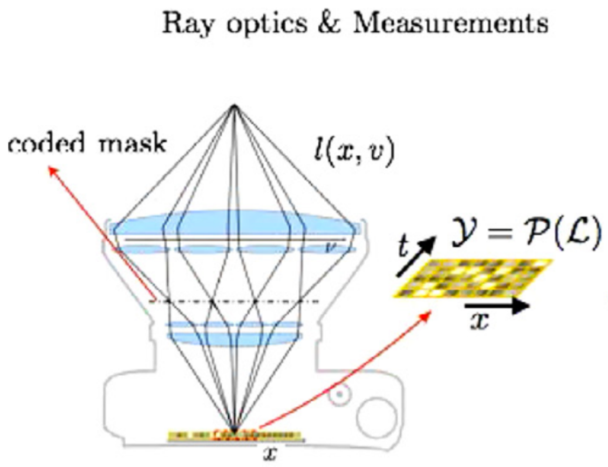
Ray optics & Measurements
coded mask
$l(x,v)$
v
t
$\mathcal{Y} = \mathcal{P}(\mathcal{L})$
x
x

Figure 1: Illustration of acquisition process of the light filed camera used in this work. Images from different views are weighted by the coded mask and then combined [7].

In this paper, we explored images reconstruction techniques based on a slightly different physical design of the light field camera [7]. Conventional light field cameras based on microlens arrays/camera arrays assign physically different pixels/cameras for capturing information coming from different angles (views) and then try to reconstruct from those images of slightly different views. The physical design that our algorithms based upon, however, operates in a different way. Instead of assigning physical sensing power for angular information, the camera design in ref [1] compresses angular information by collecting light from different angle altogether at the same time and then tries to extract spatial and angular information using compressive sensing techniques, which is similar to the design of single-pixel camera. This process is illustrated in Figure 1. The output from this light field camera design are 3D images with two spatial dimensions and one temporal dimension (frames). Then from those 3D images, optimization algorithms are used to recover angular information without sacrificing the spatial resolution. We explored different models, priors/constraints for the reconstruction and the detailed methods and results will be discussed in later sections.

2. Related work

Recently, people have been researching on various light field camera designs and ways to reconstruct images with better spatial resolution. One popular and well exploited solution to improving resolution in light field imaging is to use superresolution (SR) algorithms. This technique can extrapolate additional information for better resolution from the existing data. For instance, Bishop et. al exploited the fact that light fields can usually be modeled with limited complexity and in their work, they used the bidirectional reflectance distribution function (BRDF) to describe the properties of the light fields [5]. The light field reconstruction problem is then formulated into a Bayesian framework that allows them to exploit Lambertian reflectance priors and super resolve the light field images. However, those linear optimization based SR methods could not be applied to compressive light field data sets.

Another approach of achieving high resolution light field image reconstruction that is also compatible with compressive data acquisition is through the use of a learned dictionary [6]. In spirit this approach resembles the idea of machine learning. By learning from a large amount of light field images the dictionary is trained to be able to recover images from different views by exploiting prelearned information of basic properties of light fields contained in the dictionary. In the work of Marwah et al., they physically and computationally implemented this idea and were able to obtain images with higher resolution from a single coded input image. The challenge of this technique is, however, that the learning phase of the dictionary is very slow and computationally expensive.

People have also explored various methods to use optimization techniques to recover high resolution images from a compressive data acquisition setting. Kamal et al. proposed using the camera design shown in Figure 1, in which images from different views pass through randomly generated masks and then are combined into one single image with angular information hided in it [7]. Recovering angular information from those coded images become a compressive sensing problem when the number of time frames is smaller than the number of compressed views. Instead of using the prelearned dictionary approach which is expensive, they also proposed a reconstruction method which assumes that compressive light fields can be decomposed into a low-rank and a sparse component, they use parallel proximal algorithm to iteratively solve this problem and reconstruct images.

In our work, we follow a similar framework as Kamal et al. to address the compressive light field reconstruction problem. We developed a more generalized two-component model, used ADMM to solve this problem and tested various combinations of priors to approach this problem.

