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 \( P : \mathbb{R}^3 \times S^2 \times R_{++} \times \mathbb{R} \rightarrow \mathbb{R}^+ \).

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.

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...
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. These 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.

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

\[
\hat{x} = \arg\min_{x} \lambda \|Ax - y\|_2^2 + R(x)
\]

where \(A\) is a wide matrix encoding the linear operation relating the unknown light-field \(x\) to the measurement \(y\), \(\lambda\) is a hyperparameter representing our confidence in the measurements, and \(R(x)\) 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-BMSD [2] does employ instance-adaptive thresholding and filtering.

Transform sparsity models data as being locally sparsifiable. In other words, we assume $W P_j x$ is sparse, where $P_j$ is a matrix of 0 and 1 elements that extracts the $j$th, for example, $p_x \times p_y \times p_u \times p_v \times p_e$ patch or window from the data, and $W$ 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 $\{x_1, \ldots, x_K\}$ and learn a transform $W$ that is effective for sparsifying patches drawn from those signals. In words, we want $W$ such that $W P_j x_k$ is typically sparse. One way this can be done is by minimizing the following cost function

$$\hat{W} = \arg\min_{W \in \mathcal{U}} \min_{\{z_{j,k}\}} \sum_{k=1}^{K} \sum_{j} \|W P_j x_k - z_{j,k}\|_2^2 + \gamma^2 \|z_{j,k}\|_0, \quad (2)$$

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

$$\hat{D} = \arg\min_{D} \min_{\{z_{j}\}} \sum_{k=1}^{K} \sum_{j} \|P_j x_k - D z_{j,k}\|_2^2 + \gamma^2 \|z_{j,k}\|_0. \quad (3)$$

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 $\ell_2$ 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 an 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 $x$ from measurements $y$ by minimizing the following cost function using block coordinate descent (BCD):

$$\hat{x} = \arg\min_{x \in \mathbb{R}^N} \min_{\{z_{j}\}} \min_{\mathcal{U}} \lambda \|Ax - y\|_2^2 + \sum_{j} \|W P_j x - z_{j}\|_2^2 + \gamma^2 \|z_{j}\|_0. \quad (4)$$

We let $A \in \mathbb{R}^{M \times N}$ represent our system model that generated vectorized measurements $y \in \mathbb{R}^M$. Here $P_j \in \mathbb{R}^{M \times U}$.
Algorithm 1 Blind UTL

Require: $x^{(0)}, W^{(0)}, y, A, \lambda, \gamma > 0$

Let $G = \sum_j P_j^T P_j$

for $i = 1, \ldots, I$ do

Construct $X = [P_1 x^{(i-1)} \ldots P_j x^{(i-1)} \ldots P_J x^{(i-1)}]$

$U, \Sigma, V^T = \text{svd}(Z^{(i-1)} X^T)$

$W^{(i)} = U V^T$

$Z^{(i)} = H_\gamma(W^{(i)} X)$

Construct $\tilde{x} = \sum_j P_j^T W^{(i)} T Z_{:, j}^{(i)}$

$x^{(i)} = (\lambda A^T A + G)^{-1} (\lambda A^T y + \tilde{x})$

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 $W$ in (8) each iteration is not pictured.)
the channels. Figure 3 shows a diagram of the flow of $\mathbf{x}$ in one iteration. (Note that it does not show the update of the filters $\mathbf{W}$). 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 $\mathbf{y}$ and $\mathbf{A}$.

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 $5 \times 5$ 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 spatial cubic interpolation to initialize $\mathbf{x}$.

For the inpainting problem, we let $\lambda = 10^8$ and tuned the patch shape and sparsity threshold, $\gamma$, 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 $\mathbf{W}$ in a blind setting precludes computing $\mathbf{Z}$ 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>$n$</th>
<th>Patch Shape $(p_x, p_y, p_u, p_v, p_c)$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAI UTL</td>
<td>108</td>
<td>(6, 6, 1, 1, 3)</td>
<td>0.0625</td>
</tr>
<tr>
<td>EPI UTL</td>
<td>135</td>
<td>(9, 1, 5, 1, 3)</td>
<td>0.0582</td>
</tr>
<tr>
<td>LF UTL</td>
<td>243</td>
<td>(3, 3, 3, 3, 3)</td>
<td>0.0454</td>
</tr>
</tbody>
</table>

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.
3.2. Reconstruction from Focal Stack Images

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

\[ I(x, y, c) = \int_{\mathcal{A}} L(x + su, y + sv, u, v, c) \, du \, dv, \quad (11) \]

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 \( \mathbf{A}_s \) by shifting subaperture images by \( s \) times

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

\[ y = \mathbf{A} x + \eta \quad (12) \]

where \( \eta \) denotes additive white Gaussian noise with standard deviation \( \sigma \). In our experiments, we retrospectively added \( \eta \) with \( \sigma = 1\% \) of the peak value of the photographs \( y \). 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-

