Blind Unitary Transform Learning for Inverse Problems in Light-Field Imaging

Cameron J. Blocker, Jeffrey A. Fessler*
Department of Electrical and Computer Engineering
The University of Michigan, Ann Arbor, MI, USA
cblocker@umich.edu, fessler@umich.edu

Abstract

Light-field cameras have enabled a new class of digital post-processing techniques. Unfortunately, the sampling requirements needed to capture a 4D color light-field directly using a microlens array requires sacrificing spatial resolution and SNR in return for greater angular resolution. Because recovering the true light-field from focal-stack data is an ill-posed inverse problem, we propose using blind unitary transform learning (UTL) as a regularizer. UTL attempts to learn a set of filters that maximize the sparsity of the encoded representation. This paper investigates which dimensions of a light-field are most sparsifiable by UTL and lead to the best reconstruction performance. We apply the UTL regularizer to light-field inpainting and focal stack reconstruction problems and find it improves performance over traditional hand-crafted regularizers.

1. Introduction

1.1. Light-field Imaging

In optical imaging, it is often sufficient to characterize light from a geometric optics perspective that treats all light as rays. If one can characterize all the rays of light within a space, then one can simulate all possible images taken within that space. A ray $r = (x, y, z, \theta, \phi, \lambda, t)$ is parameterized by its spatial position, its angular orientation, and its spectral color, as a function of time. We would like to know the value of the *plenoptic function* that assigns a non-negative scalar irradiance $P(x, y, z, \theta, \phi, \lambda, t)$ for every ray in ray space, where $P : \mathbb{R}^3 \times \mathbb{S}^2 \times \mathbb{R}_{++} \times \mathbb{R} \mapsto \mathbb{R}_+$.

Characterizing the plenoptic function over an arbitrary space is difficult and rarely undertaken in practice. To simplify, often one considers only the rays in a space bounded by two planes that is free of occluders or light-sources, where light propagates freely in one general direction (see Fig. 1). In this context, one can reparameterize the 5-D spatio-angular coordinates of the plenoptic function in 4 dimensions: the (u, v) coordinate where rays intercept the entry plane, and the (x, y) coordinate where they intercept the exit plane. A scalar function over these free space parameterizations is called a *light-field* $L(x, y, u, v)$, and one generally drops the spectral and temporal dimensions when not needed.

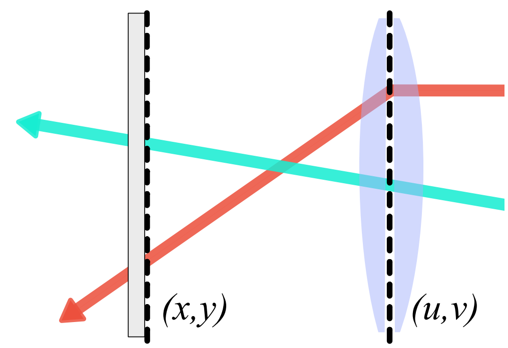

Figure 1. Parameterization of all light rays within a camera using the sensor and aperture planes. All rays of interest must intersect these two planes.

While this context may seem restrictive at first, it is exactly the situation that arises for light rays inside a camera. Every ray of interest in a camera must enter the camera through the aperture plane and terminate at the sensor plane. These two planes provide a natural parameterization for the rays in the camera. A light-field, once acquired, can be used to simulate different focal settings by a simple rebinning of rays to the spatial locations where they would have terminated.

Hand-held light-field cameras, such as those made by Lytro and Ratrix, acquire the 4D light-field by multiplexing angular coordinates with spatial coordinates using a microlens array. In effect, each microlens acts as a miniature camera that takes a picture of the aperture plane from within the camera, so unique rays are determined by which microlens picture they end up in and where in said picture they terminate. For a fixed sensor size, this configuration reduces the measured spatial resolution by a factor of the

*Supported in part by a W. M. Keck Foundation grant.

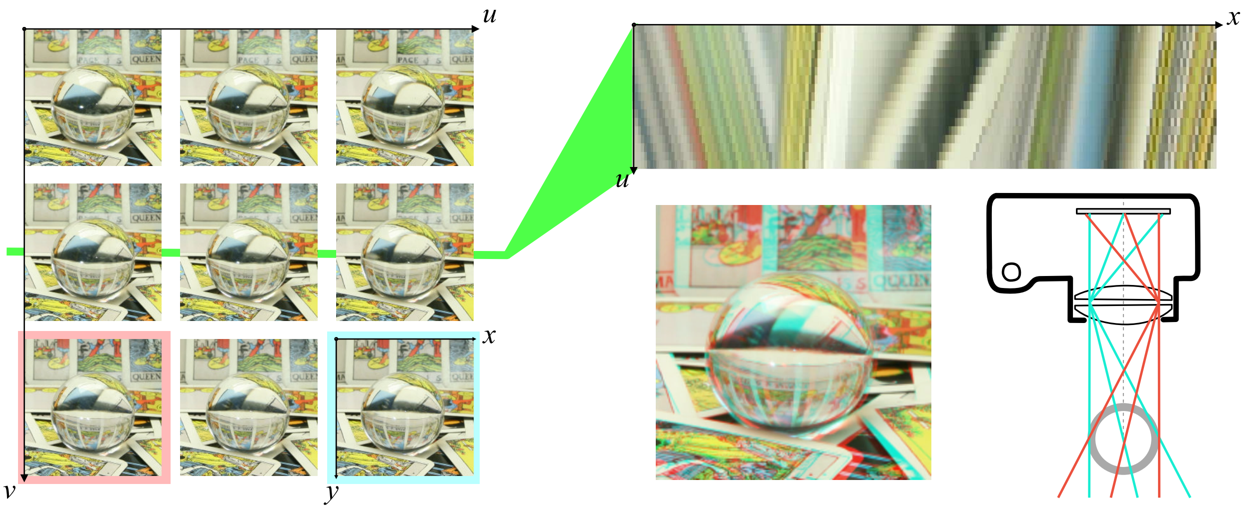

Figure 2. The anatomy of a light-field. (Left) An intuitive interpretation of light-field is a matrix of subaperture images. (Bottom) Two subaperture images, highlighted in red and cyan, exhibit a shift in perspective of the scene. Theses differences are highlighted in red and cyan in the enlarged image, and a 2D diagram shows how the perspective shift relates to the camera geometry. (Top) An epipolar image is a 2D slice of the 4D light-field in an angular and spatial dimension. Non-specular or *Lambertian* points in the scene, that emit the same ray information in all directions, draw out lines as they shift through the perspective dimension.

angular resolution, leading to an undesirable trade-off.