3. Methods

3.1. Structure of light field data

As introduced before, light field cameras capture light field information of a scene by recording images from different angles. One example of light field images from the homework is shown below,

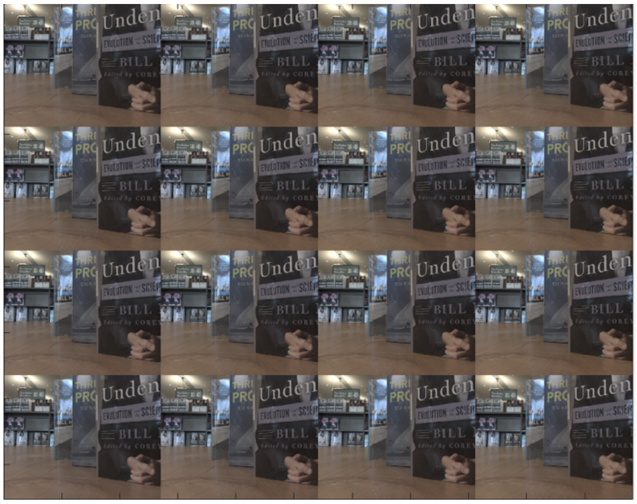

Figure 2: Example of light filed images with 16 views.

For simplicity, we only take 4 x 4 = 16 views as an example. All those images are about the same, except that objects out of focus are slightly shifted, and the amount of shift is related to the viewing angle. If we take the averaged image, we can get a blurry image as in Figure 3.

Subtracting this averaged image from all views, the residual is shown below (amplified by 3 times otherwise it would be all dark), which is already very sparse. By this simple subtraction operation, we can find that images from different views share a lot of common information, which can be even approximated by one single image relatively well.

We can also reshape the image into one vector and concatenate those vectors from all views into one matrix.

Singular values of this matrix are plotted in Figure 6, from which we can get a similar conclusion: most of information of those different view images are saved into first several components.

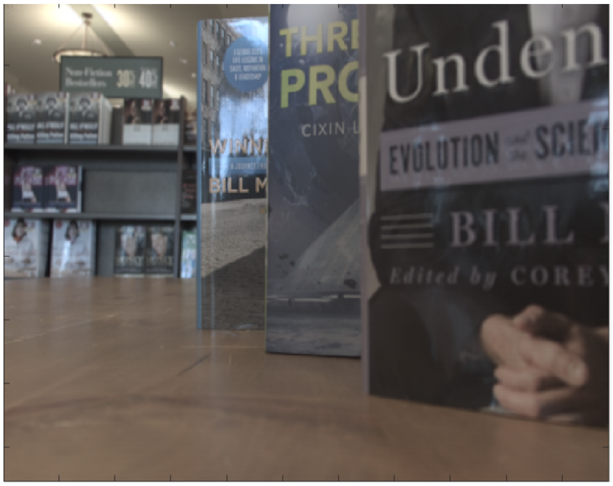

Figure 3: Averaged image which is blurry.

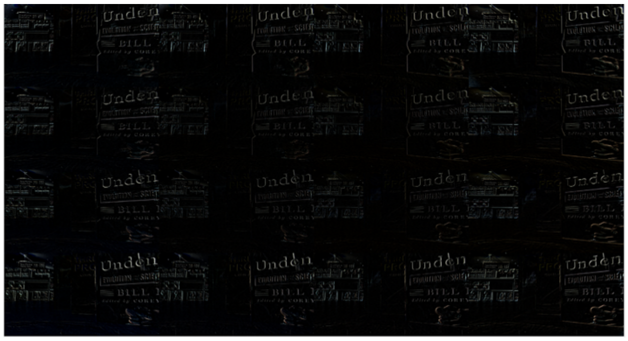

Figure 4: Residual images (3 times) after removing the averaged components.