![Truth Zero-Filled SAI Cubic Interpolation](image)

![SAI UTL EPI UTL LF UTL](image)

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

<table>
<thead>
<tr>
<th>Inpainting</th>
<th>Cubic Interpolation</th>
<th>Proposed blind UTL methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SAI UTL EPI</td>
<td>SAI UTL EPI LF UTL</td>
</tr>
<tr>
<td>amethyst</td>
<td>28.50 dB 26.74 dB</td>
<td>31.72 dB 37.67 dB</td>
</tr>
<tr>
<td>crystal ball</td>
<td>21.56 dB 21.29 dB</td>
<td>25.68 dB 30.22 dB</td>
</tr>
<tr>
<td>Lego bulldozer</td>
<td>26.69 dB 23.95 dB</td>
<td>29.94 dB 34.02 dB</td>
</tr>
<tr>
<td>bunny</td>
<td>32.21 dB 28.71 dB</td>
<td>35.21 dB 39.49 dB</td>
</tr>
<tr>
<td>bracelet</td>
<td>23.28 dB 22.84 dB</td>
<td>27.99 dB 34.70 dB</td>
</tr>
<tr>
<td>eucalyptus</td>
<td>29.32 dB 28.15 dB</td>
<td>32.24 dB 37.85 dB</td>
</tr>
<tr>
<td>Lego knights</td>
<td>26.40 dB 24.16 dB</td>
<td>31.29 dB 34.00 dB</td>
</tr>
<tr>
<td>treasure chest</td>
<td>24.57 dB 23.19 dB</td>
<td>27.86 dB 32.92 dB</td>
</tr>
<tr>
<td>jelly beans</td>
<td>35.82 dB 32.35 dB</td>
<td>38.16 dB 40.41 dB</td>
</tr>
<tr>
<td>Lego truck</td>
<td>28.95 dB 27.86 dB</td>
<td>32.17 dB 38.25 dB</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>27.73 dB 25.92 dB</td>
<td>31.23 dB 35.95 dB</td>
</tr>
</tbody>
</table>

Table 2. PSNR for each recovered light-field using different inpainting methods.
Table 3. PSNR for each light-field using different focal stack reconstruction methods

<table>
<thead>
<tr>
<th>Focal Stack Reconstruction</th>
<th>Scaled Back Projection Preserving</th>
<th>Edge-Preserving SAI UTL</th>
<th>EPI UTL</th>
<th>Proposed LF UTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>amethyst</td>
<td>28.94 dB</td>
<td>29.11 dB</td>
<td>31.57 dB</td>
<td>39.59 dB</td>
</tr>
<tr>
<td>crystal ball</td>
<td>20.33 dB</td>
<td>21.83 dB</td>
<td>30.37 dB</td>
<td>24.73 dB</td>
</tr>
<tr>
<td>Lego bulldozer</td>
<td>23.07 dB</td>
<td>25.36 dB</td>
<td>23.18 dB</td>
<td>30.01 dB</td>
</tr>
<tr>
<td>bunny</td>
<td>29.94 dB</td>
<td>31.57 dB</td>
<td>32.90 dB</td>
<td>39.35 dB</td>
</tr>
<tr>
<td>bracelet</td>
<td>18.23 dB</td>
<td>23.18 dB</td>
<td>26.26 dB</td>
<td>24.51 dB</td>
</tr>
<tr>
<td>eucalyptus</td>
<td>30.40 dB</td>
<td>30.37 dB</td>
<td>24.73 dB</td>
<td>37.56 dB</td>
</tr>
<tr>
<td>Lego knights</td>
<td>21.93 dB</td>
<td>24.72 dB</td>
<td>25.77 dB</td>
<td>32.15 dB</td>
</tr>
<tr>
<td>treasure chest</td>
<td>24.85 dB</td>
<td>25.77 dB</td>
<td>32.90 dB</td>
<td>38.44 dB</td>
</tr>
<tr>
<td>jelly beans</td>
<td>25.08 dB</td>
<td>26.26 dB</td>
<td>32.90 dB</td>
<td>38.42 dB</td>
</tr>
<tr>
<td>Lego truck</td>
<td>24.75 dB</td>
<td>27.45 dB</td>
<td>29.68 dB</td>
<td>38.42 dB</td>
</tr>
<tr>
<td>Average</td>
<td>24.75 dB</td>
<td>27.45 dB</td>
<td>29.68 dB</td>
<td>32.97 dB</td>
</tr>
</tbody>
</table>

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

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.

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 $Z$. As we assume many (but not all) of the rows of $Z$ to be sparse, we believe UTL can be optimized for more efficient storage. An alternative may be to store a subsampled or sketch of $Z$, and only approximate the $W$ 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.