Another way to capture a light-field is through a camera array or camera gantry. In this setup, a camera is placed at different locations along a virtual aperture plane. Images acquired at each location represent the (x, y) coordinates for some fixed (u, v). For a fixed array size, camera arrays are limited in their aperture plane resolution by the physical size of the camera. Increasing the array size adds both bulk and expense. Camera gantries suffer from poor temporal resolution, due to the requirement to physically move the camera.

While this capture method may not be applicable to all situations, it provides an intuitive interpretation of a light-field as a 2D array of images, each with a slight shift in perspective. Each of these *subaperture images* (SAI) provides a view of the scene through a specific point in the real or simulated aperture. If instead of fixing both angular coordinates, we fix one spatial coordinate and one angular coordinate, we get what is called an *epipolar image* (EPI). Figure 2 shows an example light-field in terms of both its SAI slices as well as an EPI slice.

Despite the redundant structure of these light-field dimensions, traditional light-field imaging methods are burdened with capturing the full 4D light-field structure directly. Microlens based light-field cameras must trade off spatial resolution, and camera arrays must add additional bulk and expense. In response to this sampling burden and the apparent redundancy in the light-field between SAI views, several compressive light-field imaging methods have been proposed. One method is focal stack reconstruction where the light-field is recovered from a series of images captured with different focal settings. While alleviating some of the sampling burden, reconstructing a light-field from a focal stack presents an additional challenge: information about the light-field is invariably lost due to the dimensionality gap [14]. The full 4D light-field can not be directly recovered from a 1D set of 2D measurements without enforcing additional assumptions.

1.2. Inverse Problems

Reconstructing a light-field from a set of compressed or subsampled measurements is an underdetermined inverse problem. There can be many possible light-fields that will perfectly match our data. Thus a model is needed to select one of the many candidate light-fields, by choosing one that is consistent with our assumptions about the true light-field's properties. A common paradigm that we use in this work is to include the model as regularization in a minimization problem

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}}\, \lambda \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \mathrm{R}(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{A}$ is a wide matrix encoding the linear operation relating the unknown light-field $\boldsymbol{x}$ to the measurement $\boldsymbol{y}$, λ is a hyperparameter representing our confidence in the measurements, and $\mathrm{R}(\boldsymbol{x})$ is a regularization function representing our signal model.

A number of previous works attempt to restore a light-field from a set of compressed or corrupted measurements, such as view inpainting, focal stack reconstruction, coded aperture reconstruction, super-resolution, denoising, and inpainting. A majority of these approaches can be divided into linear filtering based methods [6, 7, 14], depth-estimation-dependent methods [15, 18, 21], deep learning methods [10, 13, 19, 31, 32, 33], and low-rank or sparse methods [2, 3, 5, 8, 9, 11, 12, 16, 17, 25, 26, 27, 28]. Most of the sparsity based methods assume a hand-crafted transform, such as the discrete cosine transform (DCT) [17] or shearlets [27, 28]. A notable exception is [16] that applies K-SVD to learn a dictionary for light-field patches from training data apriori. While using hand-crafted transforms, LF-BM5D [2] does employ instance-adaptive thresholding and filtering.

Transform sparsity models data as being locally sparsifiable. In other words, we assume $\boldsymbol{W}\boldsymbol{P}_j\boldsymbol{x}$ is sparse, where $\boldsymbol{P}_j$ is a matrix of 0 and 1 elements that extracts the jth, for example, $p_x \times p_y \times p_u \times p_v \times p_c$ patch or window from the data, and $\boldsymbol{W}$ is a transform that sparsifies the patch. Compared to dictionary methods, that generally synthesize a signal vector from a set of sparse codes, transform sparsity encourages a signal to be sparsifiable. These conditions are not necessarily equivalent, except in the uncommon case when the dictionary and transform are inverses of each other.

In transform learning, we attempt to learn a transform from data, instead of using a hand-crafted transform such as wavelets or the DCT. There are multiple modes of transform learning. One mode is to use a set of training signals $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K\}$ and learn a transform $\boldsymbol{W}$ that is effective for sparsifying patches drawn from those signals. In words, we want $\boldsymbol{W}$ such that $\boldsymbol{W}\boldsymbol{P}_j\boldsymbol{x}_k$ is typically sparse. One way this can be done is by minimizing the following cost function

$$\hat{\boldsymbol{W}} = \operatorname*{argmin}_{\boldsymbol{W}\in\mathcal{U}} \min_{\{\boldsymbol{z}_{j,k}\}} \sum_{k=1}^{K} \sum_{j} \|\boldsymbol{W}\boldsymbol{P}_j\boldsymbol{x}_k - \boldsymbol{z}_{j,k}\|_2^2 + \gamma^2 \|\boldsymbol{z}_{j,k}\|_0, \tag{2}$$

where $\mathcal{U}$ denotes the set of unitary matrices. This approach bears many similarities to a standard dictionary learning formulation

$$\hat{\boldsymbol{D}} = \operatorname*{argmin}_{\boldsymbol{D}} \min_{\{z_j\}} \sum_{k=1}^{K} \sum_{j} \|\boldsymbol{P}_j\boldsymbol{x}_k - \boldsymbol{D}\boldsymbol{z}_{j,k}\|_2^2 + \gamma^2 \|z_{j,k}\|_0. \tag{3}$$

Transform learning methods have been applied in the context of 2D image denoising [23], MR image reconstruction from undersampled k-space measurements [24], and video denoising [30].

1.3. Contributions

Because of the unitary invariance of the ℓ_2 norm, unitary transform learning is equivalent to a formulation of unitary dictionary learning. Thus our proposed method is most similar to that of Marwah et al. [16]. The work proposed here differs in two major aspects.