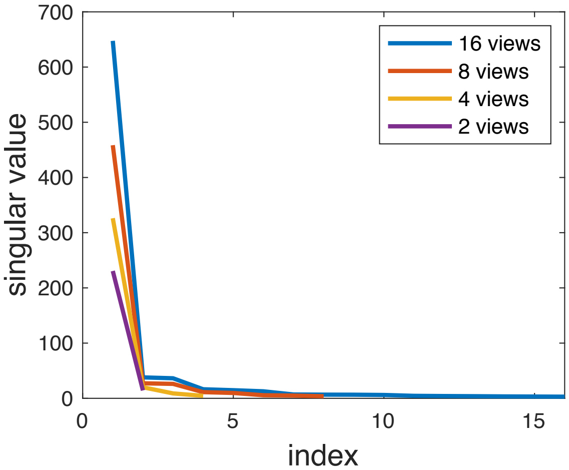

Figure 5: Singular values of the spatial-view matrices constructed with 16, 8, 4, 2 views.

We can also find that even with only 2 views, there is still one denominate component. Based on these observations, the assumption of our model is that

1) the common information along different views lies in some low-dimension space, which can be represented as some low-rank structure.

2) the difference can be captured by some additoinal sparse components.

Similar to other compressed sensing methods [8], we also tried transforming the data into other domains to gain better sparsity, which can be visually seen as the following (first column shows the image in spatial and other domains, the second column shows its difference from the averaged image and corresponding transforms):

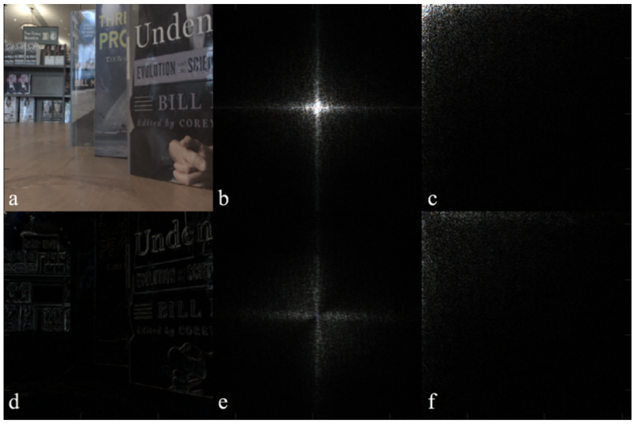

Figure 6: original image (a), residual image (d) and their Fourier transform (b and e), and discrete cosine transform (c and f).

3.2. Two-component model

By assuming the light filed data can be decomposed into two components, we propose our model as the following minimization problem:

$$\min_{X_1,X_2} \|M_0(M_1X_1+M_2X_2)-b\|_2^2 + \lambda_1 P_1(\Psi_1(X_1)) + \lambda_2 P_2(\Psi_2(X_2)) \quad (1)$$

Where X1 and X2 are the two components we want to estimate, b is the acquired view-combined image, M0 is a linear operator to combine different views based on the random mask, M1 and M2 are two linear operators on those two components, P1 and P2 represents the constraints, λ1 and λ2 are corresponding regularization parameters, Ψ1 and Ψ2 are two linear transformation operators.

This model can be viewed as a generalization of many models, and the following are some special cases:

1) if M1 and M2 are both an identity operator, then this is the well-known low rank + sparse model.

2) if either M1 or M2 is zero operator, then this model becomes the conventional one-component model, e.g. low-rank or sparse model.
3) if M1 is a duplication operator along the view dimension, and M2 is still an identity operator, then we are assuming different views share one common structure, and X2 has the view-specific information.

3.3. Algorithm and Implementation

Our model can be written as following

$$\min_{X_1,X_2} \|M_0(M_1X_1 + M_2X_2) - b\|_2^2 \quad (2)$$

$$+ \lambda_1 P_1(Z_1) + \lambda_2 P_2(Z_2) \quad (3)$$

$$s.t. \Psi_1(X_1) - Z_1 = 0 \quad (4)$$

$$\Psi_2(X_2) - Z_2 = 0 \quad (5)$$

Let

$$f(X_1, X_2) = \|M_0(M_1X_1 + M_2X_2) - b\|_2^2, \quad (6)$$

$$g_1 = \lambda_1 P_1,\ g_2 = \lambda_2 P_2, \quad (7)$$

$$A_1 = \Psi_1,\ A_2 = \Psi_2, \quad (8)$$

$$B_1 = -I,\ B_2 = -I,\ c_1 = c_2 = 0 \quad (9)$$

This problem is in the format of the problem in Appendix which can be optimized by Alternating direction method of multipliers (ADMM) [9].

