This CVPR workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Compressive Light Field Reconstructions using Deep Learning

Mayank Gupta*
Arizona State University

Arjun Jauhari*
Cornell University

Kuldeep Kulkarni
Arizona State University

Suren Jayasuriya
Carnegie Mellon University

Alyosha Molnar
Cornell University

Pavan Turaga
Arizona State University

Abstract

Light field imaging is limited in its computational processing demands of high sampling for both spatial and angular dimensions. Single-shot light field cameras sacrifice spatial resolution to sample angular viewpoints, typically by multiplexing incoming rays onto a 2D sensor array. While this resolution can be recovered using compressive sensing, these iterative solutions are slow in processing a light field. We present a deep learning approach using a new, two branch network architecture, consisting jointly of an autoencoder and a 4D CNN, to recover a high resolution 4D light field from a single coded 2D image. This network decreases reconstruction time significantly while achieving average PSNR values of 26-32 dB on a variety of light fields. In particular, reconstruction time is decreased from 35 minutes to 6.7 minutes as compared to the dictionary method for equivalent visual quality. These reconstructions are performed at small sampling/compression ratios as low as 8%, allowing for cheaper coded light field cameras. We test our network reconstructions on synthetic light fields, simulated coded measurements of real light fields captured from a Lytro Illum camera, and real coded images from a custom CMOS diffractive light field camera. The combination of compressive light field capture with deep learning allows the potential for real-time light field video acquisition systems in the future.

1. Introduction

Light fields, 4D representations of light rays in unoccluded space, are ubiquitous in computer graphics and vision. Light fields have been used for novel view synthesis [24], synthesizing virtual apertures for images post-capture [26], and 3D depth mapping and shape estimation [35]. Recent research has used light fields as the raw input for visual recognition algorithms such as identifying materials [40]. Finally, biomedical microscopy has employed light field techniques to improve issues concerning aperture and depth focusing [28].

While the algorithmic development for light fields has yielded promising results, capturing high resolution 4D light fields at video rates is difficult. For dense sampling of the angular views, bulky optical setups involving gantries, mechanical arms, or camera arrays have been introduced [44, 37]. However, these systems either cannot operate in real-time or must process large amounts of data, preventing deployment on embedded vision platforms with tight energy budgets. In addition, small form factor, single-shot light field cameras such as pinhole or microlens arrays above image sensors sacrifice spatial resolution for angular resolution in a fixed trade-off [36, 32]. Even the Lytro Illum, the highest resolution consumer light field camera available, does not output video at 30 fps or higher. There is a clear need for a small form-factor, low data rate, cheap light field camera that can process light field video data efficiently.

To reduce the curse of dimensionality when sampling light fields, we turn to compressive sensing (CS). CS states that it is possible to reconstruct a signal perfectly from small number of linear measurements, provided the number of measurements is sufficiently large, and the signal is sparse in a transform domain. Thus CS provides a principled way to reduce the amount of data that is sensed and transmitted through a communication channel. Moreover, the number of sensor elements also reduces significantly, paving a way for cheaper imaging. Recently, researchers introduced *compressive light field photography* to reconstruct light fields captured from coded aperture/mask based cameras at high resolution [30]. The key idea was to use dictionary-based learning for local light field atoms (or patches) coupled with sparsity-constrained optimization to recover the missing information. However, this technique required extensive computational processing on the order of hours for each light field.

In this paper, we present a new class of solutions for the recovery of compressive light fields at a fraction of the time-complexity of the current state-of-the-art, while delivering comparable (and sometimes even better) PSNR. We lever-

*Authors contributed equally to this paper.

age hybrid deep neural network architectures that draw inspiration from simpler architectures in 2D inverse problems, but are redesigned for 4D light fields. We propose a new network architecture consisting of a traditional autoencoder and a 4D CNN which can invert several types of compressive light field measurements including those obtained from coded masks [36] and Angle Sensitive Pixels [39, 15]. We benchmark our network reconstructions on simulated light fields, simulated compressive capture from real Lytro Illum light fields provided by Kalantari *et al*. [18], and real images from a prototype ASP camera [15]. We achieve processing times on the order of a few minutes, which is an order of magnitude faster than the dictionary-based method. This work can help bring real-time light field video at high spatial resolution closer to reality.

2. Related Work

Light Fields and Capture Methods: The modern formulation of light fields were first introduced independently by Levoy and Hanrahan [27] and Gortler *et al*. [14]. Since then, there has been numerous work in view synthesis, synthetic aperture imaging, and depth mapping, see [26] for a broad overview. For capture, gantries or camera arrays [44, 37] provide dense sampling while single-shot camera methods such as microlenses [32], coded apertures [25], masks [36], diffractive pixels [15], and even diffusers [2] and random refractive water droplets [41] have been proposed. All these single-shot methods multiplex angular rays into spatial bins, and thus need to recover that lost information in post-processing.

Light Field Reconstruction: Several techniques have been proposed to increase the spatial and angular resolution of captured light fields. These include using explicit signal processing priors [24] and frequency domain methods [34]. The work closest to our own is compressive light field photography [30] that uses learned dictionaries to reconstruct light fields, and extending that technique to Angle Sensitive Pixels [15]. We replace their framework by using deep learning to perform both the feature extraction and reconstruction with a neural network. Similar to our work, researchers have recently used deep learning networks for view synthesis [18] and spatio-angular superresolution [45]. However, all these methods start from existing 4D light fields, and thus they do not recover light fields from compressed or multiplexed measurements. Recently, Wang *et al*. proposed a hybrid camera system consisting of a DSLR camera at 30 fps with a Lytro Illum at 3fps, and used deep learning to recover light field video at 30 fps [**?**]. Our work hopes to make light field video processing cheaper by decreasing the spatio-angular measurements needed at capture time.