First, we do not learn our transforms from training data *a priori*. We instead opt for a *blind* UTL method that learns sparsifying transforms blindly in a instance-adaptive fashion. To the authors' knowledge, this is the first time instance-adaptive transform or dictionary sparsity has been applied to light-field imaging.

Second, we investigate the sparsifiability of different dimensions of the light-field. [16] used a 5D (4D + color) light-field patch in learning and fitting their dictionary. While dictionary atoms describing epipolar patches or spatial patches could, in theory, be learned inside of a 5D patch, often dense patches are learned. Due to the non-convexity of these learning methods, it is unclear if the learned dense 5D patches are optimal. Indeed much of the prior work can be divided among epipolar methods such as [27, 28, 31] and 4D+ methods [2, 11, 17, 19].

This work explores multiple approaches to choosing light-field patches for transform and dictionary learning, including subaperture image (SAI) patches (x, y, c), epipolar image (EPI) patches in both the horizontal (x, u, c) and vertical (y, v, c) directions as well as full dimensional light-field (LF) patches (x, y, u, v, c). In a hand-crafted and pre-learned setting, applying a method only spatially along subaperture images completely ignores light-field structure. In contrast, in the blind setting, features can in theory be learned more effectively due to the light-field redundancy. As different light-field imaging applications may benefit more from different types of patches, we compare different patch dimension choices on a couple of inverse problems in light-field imaging: inpainting and reconstruction from focal stack images.

Section 2 provides a general description of unitary transform learning (UTL) as used in this work. For a more detailed description of UTL, including a convergence analysis, see [24]. Section 3 applies UTL with different patch structures to inverse problems in light-field imaging. Section 4 compares the performance of the different methods and analyzes the learned transforms.

2. Methods

We apply blind unitary transform learning as a regularizer for the problem of recovering a light-field $\boldsymbol{x}$ from measurements $\boldsymbol{y}$ by minimizing the following cost function using block coordinate descent (BCD):

$$\begin{aligned} \hat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^N} & \min_{\{\boldsymbol{z}_j\in\mathbb{R}^N\}} \min_{\boldsymbol{W}\in\mathcal{U}} \\ & \lambda\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \sum_j \|\boldsymbol{W}\boldsymbol{P}_j\boldsymbol{x} - \boldsymbol{z}_j\|_2^2 + \gamma^2\|\boldsymbol{z}_j\|_0. \end{aligned} \tag{4}$$

We let $\boldsymbol{A} \in \mathbb{R}^{M\times N}$ represent our system model that generated vectorized measurements $\boldsymbol{y} \in \mathbb{R}^M$. Here $\boldsymbol{P}_j \in$

$\{0,1\}^{n\times N}$ is a matrix that extracts the jth $p_x \times p_y \times p_u \times p_v \times p_c$ patch from a vectorized light-field $\boldsymbol{x}$ and we sum over all such j with overlapping windows of stride 1. Here, $\|\cdot\|_0$ denotes the so-called zero "norm" or counting measure (number of nonzero vector elements).

While (4) is a nonconvex cost function, due both to the nonconvex zero "norm" and product between $\boldsymbol{W}$ and $\boldsymbol{x}$, applying BCD to it is globally convergent, *i.e.*, BCD converges to a local minima from any initial starting point [24]. Note that without a constraint on $\boldsymbol{W}$, a trivial minimizer would be $\boldsymbol{W} = 0, \boldsymbol{z}_j = 0$. A unitary constraint avoids this problem and has an efficient closed-form update.

We alternate between minimizing the sparse codes $\{\boldsymbol{z}_j\}$, the unitary transform $\boldsymbol{W}$, and the light-field $\boldsymbol{x}$. Minimizing (4) with respect to $\boldsymbol{z}_j$ leads to the following proximal problem with known closed-form solution:

$$\hat{\boldsymbol{z}}_j = \underset{\boldsymbol{z}_j}{\operatorname{argmin}} \|\boldsymbol{W}\boldsymbol{P}_j\boldsymbol{x} - \boldsymbol{z}_j\|_2^2 + \gamma^2\|\boldsymbol{z}_j\|_0 \tag{5}$$

$$\hat{\boldsymbol{z}}_j = \mathrm{H}_\gamma(\boldsymbol{W}\boldsymbol{P}_j\boldsymbol{x}) \tag{6}$$

where $\mathrm{H}_\gamma(\cdot)$ is element-wise hard-thresholding by threshold γ

$$\mathrm{H}_\gamma(a) = \begin{cases} 0 & |a| \leq \gamma \\ a & |a| > \gamma. \end{cases} \tag{7}$$

Compared to dictionary learning, where sparse coding is NP-Hard and requires an expensive step such as Orthogonal Matching Pursuit (OMP), transform learning provides a simple closed-form sparse code update.

Defining $\boldsymbol{X} = [\boldsymbol{P}_1\boldsymbol{x} \ldots \boldsymbol{P}_J\boldsymbol{x}]$ and $\boldsymbol{Z} = [\boldsymbol{z}_1 \ldots \boldsymbol{z}_J]$, we rewrite the $\boldsymbol{W}$ update as a Procrustes problem with known closed-form solution:

$$\begin{aligned} \hat{\boldsymbol{W}} &= \underset{\boldsymbol{W}\in\mathcal{U}}{\operatorname{argmin}} \sum_j \|\boldsymbol{W}\boldsymbol{P}_j\boldsymbol{x} - \boldsymbol{z}_j\|_2^2 \\ &= \underset{\boldsymbol{W}\in\mathcal{U}}{\operatorname{argmin}} \|\boldsymbol{W}\boldsymbol{X} - \boldsymbol{Z}\|_F^2 = \boldsymbol{U}\boldsymbol{V}^T \end{aligned} \tag{8}$$

where $\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}^T$ denotes the SVD of $\boldsymbol{Z}\boldsymbol{X}^T$. Because $\boldsymbol{W}$ is an $n \times n$ matrix, where n is the number of elements in a patch, this SVD is performed on a relatively small matrix.