The update rule can be derived as follows:

$$X_1^{k+1} = argmin_{X_1} \left\|M_0M_1X_1 + M_0M_2X_2^k - b\right\|_2^2 \quad (10)$$

$$+ \rho/2 \left\|\Psi_1(X_1) - Z_1^k + U_1^k\right\|_2^2 \quad (11)$$

$$X_2^{k+1} = argmin_{X_2} \left\|M_0M_1X_1^k + M_0M_2X_2 - b\right\|_2^2 \quad (12)$$

$$+ \rho/2 \left\|\Psi_2(X_2) - Z_2^k + U_2^k\right\|_2^2 \quad (13)$$

$$Z_1^{k+1} = prox_{\frac{\lambda_1}{\rho}P_1}(\Psi_1(X_1^{K+1}) + U_1^{k+1}) \quad (14)$$

$$Z_2^{k+1} = prox_{\frac{\lambda 2}{\rho}P_2}(\Psi_2(X_2^{K+1}) + U_2^{k+1}) \quad (15)$$

$$U_1^{k+1} = U_1^k + \Psi_1 X_1^{k+1} - Z_1^{k+1} - c_1 \quad (16)$$

$$U_2^{k+1} = U_2^k + \Psi_2 X_2^{k+1} - Z_2^{k+1} - c_2 \quad (17)$$

For X1 and X2 updates, the problem can be solved by conjugate gradient. Z1 and Z2 updates involves the proximal operators of the constraints. Many commonly used penalty terms have analytical solution, like for l1-norm penalty, its proximal operator is soft thresholding. For nuclear norm penalty, which is the closest convexity form of the rank constraint, the proximal operator is soft thresholding on the eigenvalues.

Our algorithm and some commonly used transformation and proximal operators are implemented in Matlab, and the code is available at https://drive.google.com/open?id=1XDZsROvJaLwXzfpkCJR6GRPxOVA9FVYf.

3.4. Priors and parameters

In this work, we mainly explored the low-rank and sparse models. For the prior on low rank components, we tried locally low-rank and globally low-rank. We are assuming those components are similar among different views, and to utilize those similarities, we construct spatial-view matrices, and apply rank constraints on those matrices. For locally low-rank [10], we take one small spatial block each time, and reshape this block into one vector, concatenate all those vectors into one matrix, and we are constraining on the sum of ranks of those matrices, which together cover the whole field of view. For globally low-rank, we only construct one such matrix with the whole image. For the sparse components, we applied some linear transforms first, and then applied the L1 penalty. We tried wavelet transform, Fourier transform, and discrete cosine transform.

We used 15 iterations for the ADMM algorithm, and we set the maximal iteration number as 20 for inner conjugate gradient optimization for X1 and X2 update.

3.5. Simulation

We take light filed data from The Stanford Light Field Archive (http://graphics.stanford.edu/data/LF/lfs.html), and Rectified and cropped images are used. The uniform random masks are generated using Matlab function “rand”. We use these masks to weight images from different views and combine them to simulate the data acquisition process. The combined data is given to the implemented algorithm to recover the original multi-view images.

4. Results

Following our two-component decomposition algorithm implemented using ADMM, we applied various priors/constraints to 3 data sets: ‘Rabbit’, ‘Beans’ and ‘Lego’. The priors/constraints we used include locally low-rank (LLR), globally low-rank (GLR), l1-wavelet, l1-discrete cosine transform (DCT) and l1-fast Fourier transform (FFT). We tested different combinations of priors on the two different components from the decomposition. We explored the parameter space and the reconstruction results are summarized in Table 1.

These results are from reconstruction of 8 views given 4 coded input images (number of frames is 4). From the reconstructed images and the original images we used to generate the coded input, we computed the PSNR and SSIM between each corresponding pair of the same view and averaged them over 8 views to generate the results shown in the table.