Compressive Sensing: There have been numerous works in compressed sensing [8] resulting in various algorithms to recover the original signal. The classical algorithms [11, 7, 6] rely on the assumption that the signal is sparse or compressible in transform domains like wavelets, DCT, or data dependent pre-trained dictionaries. More sophisticated algorithms include model-based methods [3, 19] and message-passing algorithms [12] which impose a complex image model to perform reconstruction. However, all of these algorithms are iterative and hence are not conducive for fast reconstruction. Similar to our work, deep learning has been used for recovering 2D images from compressive measurements at faster speeds than iterative solvers. Researchers have proposed stacked-denoising autoencoders to perform CS image and video reconstruction respectively [31, 16]. In contrast, Kulkarni *et al*. show that CNNs, which are traditionally used for inference tasks, can also be used for CS image reconstruction [21] . We marry the benefits of the two types of architectures mentioned above and propose a novel architecture to 4D light fields that introduce additional challenges and opportunities for deep learning + compressive sensing.

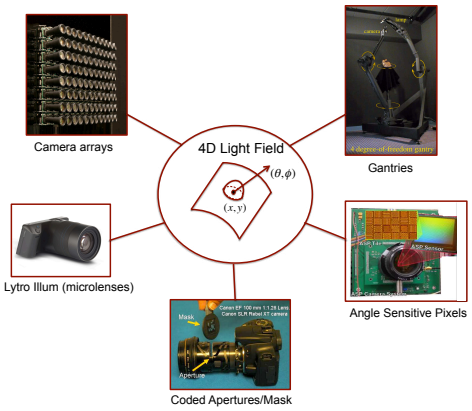

Figure 1. **Light Field Capture:** Light field capture has been performed with various types of imaging systems, but all suffer from challenges with sampling and processing this high dimensional information.

3. Light Field Photography

In this section, we describe the image formation model for capturing 4D light fields and how to reconstruct them.

A 4D light field is typically parameterised with either two planes or two angles [27, 14]. We will represent light fields $l(x, y, \theta, \phi)$ with two spatial coordinates and two angular coordinates. For a regular image sensor, the angular coordinates for the light field are integrated over the main lens, thus yielding the following equation:

$$i(x, y) = \int_{\theta} \int_{\phi} l(x, y, \theta, \phi) d\phi d\theta, \tag{1}$$

where $i(x, y)$ is the image and $l(x, y, \theta, \phi)$ is the light field.

Single-shot light field cameras add a modulation function $\Phi(x, y, \theta, \phi)$ that weights the incoming rays [43]:

$$i(x, y) = \int_{\theta} \int_{\phi} \Phi(x, y, \theta, \phi) \cdot l(x, y, \theta, \phi) d\phi d\theta. \quad (2)$$

When we vectorize this equation, we get $\vec{i} = \Phi \vec{l}$ where the $\vec{l}$ is the vectorized light field, $\vec{i}$ is the vectorized image, and Φ is the matrix discretizing the modulation function. Since light fields are 4D and images are 2D, this is inherently an underdetermined set of equations where Φ has more columns than rows.

The matrix Φ represents the linear transform of the optical element placed in the camera body. This is a decimation matrix for lenslets, comprised of random rows for coded aperture masks, or Gabor wavelets for Angle Sensitive Pixels (ASPs).

3.1. Reconstruction

To invert the equation, we can use a pseudo-inverse $\vec{l} = \Phi^{\dagger} \vec{i}$, but this solution does not recover light fields adequately and is sensitive to noise [43]. Linear methods do exist to invert this equation, but sacrifice spatial resolution by stacking image pixels to gain enough measurements so that Φ is a square matrix.

To recover the light field at the high spatial image resolution, compressive light field photography [30] formulates the following ℓ_1 minimization problem:

$$\min_{\alpha} ||\vec{i} - \Phi D \alpha||_2^2 + \lambda ||\alpha||_1 \quad (3)$$

where the light field can be recovered by performing $l = D\alpha$. Typically the light fields were split into small patches of $9 \times 9 \times 5 \times 5$ (x, y, θ, ϕ) or equivalently sized atoms to be processed by the optimization algorithm. Note that this formulation enforces a sparsity constraint on the number of columns used in dictionary D for the reconstruction. The dictionary D was learned using a set of million light field patches captured by a light field camera and trained using a K-SVD algorithm [1]. To solve this optimization problem, solvers such as ADMM [4] were employed. Reconstruction times ranged from several minutes for non-overlapping patch reconstructions to several hours for overlapping patch reconstructions.

4. Deep Learning for Light Field Reconstruction

We first discuss the datasets of light fields we use for simulating coded light field capture along with our training strategy before discussing our network architecture.

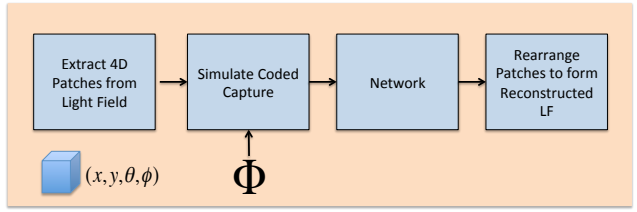

Figure 2. **Pipeline:** An overview of our pipeline for light field reconstruction.

