

# Harnessing Multi-View Perspective of Light Fields for Low-Light Imaging

Mohit Lamba\*, Kranthi Kumar Rachavarapu\* and Kaushik Mitra

<https://mohitlamba94.github.io/L3Fnet/>

**Abstract**—Light Field (LF) offers unique advantages such as post-capture refocusing and depth estimation, but low-light conditions severely limit these capabilities. To restore low-light LFs we should harness the geometric cues present in different LF views, which is not possible using single-frame low-light enhancement techniques. We, therefore, propose a deep neural network architecture for Low-Light Light Field (L3F) restoration, which we refer to as L3Fnet. The proposed L3Fnet not only performs the necessary visual enhancement of each LF view but also preserves the epipolar geometry across views. We achieve this by adopting a two-stage architecture for L3Fnet. Stage-I looks at all the LF views to encode the LF geometry. This encoded information is then used in Stage-II to reconstruct each LF view.

To facilitate learning-based techniques for low-light LF imaging, we collected a comprehensive LF dataset of various scenes. For each scene, we captured four LFs, one with near-optimal exposure and ISO settings and the others at different levels of low-light conditions varying from low to extreme low-light settings. The effectiveness of the proposed L3Fnet is supported by both visual and numerical comparisons on this dataset. To further analyze the performance of low-light reconstruction methods, we also propose an L3F-wild dataset that contains LF captured late at night with almost zero lux values. No ground truth is available in this dataset. To perform well on the L3F-wild dataset, any method must adapt to the light level of the captured scene. To do this we use a pre-processing block that makes L3Fnet robust to various degrees of low-light conditions. Lastly, we show that L3Fnet can also be used for low-light enhancement of single-frame images, despite it being engineered for LF data. We do so by converting the single-frame DSLR image into a form suitable to L3Fnet, which we call as *pseudo-LF*.

**Index Terms**—Low-Light, Light Field enhancement, Light Field dataset.

## I. INTRODUCTION

**I**MAGES captured in low-light such as in the dark or nighttime, not only lack the pleasing visual aesthetics but are also corrupted by high amount of noise and color distortions. This forthrightly hinders the performance of many computer vision algorithms [1]. This plight of low-light conditions is not unique to conventional 2D photographs, but is also shared by Light Fields (LFs) [2]–[5]. In contrast with a conventional camera, which captures only 2-D spatial information, a light field camera captures both 2-D spatial and 2-D angular

Mohit Lamba, Kranthi Kumar Rachavarapu and Kaushik Mitra are with the Department of Electrical Engineering, Indian Institute of Technology Madras, India (e-mail: ee18d009@smail.iitm.ac.in, ee18d004@smail.iitm.ac.in, kmitra@ee.iitm.ac.in).

This paper has a supplementary downloadable PDF report available at <http://ieeexplore.ieee.org>, provided by the author. Contact ee18d009@smail.iitm.ac.in for further questions about this work.

\*M. Lamba and K. Kumar have contributed equally to the work

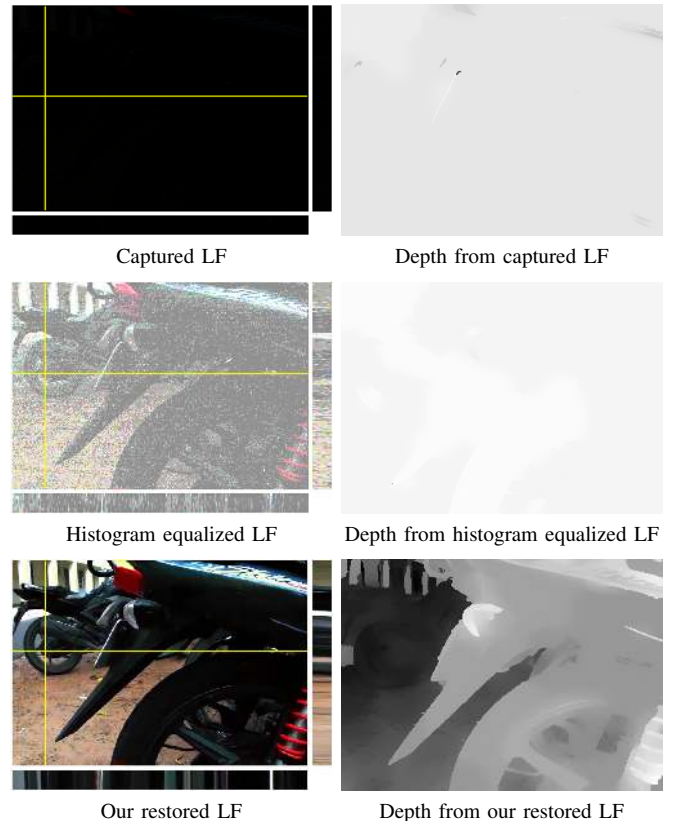


Fig. 1. Low-light is a severe bottleneck to Light Field (LF) applications. For example, the depth estimate of LF captured in low light is very poor. Our proposed method not only visually restores each of the LF views but also preserves the LF geometry for faithful depth estimation.