The light-field update is then a standard quadratic minimization problem

$$\begin{aligned} \hat{\boldsymbol{x}} &= \underset{x}{\operatorname{argmin}}\, \lambda\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \sum_j \|\boldsymbol{W}\boldsymbol{P}_j\boldsymbol{x} - \boldsymbol{z}_j\|_2^2 \\ &= (\lambda\boldsymbol{A}^T\boldsymbol{A} + \sum_j \boldsymbol{P}_j^T\boldsymbol{P}_j)^{-1}(\lambda\boldsymbol{A}^T\boldsymbol{y} + \sum_j \boldsymbol{P}_j^T\boldsymbol{W}^T\boldsymbol{z}_j) \end{aligned} \tag{9}$$

Because $\sum_j \boldsymbol{P}_j^T\boldsymbol{P}_j$ is a diagonal matrix, system models that have a diagonizable Hessian matrix $\boldsymbol{A}^T\boldsymbol{A}$, such as denoising, inpainting or deblurring, are efficiently computed in closed-form. For all other cases, running a few iterations of conjugate gradient provides a suitable approximation.

Algorithm 1 Blind UTL

Require: $\boldsymbol{x}^{(0)}$, $\boldsymbol{W}^{(0)}$ $\boldsymbol{y}$, $\boldsymbol{A}$, $\lambda, \gamma > 0$
  Let $\boldsymbol{G} = \sum_j \boldsymbol{P}_j^T\boldsymbol{P}_j$
  for $i = 1, \ldots, I$ **do**
    Construct $\boldsymbol{X} = [\boldsymbol{P}_1\boldsymbol{x}^{(i-1)} \ldots \boldsymbol{P}_j\boldsymbol{x}^{(i-1)} \ldots \boldsymbol{P}_J\boldsymbol{x}^{(i-1)}]$
    $\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}^T = \mathrm{svd}(\boldsymbol{Z}^{(i-1)}\boldsymbol{X}^T)$
    $\boldsymbol{W}^{(i)} = \boldsymbol{U}\boldsymbol{V}^T$
    $\boldsymbol{Z}^{(i)} = \mathrm{H}_\gamma(\boldsymbol{W}^{(i)}\boldsymbol{X})$
    Construct $\bar{\boldsymbol{x}} = \sum_j \boldsymbol{P}_j^T\boldsymbol{W}^{(i)T}\boldsymbol{Z}^{(i)}_{:,j}$
    $\boldsymbol{x}^{(i)} = (\lambda\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{G})^{-1}(\lambda\boldsymbol{A}^T\boldsymbol{y} + \bar{\boldsymbol{x}})$
  end for

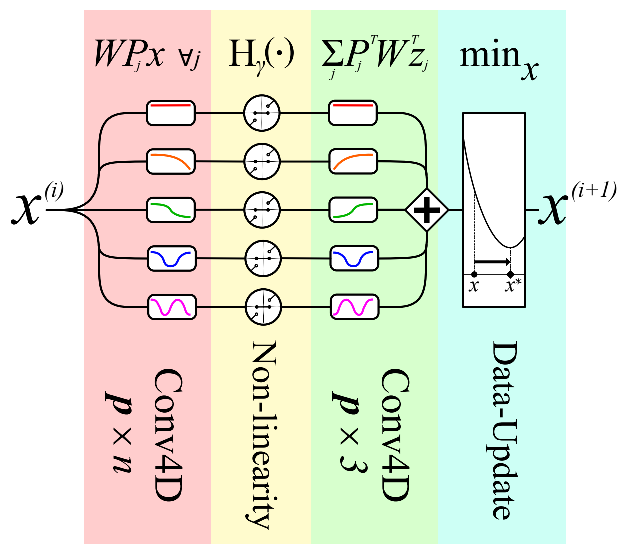

Figure 3. Regularization based on transform learning can be interpreted as a filter bank followed by a data update term. A filter bank can be interpreted as a shallow Convolutional Neural Network (CNN). The red and yellow regions correspond to (6) and the green and blue regions correspond to (9). (The update of the transform $\boldsymbol{W}$ in (8) each iteration is not pictured.)

Algorithm 1 summarizes a patch-wise implementation of blind unitary transform learning as described above. Another interpretation can be understood by examining the relationship of the rows of $\boldsymbol{W}$ with the original light-field $\boldsymbol{x}$. Each row of $\boldsymbol{W}$ performs an inner-product with a sliding window in $\boldsymbol{x}$, which is equivalent to filtering [22]. Thus the set of sparse codes $\boldsymbol{Z}$ represent the thresholded output of a filter bank of n filters, where n is the number of rows of $\boldsymbol{W}$. Applying the inverse transform $\boldsymbol{W}^T$ and aggregating is equivalent to filtering with matched filters and summing

Figure 4. Central subaperture images from The (New) Stanford Light Field Archive [1]. From left to right, (Top) *amethyst, Lego knights, crystal ball*, (Middle) *bracelet, jelly beans, bunny, eucalyptus*, (Bottom) *treasure chest, Lego truck, Lego bulldozer*. Images resized independently.

the channels. Figure 3 shows a diagram of the flow of $\boldsymbol{x}$ in one iteration. (Note that it does not show the update of the filters $\boldsymbol{W}$). Thus we can interpret each iteration of blind unitary transform learning as an *instance-adaptive* shallow CNN, where the filters are learned dynamically in an unsupervised fashion, followed by a data update that incorporates our prior knowledge on $\boldsymbol{y}$ and $\boldsymbol{A}$.

3. Experiments

We validated the proposed method on 10 light-fields from the Stanford Light-field Dataset [1]; see Figure 4. From each light-field, we extracted the central 5×5 views and spatially downsampled by a factor of 3 for testing our method. For each patch shape, we tuned all hyperparameters, unless otherwise stated, using the Tree of Parzen Estimators as implemented in the `hyperopt` Python package [4]. For hyperparameter tuning, we used smaller $5 \times 5 \times 192 \times 192$ light-fields cropped from the bunny, crystal ball, and Lego bulldozer light-fields to reduce tuning time. We used peak signal-to-noise ratio (PSNR) as the criterion for tuning and for method evaluation.