4.1. Light Field Simulation and Training

One of the main difficulties for using deep learning for light field reconstructions is the scarcity of available data for training, and the difficulty of getting ground truth, especially for compressive light field measurements. We employ a mixture of simulation and real data to overcome these challenges in our framework.

Synthetic Light Field Archive: We use synthetic light fields from the Synthetic Light Field Archive [42] which have resolution $(x, y, \theta, \phi) = (593, 840, 5, 5)$. Since the number of parameters for our fully-connected layers would be prohibitively large with the full light field, we split the light fields into $(9, 9, 5, 5)$ patches and reconstruct each local patch. We then stitch the light field back together using overlapping patches to minimize edge effects. This however does limit the ability of our network to use contextual light field information from outside this $(9, 9, 5, 5)$ patch for reconstruction. However, as GPU memory improves with technology, we anticipate that larger patches can be used in the future with improved performance.

Our training procedure is outlined in Figure 2. We pick 50,000 random patches from four synthetic light fields, and simulate coded capture by multiplying by Φ to form images. We then train the network on these images with the labels being the true light field patches. Our training/validation split was 85:15. We finally test our network on a brand new light field never seen before, and report the PSNR as well as visually inspect the quality of the data. In particular, we want to recover parallax in the scenes, i.e. the depth-dependent shift in pixels away from the focal plane as the angular view changes.

Lytro Illum Light Field Dataset: In addition to synthetic light fields, we utilize real light field captured from a Lytro Illum camera [18]. To simulate coded capture, we use the same Φ models for each type of camera and forward model the image capture process, resulting in simulated images that resemble what the cameras would output if they captured that light field. There are a total of 100 light fields, each of size $(364, 540, 14, 14)$. For our simulation purposes, we use only views $[6, 10]$ in both θ and ϕ, to generate $5x5$ angular viewpoints. We extract 500,000 patches from these light fields of size $(9, 9, 5, 5)$, simulate

coded capture, and use a training/validation split of 85:15.

4.2. Network Architecture

Our network architecture consists of a two branch network, which one can see in Figure 3. In the upper branch, the 2D input patch is vectorized to one dimension, then fed to a series of fully connected layers that form a stacked autoencoder (i.e. alternating contracting and expanding layers). This is followed by a 4D convolutional layer. The lower branch is a 4D CNN which uses a fixed interpolation step of multiplying the input image by Φ^T to recover a 4D spatio-angular volume, and then fed through a series of 4D convolutional layers with ReLU nonlinearities. Finally the outputs of the two branches are combined with weights of 0.5 to estimate the light field.

There are several reasons why we converged on this particular network architecture. Autoencoders are useful at extracting meaningful information by compressing inputs to hidden states [38], and our autoencoder branch helped to extract parallax (angular views) in the light field. In contrast, our 4D CNN branch utilizes information from the linear reconstruction by interpolating with Φ^T and then cleaning the result with a series of 4D convolutional layers for improved spatial resolution. Combining the two branches thus gave us good angular recovery along with high spatial resolution (please view the supplemental video to visualize the effect of the two branches). Our approach here was guided by a high-level empirical understanding of the behavior of these network streams, and thus, it is likely to be one of several architecture choices that could lead to similar results. In Figure **??**, we show the results of using solely the upper or lower branch of the network versus our two stream architecture, which helped influence our design decisions. To combine the two branches, we chose to use simple averaging of the two branch outputs. While there may be more intelligent ways to combine these outputs, we found that this sufficed to give us a 1-2 dB PSNR improvement as compared to the autoencoder or 4D CNN alone, and one can observe the sharper visual detail in the inlets of the figure.

For the loss function, we observed that the regular ℓ_2 loss function gives decent reconstructions, but the amount of parallax and spatial quality recovered in the network at the extreme angular viewpoints were lacking. We note this effect in Figure 4. To remedy this, we employ the following weighted ℓ_2 loss function which penalizes errors at the extreme angular viewpoints of the light field more heavily:

$$L(l, \hat{l}) = \sum_{\theta,\phi} W(\theta, \phi) \cdot ||l(x, y, \theta, \phi) - \hat{l}(x, y, \theta, \phi)||_2^2, \quad (4)$$

where $W(\theta, \phi)$ are weights that increase for higher values of θ, ϕ. The weight values were picked heuristically for large weights away from the center viewpoint with the following values: $W(\theta, \phi) =$

$$\begin{pmatrix} \sqrt{5} & 2 & \sqrt{3} & 2 & \sqrt{5} \\ 2 & \sqrt{3} & \sqrt{2} & \sqrt{3} & 2 \\ \sqrt{3} & \sqrt{2} & 1 & \sqrt{2} & \sqrt{3} \\ 2 & \sqrt{3} & \sqrt{2} & \sqrt{3} & 2 \\ \sqrt{5} & 2 & \sqrt{3} & 2 & \sqrt{5} \end{pmatrix}$$

. This loss function gave an average improvement of 0.5dB in PSNR as compared to ℓ_2.

4.2.1 Training Details

All of our networks were trained using Caffe [17] and using a NVIDIA Titan X GPU. Learning rates were set to $\lambda = .00001$, we used the ADAM solver [20], and models were trained for about 60 epochs for 7 hours or so. We also finetuned models trained on different Φ matrices, so that switching the structure of a Φ matrix did not require training from scratch, but only an additional few hours of finetuning.