information about the scene. Capturing the full light field allows us to perform post-capture controls such as digital refocusing and aperture control [3], [6]. It also enables easy scene depth estimation [7]–[9]. This has resulted in many LF applications such as view synthesis [4], [5], [10], structure from motion [11], pedestrian identification [12], reflection removal [13], and various other real-world application like autonomous driving and plant monitoring [14]. But, as stated previously, these LF applications are not immune to the challenges of low-light imaging. This is depicted in Fig. 1 where the depth estimation capability of LF is foiled due to low-light conditions. Commonly used techniques such as histogram equalization or having higher ISO does not help much in this regard as they boost the noise levels and introduce unwanted artifacts. Our goal, therefore, is to design a low-light LF restoration technique to mitigate these problems.

Low-light restoration is an ill-posed problem because of the large amount of noise present in the low-light signal. Also, color information is not present adequately. An LF, however, by capturing multiple views of the low-light scene, contains rich geometric cues about the scene. Harnessing the complimentary information spread across the LF views should help in reducing the ill-posedness of the problem and result in better visual reconstruction. In addition to this, the restoring process should also preserve the LF epipolar constraints to allow for subsequent tasks like depth estimation. The existing works on the low-light enhancement [15]–[20] are, however, not designed to keep these points in consideration and so are not suitable candidates for enhancing low-light LFs.

We propose a two-stage deep neural network for Low-Light Light Field (L3Fnet) reconstruction. Stage-I of the L3Fnet operates on the full LF to first encode the LF geometry. This encoded information is then used as an auxiliary information in Stage-II for actual view restoration. Adopting such a two-staged architecture helps us to restore low-light LFs both aesthetically and with geometric correctness.

Training a learning-based model such as L3Fnet requires a low-light LF dataset but, unfortunately, there is no such dataset. An easy way out would have been to create a synthetic dataset using gamma correction, noise addition, or even retouching the images in software such as Adobe Photoshop and GIMP as was done for the single frame case [16], [17], [20]–[22]. But these are only proxy solutions, and as pointed out by Plötz and Stefan [23], benchmarking algorithms on synthetic dataset may not correlate with their performance on real-world data. We, therefore, collected our own Low-Light Light Field (L3F) dataset using commercially available Lytro Illum LF camera. L3F dataset was captured in the evenings when the visibility was below normal level. For each scene, 4 shots were taken. For the first shot, optimal camera exposure and ISO settings were used to get the ground truth LF. Lytro Illum’s exposure was then reduced in definite proportions to capture three more low-light LFs. While taking these shots, much care has been taken to avoid moving objects such as passing vehicles and wavering leaves to prevent any registration issues. Camera shake is another cause of incurring registration problems and is much more pronounced for Lytro Illum. Lytro Illum, unlike modern single-frame DSLR cameras, does not allow remote connectivity for capturing data. So the only way to capture LF was by physically pressing the shutter button causing unintentional camera shake. To limit this we captured multiple sets of four LFs for each scene and employed additional measures to keep the camera rigidly fixed. We then did a manual check to identify the set exhibiting the least amount of camera shake. Using these techniques the maximum pixel shift in the captured LFs is about 3–4 pixels.

While capturing the L3F dataset we were constrained to capture the ground truth for each scene. This was essential to train our network. Because of this constraint, we were unable to capture extremely low-light LFs. We, therefore, decided to forsake this restriction and collected a separate low-light LF dataset captured late in the night with near *zero* lux conditions at the camera lens. We call this dataset L3F-wild, which we use only for evaluation. L3F-wild LFs were captured

with typical camera exposure of 1/5 second and nominal ISO levels. Note that we do not use the standard practice of capturing images with a long exposure, of say 10 seconds, in low-light conditions. This is to avoid possible motion blur and is a step towards fast low-light imaging. The L3F-wild dataset has a good amount of variation in scene brightness level, portraying a real-world scenario. We, consequently, use a pre-processing block which when appended to L3Fnet makes it robust to such variations in light levels by automatically estimating an appropriate amplification factor.