We investigated light-field inpainting and light-field reconstruction from focal stack images. In all cases, we initialized the transform $\boldsymbol{W}$ with the $p_x \times p_y \times p_u \times p_v \times p_c$-point DCT and ran UTL for 120 iterations.

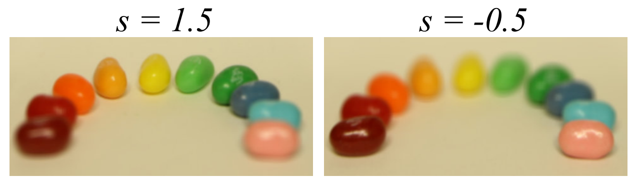

Figure 5. An example 2-image focal stack and the amount of pixel shift, s, applied to light-field before summing. Different jelly beans come into focus as the focal plane passes through the scene

Method	n	Patch Shape $(p_x, p_y, p_u, p_v, p_c)$	γ
SAI UTL	108	(6, 6, 1, 1, 3)	0.0625
EPI UTL	135	(9, 1, 5, 1, 3)	0.0582
LF UTL	243	(3, 3, 3, 3, 3)	0.0454

Table 1. Patch Shape and Thresholds for inpainting problem. For brevity, we only list (x, u) patch dimension for EPI UTL, although we learn filters for the corresponding patches in (y, v) as well.

3.1. Inpainting light-fields

We apply blind UTL to light-field inpainting by minimizing (4) with $\boldsymbol{A} = \mathrm{diag}(\mathrm{vec}(M))$ where

$$M[i,j,k,l,c] = \begin{cases} 1 & (i,j,k,l,c) \in \Omega \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

and Ω is the set of samples taken of the light-field. In our experiments, Ω is such that only 20% of all samples are kept at random. Samples in Ω were drawn independently for each light-field tested. We used spatial cubic interpolation to initialize $\boldsymbol{x}$.

For the inpainting problem, we let $\lambda = 10^8$ and tuned the patch shape and sparsity threshold, γ, for all three patch shapes. In all cases, we used the full color patch dimension of 3. For SAI UTL and LF UTL, patch dimensions in x and y were constrained to be equal, while in LF UTL patch dimensions in u and v were similarly constrained. All methods had an upper bound on the largest patch that could be chosen due to memory constraints, as updating $\boldsymbol{W}$ in a blind setting precludes computing $\boldsymbol{Z}$ on-the-fly. While SAI UTL and EPI UTL did not reach that bound, and instead settled on smaller patch sizes, LF UTL did, due to its increased dimensionality. Table 1 lists the tuned hyperparameters for the three cases.

We applied each of the three methods with their tuned hyperparameters to the 10 light-field datasets. Table 2 shows the PSNR of each of the reconstructions. LF UTL surpassed EPI UTL by 1.5dB and SAI UTL by 6.2dB on average. Figure 6 compares the performance of each of the methods on a zoomed in section of the Lego truck light-field. LF UTL is able to preserve fine features more accurately than any of the other methods.

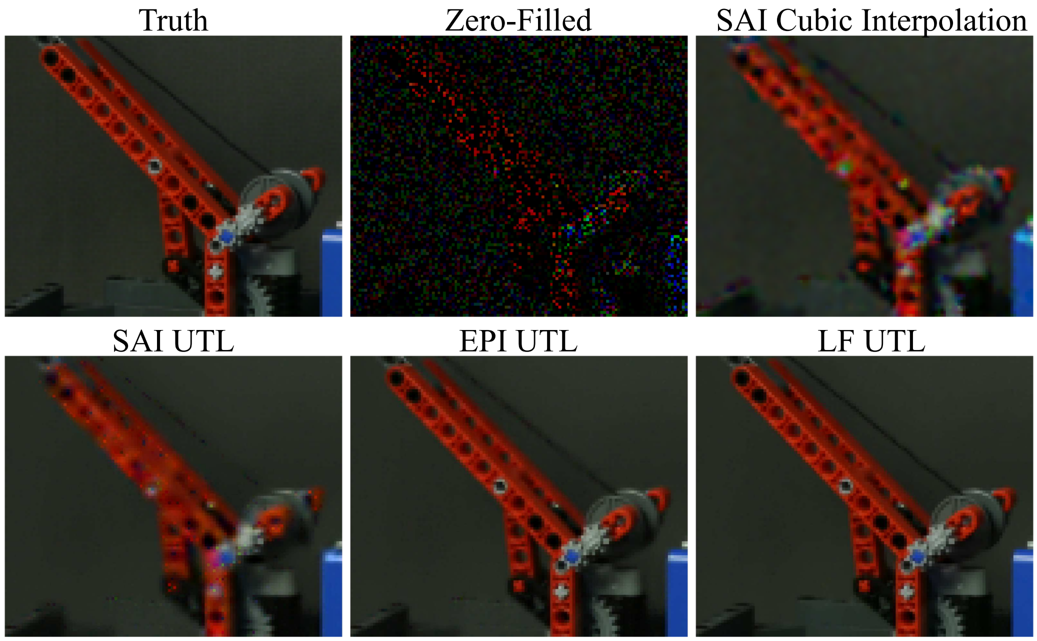

Figure 6. Zoomed in view of the central perspective of inpainted Lego truck light-fields.

Inpainting	Cubic Interpolation		Proposed blind UTL methods		
	SAI	EPI	SAI UTL	EPI UTL	LF UTL
amethyst	28.50 dB	26.74 dB	31.72 dB	37.67 dB	**39.79** dB
crystal ball	21.56 dB	21.29 dB	25.68 dB	30.22 dB	**30.71** dB
Lego bulldozer	26.69 dB	23.95 dB	29.94 dB	34.02 dB	**35.63** dB
bunny	32.21 dB	28.71 dB	35.21 dB	39.49 dB	**42.04** dB
bracelet	23.28 dB	22.84 dB	27.99 dB	34.70 dB	**34.90** dB
eucalyptus	29.32 dB	28.15 dB	32.24 dB	37.85 dB	**39.69** dB
Lego knights	26.40 dB	24.16 dB	31.29 dB	34.00 dB	**36.68** dB
treasure chest	24.57 dB	23.19 dB	27.86 dB	**32.92** dB	32.46 dB
jelly beans	35.82 dB	32.35 dB	38.16 dB	40.41 dB	41.81 dB
Lego truck	28.95 dB	27.86 dB	32.17 dB	38.25 dB	**40.62** dB
Average	27.73 dB	25.92 dB	31.23 dB	35.95 dB	**37.43** dB

Table 2. PSNR for each recovered light-field using different inpainting methods.