For training, we found the best performance was achieved when we trained each branch separately on the data, and then combined the branches and jointly finetuned the model further on the data. Training from scratch the entire two branch network led to suboptimal performance of 2-3 dB in PSNR, most likely because of local minima in the loss function as opposed to training each branch separately and then finetuning the combination.

5. Experimental Results

In this section, we show experimental results on both simulated light fields, real light fields with simulated capture, and finally real data taken from a prototype ASP camera [15]. We compare both visual quality and reconstruction time for our reconstructions, and compare against baselines for each dataset.

5.1. Synthetic Experiments

We first show simulation results on the Synthetic Light Field Archive. We used as our baseline the dictionary-based method from [30, 15] with the dictionary trained on synthetic light fields, and we use the dragon scene as our test case. We utilize three types of Φ matrices, a random Φ matrix that represents the ideal 4D random projections matrix (satisfying RIP [5]), but is not physically realizable in hardware (rays are arbitrarily summed from different parts of the image sensor array). We also simulate Φ for coded masks placed in the body of the light field camera, a repeated binary random code that is periodically shifted in angle across the sensor array. Finally, we use the Φ matrix for ASPs which consists of 2D oriented sinusoidal responses to angle as described in [15]. As can be seen in Figure 5, the ASPs

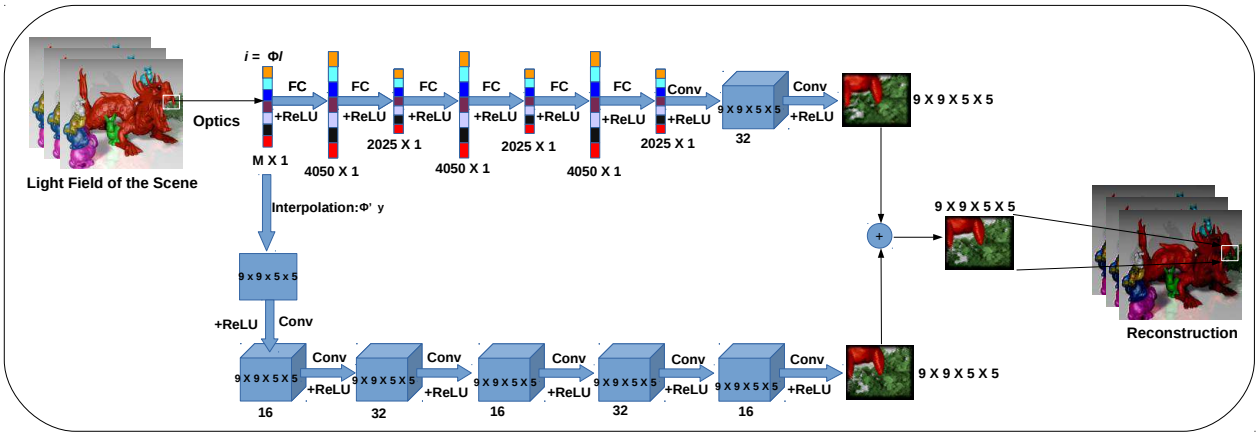

Figure 3. **Network Architecture:** Our two branch architecture for light-field reconstruction. Measurements for every patch of size $(9, 9, 5, 5)$ are fed into two parallel paths, one autoencoder consisting of 6 fully connected followed by one 4D convolution layer, and the other consisting of five 4D convolutional layers. The outputs of the two branches are added with equal weights to obtain the final reconstruction for the patch. Note that the size of filters in all convolution layers is $3 \times 3 \times 3 \times 3$.

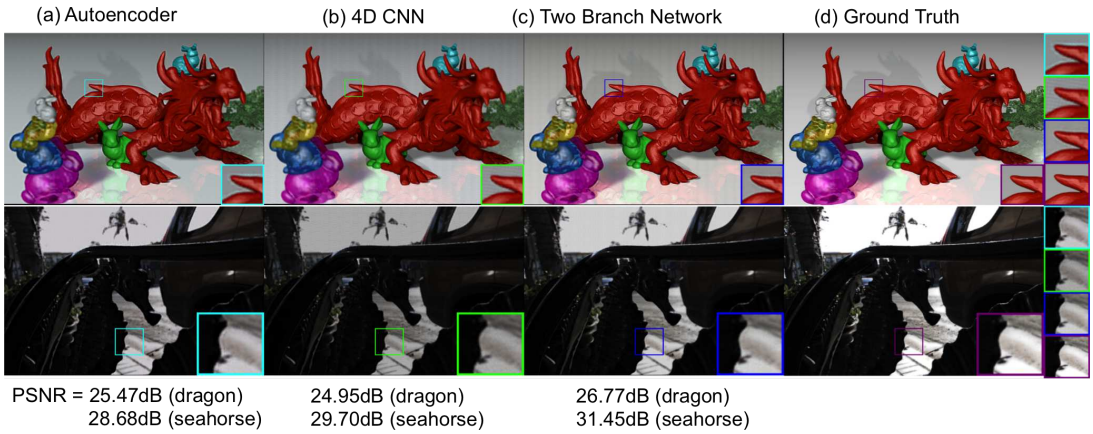

Figure 4. **Branch Comparison:** We compare the results of using only the autoencoder or 4D CNN branch versus the full two branch network. We obtain better results in terms of PSNR for the two-stream network than the two individual branches.

and the mask reconstructions perform slightly better than the ideal random projections. It is hard to justify why ideal projections are not the best reconstruction in practice, but it might be because the compression ratio is too low at 8% for random projections or because there are no theoretical guarantees that the network can solve the CS problem. All the reconstructions do suffer from blurred details in the zoomed inlets, which means that there is still spatial resolution that is not recovered by the network.