PSNR/SSIM	Prior Combinations							
Data Set		LLR+FFT	LLR+Wave	LLR+DCT	GLR+DCT	DCT+Wave	GLR+Wave	GLR+FFT
	Rabbit	25.0/0.88	28.9/0.91	24.9/0.88	24.0/0.85	30.2/0.92	24.8/0.87	23.9/0.85
	Beans	26.9/0.92	29.8/0.94	27.2/0.93	27.0/0.91	31.1/0.93	26.4/0.70	25.5/0.69
	Lego	21.9/0.74	22.8/0.76	21.2/0.72	21.7/0.74	23.4/0.75	22.8/0.77	21.6/0.74

Table 1. PSNR and SSIM summary.

From this table we see that, in general, the locally low rank (LLR) prior on the first component works slightly better than the globally low rank (GLR) prior. This might be because LLR divides the image into smaller blocks and enforce low rank constraints, and thus capturing more local details. For the sparse component, the wavelet transform seems to be able to capture most of the structures and yields the best results. We also note DCT could also be applied to the low rank component which yields decent results. The PSNR and SSIM values could be really different from data set to data set. For instance, the values for the 'Lego' data set are much lower than the other two, mainly because the 'Lego' has much more complex structures in the images.

The original 8 views used for simulation are shown in Figure 7 a). Part b) shows the 4 random mask coded images as input to our algorithm. They look like noisy averages of the 8 original views. Then part c) shows our reconstructed 8 images for the corresponding views. By comparing the recovered images to the original ones, we see that the angular information is successfully extracted. However, if we look at some of the details in the reconstructed images, we can still notice noisy regions due to imperfect decomposition.

Figure 8 shows the error plot as a function of number of iterations for the 'Rabbit' data set using the priors discussed above. Here the error is the data consistency term (first term in equation 1) which indicates how different the results is from the original images. This plot is typical among various prior combinations. It decreases and then usually saturates around 10 – 20 iterations. We note that, however, diminishing error term does not always suggest good convergence to the ground truth. Oftentimes, especially depending on the parameters in the model, the results converge to the simple average of the 8 views with some local variations.

5. Discussion

In this work, we proposed a two-component model, in which we assume the multi-view images can be approximated by two components, and these two components have different properties. It can be viewed as a generalization of the original low-rank + sparse model in [7]. Specifically, in this work, we are assuming all views share some common components, which is spatially smooth, or can be used to construct some low rank matrices. This assumption comes from the fact that images of different views are the same scene taken from different angles, and they share a lot of structures. In addition, each view has one sparse component to catch the view-specific information.

5.1. Priors and regularization parameters

Different priors and combinations of them are tested in this work. To capture the low rank structure, we constructed so-called spatial-view matrices, and added nuclear norm penalty term to enforce the low-rank structure. For the sparse term, we applied l1-regularization to its linear transform, since the sparsity can be usually better revealed after those transforms. In this work, we tried FFT, wavelet transform, and DCT, and l1-wavelet achieves highest PSNR and SSIM among these three. Currently, those linear transformations are applied only along spatial dimensions, not the angular dimension. The reason is that we are assuming the sparse term catches the view-specific information, and we don't want it to be view-related, though the linear transform along angular dimension may further increase the sparsity. Careful comparison are needed to validate this point in future work. Some other constraints, like low-rank matrix in the spatial domain, total variation can be also included into this model easily, and their performance remains to be explored.

How to choose the regularization parameter is always a problem for this kind of regularization problem. The problem is even more severe for this method since we have two regularizations parameters to be tuned. We did a grid search for the two regularization parameters, the choice of those parameters determines the allocation of the energy between the two components. When the parameter is too large, the view-by-view difference can be compromised, and when it is too small, the results can be very noisy.