3.2. Reconstruction from Focal Stack Images

The capture of a photograph in a particular focal setting can be modeled by:

$$I(x, y, c) = \int_{\mathcal{A}} L(x + su, y + sv, u, v, c)\, \mathrm{d}u\, \mathrm{d}v, \tag{11}$$

where $\mathcal{A}$ is the support set of the aperture, and s is a parameter determined by the focus setting. Thus we can collect photographs with a varying focal plane by appropriately adjusting s; see for example Figure 5. For further information regarding photograph capture and its relation to Fourier subspaces, see [20, 14].

We apply $\boldsymbol{A}_s$ by shifting subaperture images by s times their u, v coordinates and summing. $\boldsymbol{A}$ then represents the application of each $\boldsymbol{A}_s$ in a stack. We used linear interpolation to shift the subaperture images. Our measurement model is then

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\eta} \tag{12}$$

where $\boldsymbol{\eta}$ denotes additive white Gaussian noise with standard deviation σ. In our experiments, we retrospectively added $\boldsymbol{\eta}$ with $\sigma = 1\%$ of the peak value of the photographs $\boldsymbol{y}$. We used shift parameters $s \in \{-1, -0.5, 0, 0.75, 1.5\}$ to simulate 5 photographs taken with our model.

We compare our proposed method against an edge-

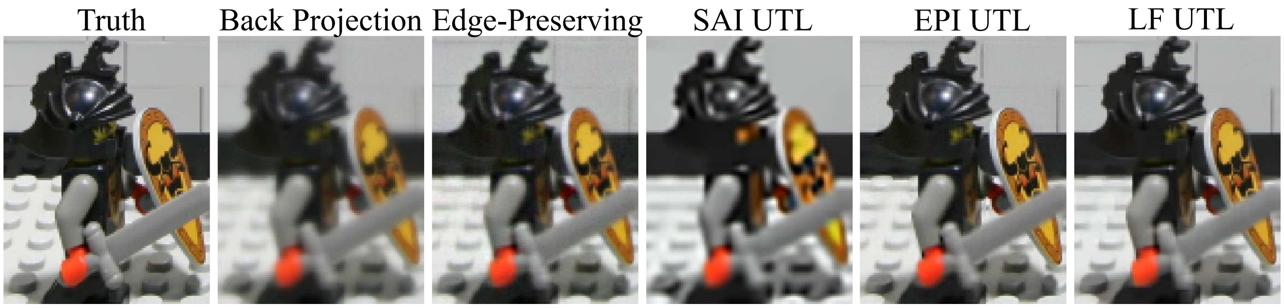

Figure 7. Zoomed in view of the central perspective of Lego knight light-fields reconstructed using different methods.

Focal Stack Reconstruction	Scaled Back Projection	Edge Preserving	Proposed SAI UTL	Proposed EPI UTL	Proposed LF UTL
amethyst	28.94 dB	33.39 dB	29.11 dB	36.19 dB	**37.23** dB
crystal ball	20.33 dB	23.30 dB	21.83 dB	24.65 dB	**24.73** dB
Lego bulldozer	23.07 dB	28.70 dB	25.36 dB	**30.01** dB	29.35 dB
bunny	29.94 dB	35.40 dB	31.57 dB	38.40 dB	**39.59** dB
bracelet	18.23 dB	24.46 dB	23.18 dB	**26.26** dB	24.51 dB
eucalyptus	30.40 dB	33.83 dB	30.37 dB	36.72 dB	**37.56** dB
Lego knights	21.93 dB	25.73 dB	24.72 dB	**28.10** dB	27.75 dB
treasure chest	24.85 dB	29.63 dB	25.77 dB	32.13 dB	**32.15** dB
jelly beans	25.08 dB	35.96 dB	32.90 dB	37.92 dB	**38.44** dB
Lego truck	24.75 dB	34.24 dB	29.68 dB	37.44 dB	**38.42** dB
Average	24.75 dB	30.46 dB	27.45 dB	32.78 dB	**32.97** dB

Table 3. PSNR for each light-field using different focal stack reconstruction methods

preserving regularizer of the form:

$$\begin{aligned}\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} & \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 \\ & + \beta_{x,y}\sum_i \psi([\boldsymbol{C}_{x,y}\boldsymbol{x}]_i;\delta_{x,y}) \\ & + \beta_{u,v}\sum_j \psi([\boldsymbol{C}_{u,v}\boldsymbol{x}]_j;\delta_{u,v}),\end{aligned} \tag{13}$$

where $\psi(\cdot,\delta)$ denotes the Huber potential function, a smooth approximation of an absolute value function. $\delta_{k,l}$ is a hyperparameter controlling the function curvature. $\boldsymbol{C}_{k,l}$ applies finite differences along dimensions k and l. We tuned $\beta_{x,y}, \delta_{x,y}, \beta_{u,v}, \delta_{u,v}$ on the same set as the UTL methods, which resulted in $7.47, 98.9, 3\times10^{-2}, 6\times10^{-4}$ for each parameter respectively.

For each of the UTL methods, we used the patch shape and threshold learned during inpainting and tuned λ, which resulted in $4.14\times10^{-2}, 2.61\times10^{-1}, 2.27\times10^{-1}$ for SAI, EPI, and LF UTL respectively. For the data update, we used 5 conjugate gradient iterations. Table 3 shows the PSNR of each of the methods applied to the 10 light-fields in our dataset. For this problem, LF UTL only out performed EPI UTL by 0.19dB on average. Figure 7 shows zoomed in views from the Lego knights light-field.