Compression ratio is the ratio of independent coded light field measurements to angular samples to reconstruct in the light field for each pixel. This directly corresponds to the number of rows in the Φ matrix which correspond to one spatial location (x, y). We show three separate compression ratios and measure the PSNR for ASP light field cameras in Table 1 with non-overlapping patches. Not surprisingly, increasing the number of measurements increased the PSNR. We also compared for ASPs using our baseline method based on dictionary learning. Our method achieves a 2-4 dB improvement over the baseline method as we vary the number of measurements.

Noise: We also tested the robustness of the networks to additive noise in the input images for ASP reconstruction. We simulated Gaussian noise of standard deviation of 0.1 and 0.2, and record the PSNR and reconstruction time which is display in Table 2. Note that the dictionary-based algorithm takes longer to process noisy patches due to its iterative ℓ_1 solver, while our network has the same flat run

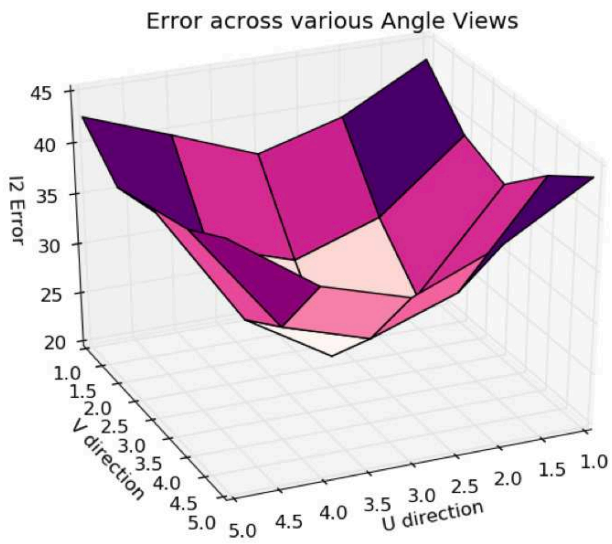

Figure 5. **Error in Angular Viewpoints:** Here we visualize the ℓ_2 error for a light field reconstruction with respect to ground truth using a standard ℓ_2 loss function for training. Notice how the extreme angular viewpoints contain the highest error. This helped motivate the use of a weighted ℓ_2 function for training the network.

Number of Measurements	Our Method (PSNR)	Dictionary Method (PSNR)
N = 2	25.40 dB	22.86 dB
N = 15	26.54 dB	24.40 dB
N = 25	27.55 dB	24.80 dB

Table 1. **Compression sweep:** Variation of PSNR for reconstructions with the number of measurements in the dragons scene for ASP (non-overlapping patches) using the two branch network versus the dictionary method.

Metrics	Noiseless	Std 0.1	Std 0.2
PSNR (Ours) [dB]	26.77	26.74	26.66
PSNR (Dictionary) [dB]	25.80	21.98	17.40
Time (Ours) [s]	242	242	242
Time (Dictionary) [s]	3786	9540	20549

Table 2. **Noise:** The table shows how PSNR varies for different levels of additive Gaussian noise for ASP reconstructions. It is clear that our method is extremely robust to high levels of noise and provides high PSNR reconstructions, while for the dictionary method, the quality of the reconstructions degrade with noise. Also shown is the time taken to perform the reconstruction. For our method, the time taken is only 242 seconds and independent of noise level whereas for dictionary learning method, it can vary from 1 hour to nearly 7 hours.

time regardless of the noise level. This is a distinct advantage of neural network-based methods over the iterative solvers. The network also seems resilient to noise in general, as our PSNR remained about 26 dB.

Lytro Illum Light Fields Dataset: We show our results on this dataset in Figure 6. As a baseline, we compare against the method from Kalantari *et al*. [18] which utilize 4 input views from the light field and generate the missing angular viewpoints with a neural network. Our network model achieves higher PSNR values of 30-32 dB on these real light fields for ASP encoding while keeping the same compression ratio of $\frac{1}{16}$ as Kalantari *et al*. While their method achieves PSNR $>$ 32dB on this dataset, their starting point is 4D light field captured by the Lytro camera and they do not have to uncompress coded measurements. In addition, our method is slightly faster as their network takes 147 seconds to reconstruct the full light field, while our method reconstructs a light field in 80 seconds (both on a Titan X GPU).

5.2. Real Experiments

Finally, to show the feasibility of our method on a real compressive light field camera, we use data collected from a prototype ASP camera [15]. This data was collected on an indoors scene, and utilized three color filters to capture color light fields.

Since we do not have training data for these scenes, we train our two branch network on synthetic data, and then apply a linear scaling factor to ensure the testing data has the same mean as the training data. We also change our Φ matrix to match the actual sensors response and measure the angular variation in our synthetic light fields to what we expect from the real light field. See Figure 7 and the supplementary videos for our reconstructions. We compare our reconstructions against the method from Hirsch *et al*. [15] which uses dictionary-based learning to reconstruct the light fields. For all reconstruction techniques, we apply post-processing filtering to the image to remove periodic artifacts due to the patch-based processing and non-uniformities in the ASP tile, as done in [15].