We have already discussed why single-frame techniques are not suitable for enhancing low-light LF. And we now address the reverse question, *can LF methods be used for single-frame DSLR images?* The proposed L3Fnet has been specifically engineered for LF and so cannot be directly operated on single-frame images. However, we propose a novel pixel shuffling mechanism, which can convert any DSLR image into a *pseudo-LF*. Pseudo-LF has a form suitable to L3Fnet, and so can be enhanced using it. Later on, the enhanced pseudo-LF is transformed back into a single-frame DSLR image in a lossless fashion. This gives L3Fnet architecture a universal appeal for both LF and single-frame image low-light enhancement with restoration being more optimized for LF but maintaining a decent recovery for single-frame images.

To summarize, we make the following contributions:

- We propose a two-stage deep neural network, L3Fnet, for restoring extremely low-light LFs.
- We collected L3F dataset consisting of real LFs with varying levels of low-light, which can be used for training and evaluation of data driven methods.
- We use a pre-processing block that automatically adapts L3Fnet to changing light levels.
- Our proposed Pseudo-L3Fnet framework enables L3Fnet to process even single-frame DSLR images for better enhancement in several cases.

## II. RELATED WORK

**LF processing algorithm:** Many techniques and models have been proposed for several LF related tasks. This includes tasks such as spatial super-resolution [24]–[27], deblurring [28]–[30], denoising [31]–[35], and depth estimation [7]–[9]. But, to the best of our knowledge, no prior work has considered solving the challenges involved in low-light LF imaging. We, therefore, outline some of the important works on LF denoising as denoising is a crucial part of low-light restoration. The easiest approach to LF denoising is to individually denoise each LF view using standard techniques like BM3D [36]. This, however, does not capture the 4D structure of LF and was addressed by Mitra and Veeraraghavan [31]. By operating on LF patches, they modeled each 4D patch using Gaussian Mixture Models (GMM) subject to the patch disparity information. This way they provided a common framework for several LF tasks such as super-resolution and denoising. Dansereau *et al.* [33] observed that the LF of a Lambertian surface has a hyperfan-like shape in the frequency domain. By appropriately choosing the passband they tried to remove noise. Sepas-Moghaddam *et al.* [34] treating LF akin to video frames,



Fig. 2. Sample center-view images from the Low-Light Light Field (L3F) dataset consisting of 27 scenes. For each scene, we capture a LF at near optimal exposure setting and at three other exposure settings with the exposures being  $1/20^{th}$ ,  $1/50^{th}$  and  $1/100^{th}$  of the optimal setting. Refer supplementary material to view the full dataset.

first converted LF into an epipolar sequence and then applied video-based techniques to denoise the LF. This 3D stacking of epipoles, however, has limitations in capturing the 4D nature of LF and was consequently addressed by LFBM5D [32], a popular LF denoising technique. LFBM5D does a realistic 4D LF modeling and is a natural extension to the then state-of-the-art denoising method BM3D. However, LFBM5D does not address the problem with low-light LFs because of its inability to enhance color.

**Single-frame low-light enhancement:** A significant amount of literature exists for low-light single-frame image enhancement. We briefly review some of them. LIME [15] used a non deep learning optimization framework for low-light enhancement. It utilized the retinex theory [37] to decouple the captured image into reflectance and illumination components for subsequent enhancement. A similar idea was used by Park *et al.* [16] and proposed another variational optimization-based retinex model. Ying *et al.* [38] also used the retinex model but combined it with the camera response model to preserve the naturalness of the enhanced image. Deep learning based methods employing encoder-decoder architecture have also been recently used for low-light enhancement [17]–[20]. Amongst all these recent works, the work by Chen *et al.* [19] is a landmark paper on extreme low-light enhancement and is closest to our work. Following this work by Chen *et al.*, [39], [40] are some of the more recent methods which show results on extreme low-light images. Other recent works [17], [18], [20], [41]–[46] also aim at low-light but their main objective is the enhancement of dim images. These images already had a good representation of the target scene and only lacked in aesthetics and contrast but had a decent amount of scene visibility. This is an interesting problem to solve but in our work we target image with much lower visibility, see Fig. 1, 6 and 15.

### III. L3F DATASET

Creating a low-light LF enhancement dataset is a major challenge in designing learning-based solutions. Synthetic

low-light LF data can be created using gamma correction and noise addition [16], [20]–[22], however, such techniques act globally and so do not mimic a real low-light situation which severely affects some regions more than others. Also, in synthetic data noise is generally modeled as Gaussian or Poisson distribution which may not hold true in real low-light data. The other popular technique is to retouch the low-light images in photo-editing software by trained photographers to obtain the ground-truth [17], [18]. Such proxy solutions are not suitable for evaluating low-light LFs because any local edit in a LF view needs to be propagated to all the other SAIs which is difficult to enforce manually. Thus, different from recent methods that adopt such techniques, we introduce a Low-Light Light Field (L3F) dataset containing both low-light LF and the corresponding ground-truth LF. To the best of our knowledge, this is the first dataset available for training and benchmarking low-light LF enhancement techniques.