4. Discussion

Figure 8 shows the reshaped rows of the transform learned on the epipolar patches of amethyst during blind inpainting. Each of these transform patches is effectively convolved with the light-field for regularization. The filters have learned the mostly vertical linear structure of the epipolar domain by learning vertical finite-difference-like operations. We also see the slight tilt in some of the structures, reflecting the skew in out-of-focus pixels. As expected, most of this vertical structure is captured in luminance rather than color channels.

Figure 9 shows the filters learned using SAI UTL during blind inpainting. Similar to the EPI case, we find finite-difference like structures in the luminance channels, but with less vertically aligned structure. In both cases, UTL learned shifted versions of the same filter. This is a weakness of the unitary constraint, because shifted versions of the same filter can be orthogonal, but provide no new information for regularizing the reconstruction. The unitary constraint also forces one to learn a low-pass filter, which one does not expect to induce sparsity in general. We leave it as future work to investigate more effective constraints on the learned filters, such as Fourier magnitude incoher-

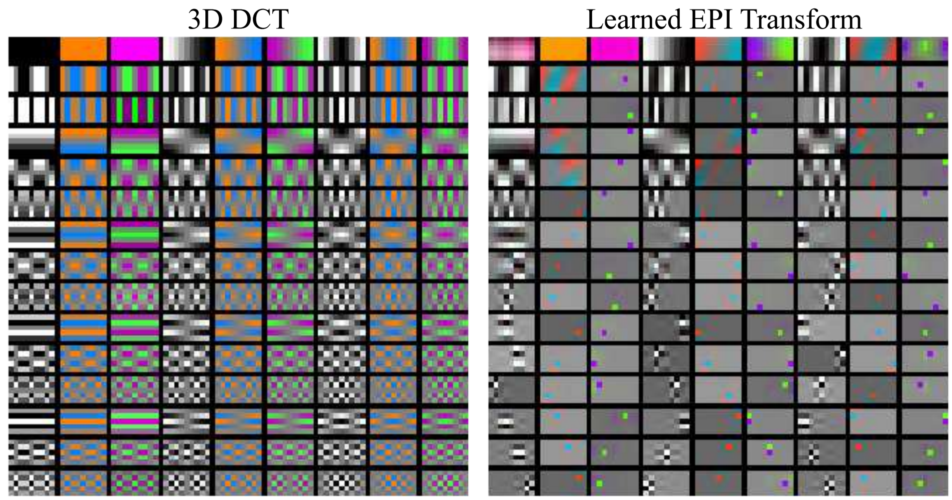

Figure 8. Comparison of the filters learned using EPI UTL (right) with those of the 3D DCT (left) used to initialize the blind inpainting method. The learned filters adapt to the vertical linear structure of the EPI light-field slices.

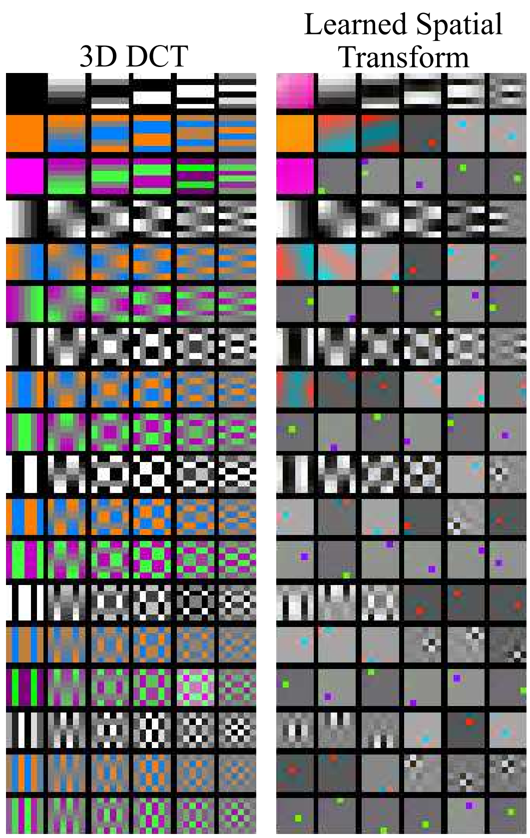

Figure 9. Comparison of the filters learned using SAI UTL (right) with those of the 3D DCT (left) used to initialize the method.

ence [22] or tight-frame conditions on a wide transform, or some other consideration of light-field physics.

We found that full dimensional patches best represented our data, but their increased dimensionality limited their receptive field in any one dimension due to memory constraints in storing $\boldsymbol{Z}$. As we assume many (but not all) of the rows of $\boldsymbol{Z}$ to be sparse, we believe UTL can be optimized for more efficient storage. An alternative may be to store a subsampled or sketch of $\boldsymbol{Z}$, and only approximate the $\boldsymbol{W}$ update. This work focused on maximally overlapping patches with a stride of 1, but larger strides could be used. We leave it as future work to investigate how these memory saving techniques impact reconstruction accuracy.

Because EPI patches were able to regularize the data nearly as well as full LF patches, presumably because of the shifting structure of the EPI dimensions, it would be interesting to see if a union of SAI and EPI transforms could capture the light-field structure as well as full LF patches. Such unions have been effective in other inverse problems [34]. Combining adaptive sparsity with other regularizers such as low-rank models may also be effective [29].

5. Conclusion

This work investigated the effectiveness of using learned sparsifying transforms for different patch structures to regularize light-field inverse problems. We found that full-dimensional patches provided the best data model, but EPI patches could capture most of the signal model with a lower dimensionality. We validated our proposed light-field models on two inverse problems: light-field inpainting and focal stack reconstruction. In both cases, regularization using transform learning yielded better reconstruction PSNR than simple hand-crafted methods.

References

[1] The (New) Stanford Light Field Archive. http://lightfield.stanford.edu/lfs.html.

[2] M. Alain and A. Smolic. Light field denoising by sparse 5d transform domain collaborative filtering. In *Proc. IEEE Wkshp. on Multimedia Signal Proc.*, pages 1–6, Oct. 2017.

[3] M. Alain and A. Smolic. Light field super-resolution via LFBM5D sparse coding. In *Proc. IEEE Intl. Conf. on Image Processing*, pages 2501–2505, Oct. 2018.

[4] J. Bergstra, D. Yamins, and D. D. Cox. Hyperopt: A Python library for optimizing the hyperparameters of machine learning algorithms, 2013. http://hyperopt.github.io/hyperopt/.