We first show the effects of stride for overlapping patch reconstructions for the light fields, as shown in Figure 8. Our network model takes a longer time to process smaller stride, but improves the visual quality of the results. This is a useful tradeoff between visual quality of results and reconstruction time in general.

Time complexity and quality of ASP reconstructions: As can be seen, the visual quality of the reconstructed scenes from the network are on-par with the dictionary-based method, but with an order of magnitude faster reconstruction times. A full color light field with stride of 5 in overlapping patches can be reconstructed in 90 seconds, while an improved stride of 2 in overlapping patches yields higher quality reconstructions for 6.7 minutes of reconstruction time. The dictionary-based method in contrast takes 35 minutes for a stride of 5 to process these light fields. However, our method has some distortions in the recovered par-

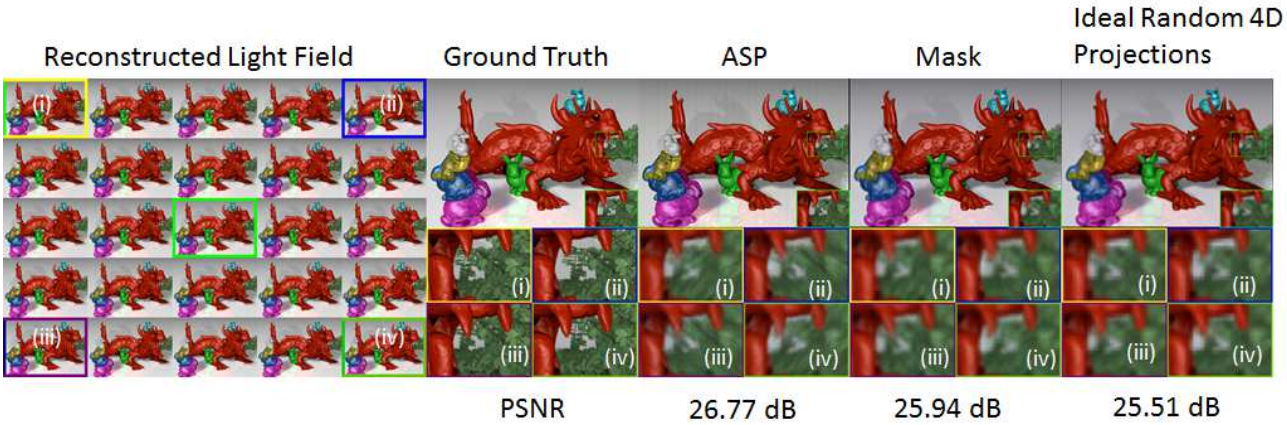

Figure 6. **Different Camera Models:** We compare reconstructions for the dragons scene for different encoding schemes, ASP, Mask and Ideal Random 4D projections (CS) using the two branch network. These reconstructions were done at a low compression ratio of 8% and with a stride of 5. At this low compression ratio, ASPs reconstruct slightly better (26.77 dB) as compared to Masks (25.96 dB) and CS (25.51 dB), although all methods are within 1 dB of each other

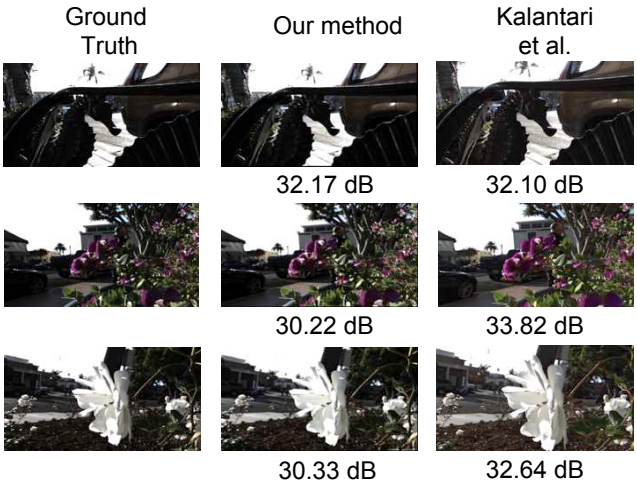

Figure 7. **Lytro Illum Light Fields:** We show reconstruction results for real Lytro Illum light fields with simulated ASP capture. We note that our network performs subpar to Kalantari *et al*. [18] since we have to deal with the additional difficulty of uncompressing the coded measurements.

allax that is seen in the supplementary videos. This could be possibly explained by several reasons. First, optical abberations and mismatch between the real optical impulse response of the system and our Φ model could cause artifacts in reconstruction. Secondly, the loss function used to train the network is the l_2 norm of the difference light field, which can lead to the well-known regress-to-mean effect for the parallax in the scene. It will be interesting to see if a l_1 based loss function or specially designed loss function can help improve the results. Thirdly, there is higher noise in the real data as compared to synthetic data. However, despite these parallax artifacts, we believe the results present here show the potential for using deep learning to recover 4D light fields from real coded light field cameras.

6. Discussion

In this paper, we have presented a deep learning method for the recovery of compressive light fields that is signifcantly faster than the dictionary-based method, while delivering comparable visual quality. The two branch structure of a traditional autoencoder and a 4D CNN lead to superior performance, and we benchmark our results on both synthetic and real light fields, achieving good visual quality while reducing reconstruction time to minutes.