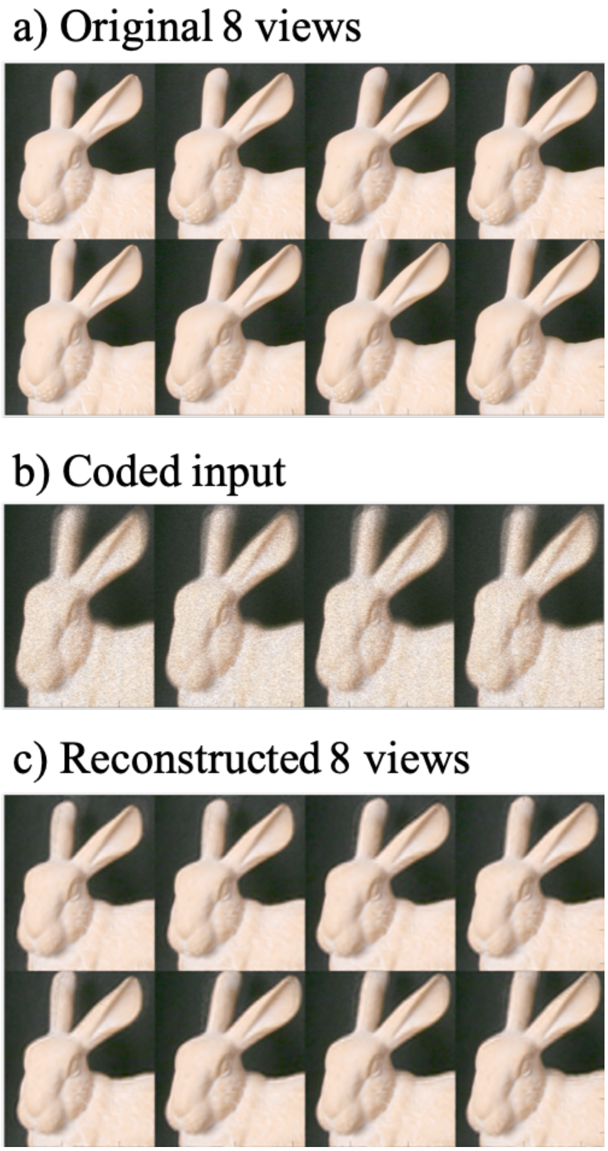

Figure 7: Reconstruction results for ‘Rabbit’

5.2. Reconstruction time

The reconstruction for 8 views, 4 acquisitions, image size 512-by-512 takes about 15 minutes on a 2015 Macbook pro with 8-core 2.3Ghz CPU and 16Gb memory. For some applications, real-time reconstruction may be necessary. This optimization-based reconstruction is time-consuming because it is iterative. One way to accelerate the reconstruction is by using better initialization instead of 0, and one feasible choice is to initialize the image by dividing the coded image by the mask. In addition, in each iteration, solving the linear problem is also iterative by conjugate gradient, which takes dominate time since the proximal update is usually fast and may takes linear time. There is not much space to accelerate this step. However, in some cases, for instance when the linear transformations are identity, the big linear problem involving multi-view images can be split into many small voxel-by-voxel problems, which may have analytical solutions and does not need to be solved iteratively. There have also been efforts of training neural networks to solve this kind of optimization problems, but this is beyond the scope of this work.

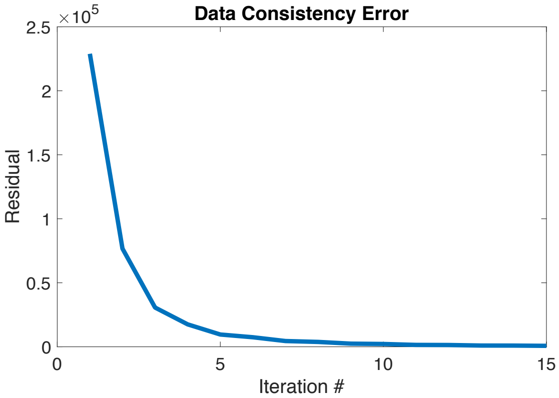

Figure 8: Residual plot as a function of iteration

5.3. Encoding mask and results evaluation

The encoding mask is a key factor of the design of this camera. For the reconstruction, we are actually trying to solve problem in the format of Ax = b, and A is based on the encoding mask. Currently we use a pixel-wise uniform distributed mask for easy implementation, and careful design of the mask may make the problem more incoherent, which can help with compressed sensing. The evaluation of results is based on the visual performance, PSNR and SSIM compared with original view. The accuracy of depth map and other information we can get from light field data can be also used to evaluate the results.