[5] C. Blocker, I. Y. Chun, and J. A. Fessler. Low-rank plus sparse tensor models for light-field reconstruction from focal stack data. In *Proc. IEEE Wkshp. on Image, Video, Multidim. Signal Proc.*, pages 1–5, 2018.

[6] D. Dansereau and L. T. Bruton. A 4-D dual-fan filter bank for depth filtering in light fields. *IEEE Transactions on Signal Processing*, 55(2):542–549, Feb. 2007.

[7] D. G. Dansereau, D. L. Bongiorno, O. Pizarro, and S. B. Williams. Light field image denoising using a linear 4D frequency-hyperfan all-in-focus filter. In *Proc. SPIE Computational Imaging XI*, volume 8657, Feb. 2013.

[8] E. Dib, M. Le Pendu, X. Jiang, and C. Guillemot. Super-ray based low rank approximation for light field compression. In *IEEE Data Compression Conference*, pages 369–378, Mar. 2019.

[9] R. A. Farrugia, C. Galea, and C. Guillemot. Super resolution of light field images using linear subspace projection of patch-volumes. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):1058–1071, Oct 2017.

[10] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. DeepStereo: Learning to Predict New Views from the World's Imagery. *arXiv e-prints*, Jun 2015. arXiv:1506.06825.

[11] M. Hosseini Kamal, B. Heshmat, R. Raskar, P. Vandergheynst, and G. Wetzstein. Tensor low-rank and sparse light field photography. *Computer Vision and Image Understanding*, 145:172 – 181, 2016. Light Field for Computer Vision.

[12] O. Johannsen, A. Sulc, and B. Goldluecke. What sparse light field coding reveals about scene structure. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*, pages 3262–3270, June 2016.

[13] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Trans. on Graphics*, 35(6):193, Nov. 2016.

[14] A. Levin and F. Durand. Linear view synthesis using a dimensionality gap light field prior. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*, pages 1831–8, 2010.

[15] A. Levin, W. T. Freeman, and F. Durand. Understanding camera trade-offs through a Bayesian analysis of light field projections. In *Proc. European Comp. Vision Conf.*, pages 88–101, 2008.

[16] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar. Compressive light field photography using overcomplete dictionaries and optimized projections. *ACM Trans. on Graphics*, 32(4):46:1–12, July 2013.

[17] Y. Miyagi, K. Takahashi, M. P. Tehrani, and T. Fujii. Reconstruction of compressively sampled light fields using a weighted 4D-DCT basis. In *Proc. IEEE Intl. Conf. on Image Processing*, pages 502–506. IEEE, Sept. 2015.

[18] A. Mousnier, E. Vural, and C. Guillemot. Partial light field tomographic reconstruction from a fixed-camera focal stack. *arXiv e-prints*, 2015. arXiv:1503.01903.

[19] O. Nabati, R. Giryes, and D. Mendlovic. Fast and accurate reconstruction of compressed color light field. In *Proc. Intl. Conf. Comp. Photography*, 2018.

[20] R. Ng. Fourier slice photography. *ACM Trans. on Graphics*, 24(3):735–44, July 2005.

[21] M. L. Pendu, C. Guillemot, and A. Smolic. A Fourier Disparity Layer representation for Light Fields. *arXiv e-prints*, Jan. 2019. arXiv:1901.06919.

[22] L. Pfister and Y. Bresler. Learning filter bank sparsifying transforms. *IEEE Trans. Sig. Proc.*, 67(2):504–19, Jan. 2019.

[23] S. Ravishankar and Y. Bresler. Learning sparsifying transforms. *IEEE Trans. Sig. Proc.*, 61(5):1072–86, Mar. 2013.

[24] S. Ravishankar and Y. Bresler. Efficient blind compressed sensing using sparsifying transforms with convergence guarantees and application to MRI. *SIAM J. Imaging Sci.*, 8(4):2519–57, 2015.

[25] L. Shi, H. Hassanieh, A. Davis, D. Katabi, and F. Durand. Light field reconstruction using sparsity in the continuous Fourier domain. *ACM Trans. on Graphics*, 34(1):12:1–13, Dec. 2014.

[26] K. Takahashi, S. Fujita, and T. Fujii. Good group sparsity prior for light field interpolation. In *Proc. IEEE Intl. Conf. on Image Processing*, pages 1447–1451, Sept. 2017.

[27] S. Vagharshakyan, R. Bregovic, and A. Gotchev. Accelerated shearlet-domain light field reconstruction. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):1082–1091, Oct 2017.

[28] S. Vagharshakyan, R. Bregovic, and A. Gotchev. Light field reconstruction using shearlet transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 40(1):133–147, Jan. 2018.

[29] B. Wen, Y. Li, L. Pfister, and Y. Bresler. Joint adaptive sparsity and low-rankness on the fly: an online tensor reconstruction scheme for video denoising. In *Proc. of the IEEE Intl. Conf. on Comp. Vision*, pages 241–250, 2017.

[30] B. Wen, S. Ravishankar, and Y. Bresler. Video denoising by online 3D sparsifying transform learning. In *Proc. IEEE Intl. Conf. on Image Processing*, pages 118–122, Sept. 2015.

[31] G. Wu, M. Zhao, L. Wang, Q. Dai, T. Chai, and Y. Liu. Light Field Reconstruction Using Deep Convolutional Network on EPI. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*, pages 1638–1646, July 2017.

[32] Y. Yoon, H.-G. Jeon, D. Yoo, J.-Y. Lee, and I. S. Kweon. Light-field image super-resolution using convolutional neural network. *IEEE Signal Processing Letters*, 24(6):848–852, June 2017.

[33] S. Zhang, Y. Lin, and H. Sheng. Residual networks for light field image super-resolution. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*, pages 11046–11055, 2019.

[34] X. Zheng, S. Ravishankar, Y. Long, and J. A. Fessler. PWLS-ULTRA: An efficient clustering and learning-based approach for low-dose 3D CT image reconstruction. *IEEE Trans. Med. Imag.*, 37(6):1498–510, June 2018.