6.1. Limitations

Since acquiring ground truth for coded light field cameras is difficult, there is no possibility of fine tuning our model for improved performance. In addition, it is hard to determine exactly the Φ matrix without careful optical calibration, and this response is dependent on the lens and aperture settings during capture time. All of this information is hard to feed into a neural network to adaptively learn, and leads to a mismatch between the statistics of training and testing data.

6.2. Future Directions

There are several future avenues for research. On the network architecture side, we can explore the use of generative adversarial networks [13] which have been shown to work well in image generation and synthesis problems [33, 23]. In addition, the network could jointly learn optimal codes for capturing light fields with the reconstruction technique, similar to the work by Chakrabarti [9] and Mousavi *et*

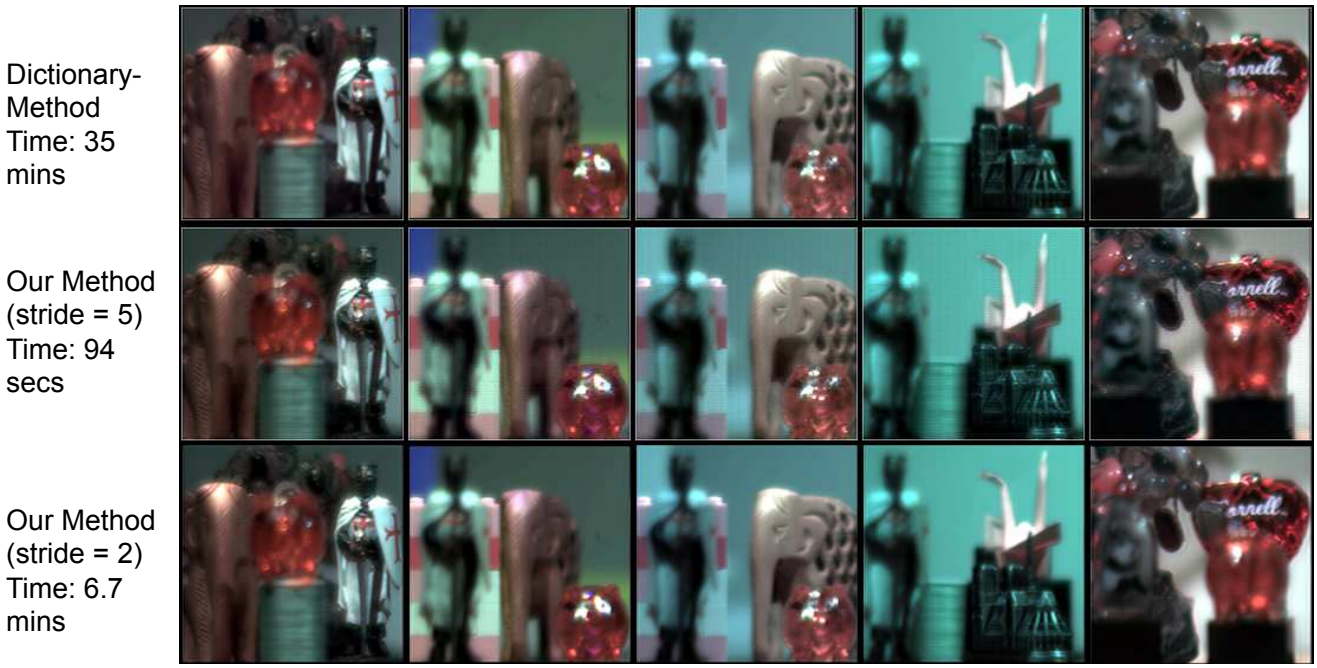

Figure 8. **Real ASP Data:** We show the reconstructions for the real data from the ASP measurements using our method (for stride 5 and stride 2) and dictionary method (for stride 5), and the corresponding time taken. It is clear that the spatial resolution for our method is comparable as that using the dictionary learning method, and the time taken for our method (94 seconds) is an order less than that for the dictionary learning method (35 minutes).

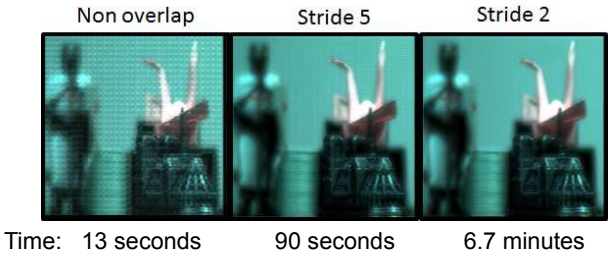

Figure 9. **Overlapping Patches:** Comparison of non-overlapping patches and overlapping patches with strides of 11 (non-overlapping), 5, and 2 for light field reconstructions.

al. [31], helping design new types of coded light field cameras. Finally, we could explore the recent unified network architecture presented by Chang *et al*. [10] that applies to all inverse problems of the form $y = Ax$. While our work has focused on processing single frames of light field video efficiently, we could explore performing coding jointly in the spatio-angular domain and temporal domain. This would help improve the compression ratio for these sensors, and potentially lead to light field video that is captured at interactive (1-15 FPS) frame rates. Finally, it would be interesting to perform inference on compressed light field measurements directly (similar to the work for inference on 2D compressed images [29, 22]) that aims to extract meaningful semantic information. All of these future directions point to a convergence between compressive sensing, deep learning, and computational cameras for enhanced light field imaging.