6. Summary

In this work, we proposed a two-component model for reconstruction of compressive light field photography. We implemented this algorithm using ADMM and tested various prior combinations. We were able to successfully recover the angular information for three different data sets and obtained high image quality.

7. Appendix

ADMM can be used to solve problems in the following format iteratively, where f and g are convex,

$$\begin{aligned} &\min_{X_1,X_2} f(X_1, X_2) + g_1(Z_1) + g_2(Z_2) \\ &s.t. A_1X_1 + B_1Z_1 = c_1 \\ &\quad A_2X_2 + B_2X_2 = c_2 \end{aligned}$$

The Augmented Lagrangian is

$$\begin{aligned} \mathcal{L}_\rho(X_1, X_2, Z_1, Z_2, U_1, U_2) = f(X_1, X_2) + g_1(Z_1) + g_2(Z_2) \\ +\rho/2(\|A_1X_1 + B_1Z_1 - c_1 + U_1\|_2^2 \\ + \|A_2X_2 + B_2X_2 - c_2 + U_2\|_2^2) \end{aligned}$$

The update rule is as follows,

$$\begin{aligned} &X_1^{k+1}, X_2^{k+1} = argmin_{X_1,X_2}\mathcal{L}_\rho(X_1, X_2, Z_1^k, Z_2^k, U_1^k, U_2^k) \\ &Z_1^{k+1} = argmin_{Z_1}\mathcal{L}_\rho(X_1^{k+1}, X_2^{k+1}, Z_1, Z_2^k, U_1^k, U_2^k) \\ &Z_2^{k+1} = argmin_{Z_2}\mathcal{L}_\rho(X_1^{k+1}, X_2^{k+1}, Z_1^k, Z_2, U_1^k, U_2^k) \\ &U_1^{k+1} = U_1^k + A_1X_1^{k+1} - B_1Z_1^{k+1} - c_1 \\ &U_2^{k+1} = U_2^k + A_2X_2^{k+1} - B_2Z_2^{k+1} - c_2 \end{aligned}$$

References

[1] E.H.Adelson, J.R.Bergen, The plenoptic function and the elements of early vision, in: Computational Models of Visual Processing, 1991.

[2] R. Ng, M. Levoy, M. Brooif, G. Duval, M. Horowitz, and P. Han¬ rahan. Light field photography with a hand-held plenoptic camera. Technical Report CSTR 2005-02,Stanford University, April 2005.

[3] LEVIN, A., HASINOFF, S. W., GREEN, P., DURAND, F., AND FREEMAN, W. T. 2009. 4D Frequency Analysis of Computational Cameras for Depth of Field Extension. ACM Trans. Graph. (SIGGRAPH) 28, 3, 97.

[4] Ting-Chun Wang, Alexei A. Efros, Ravi Ramamoorthi; The IEEE International Conference on Computer Vision (ICCV), 2015, pp. 3487-3495

[5] T.E. Bishop, S. Zanetti, P. Favaro, Light field superresolution, in: Proceedings of the ICCP, IEEE, 2009, pp. 1–9.

[6]K. Marwah,G. Wetzstein, Y. Bando, R. Raskar, Compressive light field photography using overcomplete dictionaries and optimized projections, ACM Trans. Graph. (SIGGRAPH) 32 (4) (2013) 46.

[7] Mahdad Hosseini Kamal, Barmak Heshmat, Ramesh Raskar, Pierre Vandergheynst, Gordon Wetzstein, Tensor low-rank and sparse light field photography, Computer Vision and Image Understanding 145 (2016) 172–181

[8] Lustig, Michael, David Donoho, and John M. Pauly. "Sparse MRI: The application of compressed sensing for rapid MR imaging." Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine 58.6 (2007): 1182-1195.

[9] Trzasko, Joshua D., and Armando Manduca. "Calibrationless parallel MRI using CLEAR." 2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR). IEEE, 2011.

[10] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foun- dations and Trends in Machine Learning, 3(1):1–122, 2011.