All the LFs in the proposed L3F dataset were captured in outdoor situations. The scene content predominantly includes an outdoor urban campus environment with lambertian and non-lambertian surfaces at varying depth and occlusion levels. The captured LFs do not include any dynamic objects in the scene such as moving people and vehicles. All the LFs were captured in the evening with limited natural light where the typical illuminance was between 0.1 Lux to 20 Lux.

We use the Lytro Illum Camera to capture all the LFs. For each scene we captured four LF data. One of them was captured using the least ISO and appropriate exposure settings to make it look like a well-lit image. This we use as the ground truth reference LF for that scene. The other three LFs were captured by reducing the exposure setting to  $1/20^{th}$ ,  $1/50^{th}$  and  $1/100^{th}$  of the reference LF exposure time. For brevity, we refer them as L3F – 20, 50 and 100 datasets. Chen *et al.* [19] limited their work to approximately  $1/20^{th}$  setting but we go even more low-light with L3F-100 dataset for better understanding and contrasting the effect of low-light on LF reconstruction.

Motion blur caused due to camera shake is a common artifact when capturing long-exposure sequences. To limit this we mount the Lytro camera on tripod and capture the LFs using timer mode. Unlike DSLR cameras the Lytro Illum camera does not allow remote capture which makes the data capturing process harder and laborious. Consequently, the four LFs captured for the same scene exhibits small spatial misalignment. To contain this we capture multiple sets of these four LFs for a target scene and choose the set exhibiting the least alignment problem. We could have further used matching algorithms, but they had a lot of difficulty in finding the correct correspondences between well-lit and extremely dark images. Thus, no subsequent alignment operation was performed. In the supplementary we quantitatively show that for most scenes the misalignment is  $\leq 1$  pixel. This small misalignment in L3F dataset is taken care by our proposed method.

The images were captured in the Light Field Raw (LFR) format of the Lytro Illum camera. The resolution of each LFR file is  $5368 \times 7728$  pixels. The raw images are captured in the Bayer sensor pattern where the pixels are hexagonally packed. We use Light Field Matlab Toolbox [35] to demosaic, decode,

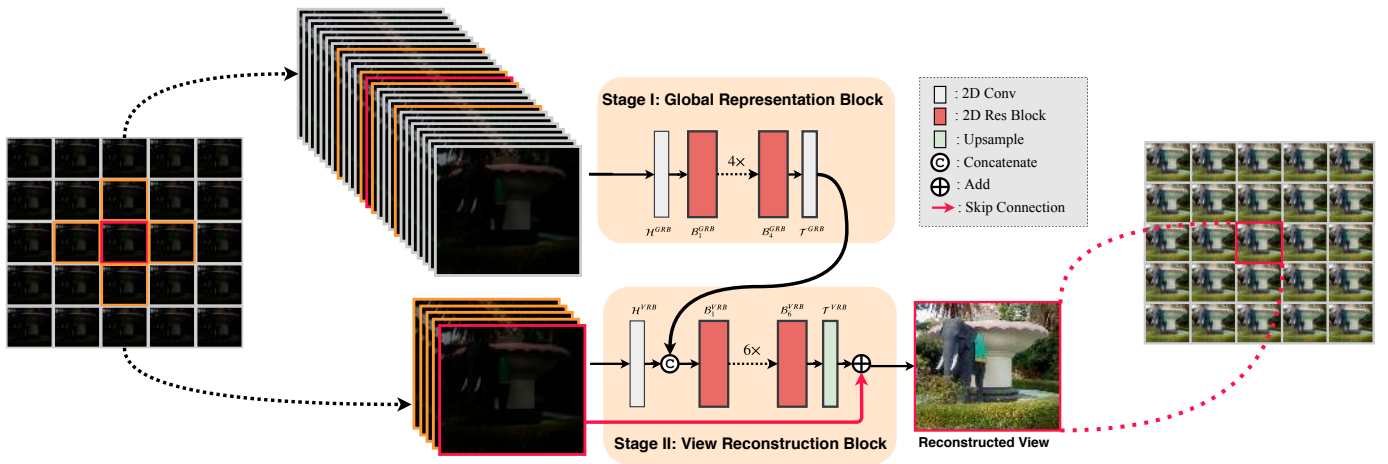


Fig. 3. L3Fnet architecture: The proposed architecture consists of a Global Representation Block (Stage-I) and a View Reconstruction block (Stage-II). Stage-I operates on the full low-light LF to obtain a latent representation that encodes the LF geometry. This latent representation is then used by Stage-II to restore each LF view.

devidnetize and color-correct the LFR files. The decoded 4D LF data have a resolution of  $15 \times 15 \times 434 \times 625 \times 3$ , where  $15 \times 15$  represents the angular resolution,  $434 \times 625$  represents the spatial resolution of each view and 3 corresponds to the RGB channels.