Acknowledgements: The authors would like to thank the anonymous reviewers for their detailed feedback, Siva Sankalp for running some experiments, and Mark Buckler for GPU computing support. AJ was supported by a gift from Qualcomm. KK and PT were partially supported by NSF CAREER grant 1451263. SJ was supported by a NSF Graduate Research Fellowship and a Qualcomm Innovation Fellowship.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

[2] N. Antipa, S. Necula, R. Ng, and L. Waller. Single-shot diffuser-encoded light field imaging. In *2016 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11. IEEE, 2016.

[3] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the al-

ternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[5] E. J. Candes. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9):589–592, 2008.

[6] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[7] E. J. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, 52(12):5406–5425, 2006.

[8] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

[9] A. Chakrabarti. Learning sensor multiplexing design through back-propagation. In *Advances in Neural Information Processing Systems*, 2016.

[10] J. Chang, C.-L. Li, B. Poczos, B. Vijaya Kumar, and A. C. Sankaranarayanan. One network to solve them all—solving linear inverse problems using deep projection models. *arXiv preprint arXiv:1703.09912*, 2017.

[11] D. L. Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[12] D. L. Donoho, A. Maleki, and A. Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[14] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proc. SIGGRAPH*, pages 43–54, 1996.

[15] M. Hirsch, S. Sivaramakrishnan, S. Jayasuriya, A. Wang, A. Molnar, R. Raskar, and G. Wetzstein. A switchable light field camera architecture with angle sensitive pixels and dictionary-based sparse coding. In *Computational Photography (ICCP), 2014 IEEE International Conference on*, pages 1–10. IEEE, 2014.

[16] M. Iliadis, L. Spinoulas, and A. K. Katsaggelos. Deep fully-connected networks for video compressive sensing. *arXiv preprint arXiv:1603.04930*, 2016.

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[18] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016)*, 35(6), 2016.

[19] Y. Kim, M. S. Nadar, and A. Bilgin. Compressed sensing using a Gaussian scale mixtures model in wavelet domain. pages 3365–3368. IEEE, 2010.

[20] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[21] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[22] K. Kulkarni and P. Turaga. Reconstruction-free action inference from compressive imagers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 38(4):772–784, 2016.

[23] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[24] A. Levin and F. Durand. Linear view synthesis using a dimensionality gap light field prior. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1831–1838. IEEE, 2010.

[25] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)*, 26(3):70, 2007.

[26] M. Levoy. Light fields and computational imaging. *IEEE Computer*, 39(8):46–55, 2006.

[27] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. SIGGRAPH*, pages 31–42, 1996.

[28] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz. Light field microscopy. *ACM Transactions on Graphics (TOG)*, 25(3):924–934, 2006.

[29] S. Lohit, K. Kulkarni, P. Turaga, J. Wang, and A. Sankaranarayanan. Reconstruction-free inference on compressive measurements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–24, 2015.

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

[31] A. Mousavi, A. B. Patel, and R. G. Baraniuk. A deep learning approach to structured signal recovery. In *Communication, Control, and Computing (Allerton), 2015 53rd Annual Allerton Conference on*, pages 1336–1343. IEEE, 2015.

[32] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11), 2005.

[33] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. *CVPR*, 2016.

[34] L. Shi, H. Hassanieh, A. Davis, D. Katabi, and F. Durand. Light field reconstruction using sparsity in the continuous fourier domain. *ACM Transactions on Graphics (TOG)*, 34(1):12, 2014.

[35] M. W. Tao, P. P. Srinivasan, S. Hadap, S. Rusinkiewicz, J. Malik, and R. Ramamoorthi. Shape estimation from shading, defocus, and correspondence using light-field angular coherence. *IEEE transactions on pattern analysis and machine intelligence*, 39(3):546–560, 2017.

[36] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Graph. (SIGGRAPH)*, 26(3):69, 2007.

[37] K. Venkataraman, D. Lelescu, J. Duparré, A. McMahon, G. Molina, P. Chatterjee, R. Mullis, and S. Nayar. Picam: an ultra-thin high performance monolithic camera array. *ACM Trans. Graph. (SIGGRAPH Asia)*, 32(6):166, 2013.

[38] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[39] A. Wang and A. Molnar. A light-field image sensor in 180 nm cmos. *Solid-State Circuits, IEEE Journal of*, 47(1):257–271, 2012.

[40] T.-C. Wang, J.-Y. Zhu, E. Hiroaki, M. Chandraker, A. A. Efros, and R. Ramamoorthi. A 4d light-field dataset and cnn architectures for material recognition. In *European Conference on Computer Vision*, pages 121–138. Springer International Publishing, 2016.

[41] A. Wender, J. Iseringhausen, B. Goldlücke, M. Fuchs, and M. B. Hullin. Light field imaging through household optics. In D. Bommes, T. Ritschel, and T. Schultz, editors, *Vision, Modeling & Visualization*, pages 159–166. The Eurographics Association, 2015.

[42] G. Wetzstein. Synthetic light field archive. http://web.media.mit.edu/ gordonw/SyntheticLightFields/,.

[43] G. Wetzstein, I. Ihrke, and W. Heidrich. On Plenoptic Multiplexing and Reconstruction. *IJCV*, 101:384–400, 2013.

[44] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM Trans. Graph. (SIGGRAPH)*, 24(3):765–776, 2005.

[45] Y. Yoon, H.-G. Jeon, D. Yoo, J.-Y. Lee, and I. So Kweon. Learning a deep convolutional network for light-field image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 24–32, 2015.