The proposed dataset contains a total of 108 LFs organized into 27 sets. Each set corresponds to a unique scene with 4 LFs captured at various exposure settings. Fig. 2 shows a few sample LF center-views from the proposed dataset. 33% of the dataset i.e., 9 sets (36 LFs) forms the test set. The dataset is available for download from the [project page](#).

While capturing L3F-20, 50, and 100 datasets, we were constrained to capture the well-lit LF image also, which is required for training the L3Fnet. Forsaking this restriction, we captured even darker low-light LFs taken late in the night. While capturing this dataset, the Lux measure at Lytro Illum’s lens was almost nil, and so no ground truth was possible for these LF images. We call this dataset *L3F-wild*. Although L3F-wild cannot be used for training, the performance of methods trained on L3F-20, 50, and 100 dataset can be checked by evaluating on L3F-wild dataset. L3F-wild dataset is a step forward towards fast low-light imaging because it does not adopt the standard practices of low-light imaging such as prolonged exposure or high ISO. Long exposures like 5 – 10 seconds cause a lot of motion blur, and high ISO boosts the noise. L3F-wild was, however, captured with a typical exposure time of  $1/4 - 1/15$  second and low ISO values around 100. In the experimental section and supplementary, we demonstrate the effectiveness of L3Fnet on both L3F-100 and L3F-wild datasets.

#### IV. LOW-LIGHT LIGHT FIELD NETWORK (L3FNET)

##### A. Important Features of L3Fnet

Our proposed method is designed to enhance LFs captured in low-light. But before describing the details of the proposed solution, we highlight some of the desired characteristics that the proposed solution should have. These characteristics are not task-specific but essential to any LF architecture. These features were cataloged after studying different architectures

for various LF related tasks. Firstly, L3Fnet should *not make strong assumptions* about the scene, so that, given any low-light LF we should be able to restore it. This is contrary to works like [24], [47], which required explicit geometric information of the scene, such as depth-map, to perform the given task. Secondly, L3Fnet should not restore the LF views independent of each other. This may lead to visually pleasing reconstruction, but the LF epipolar geometry would not be preserved. This was noticed in works such as [27], where only pairs of LF views were fed to the CNN model, failing to capture the high *view coherence* of the LF [25]. The third desired characteristic is *view parallelization*, which is the ability to restore each view parallelly. This is contrary to the approach taken by LFNet [25], which used bidirectional recurrent units to model the inter-view dependencies. But as pointed out in [26], the dependencies were not modeled well, and the algorithm was slow because of sequential processing over time. So if possible, recurrent units should not be used in L3Fnet for potential *view parallelization*. View parallelization also has the advantage that we can selectively reconstruct only some of the desired LF, without reconstructing the full Light Field, saving time and memory. Lastly, we do not want different model architectures with multiple sets of weights for different views. This was adopted by [26] to retrain their network for each Sub-Aperture Image (SAI) and consequently have a separate set of weights for each LF view. We, on the other hand, would like to have a *single architecture with shared weights* for all LF views. We have summarized this discussion in Table. I.

We have incorporated these features in the design of L3Fnet, see Fig. 3, to the extent possible without harming the primary purpose of enhancing low-light LFs. L3Fnet admits no strong assumption about the target scene because it requires no extra information other than the low-light LF for enhancement. To achieve view-coherence, we have a Global Representation Block (GRB), which operates on the entire LF to obtain a latent representation that encodes the LF epipolar geometry. The View Reconstruction Block (VRB) of L3Fnet then uses the global information encoded by GRB to restore each LF

TABLE I  
COMPARISON OF ARCHITECTURAL FEATURES PRESENT IN L3FNET.

	No strong priors	View coherence	View Parallelization	Shared weights & architecture
Wanner <i>et al.</i> [24]	✗	✗	✓	✓
Yoon <i>et al.</i> [27]	✓	✗	✓	✓
LFNet [25]	✓	✓	✗	✓
Zhang <i>et al.</i> [26]	✓	✓	✓	✗
Ours L3Fnet	✓	✓	✓	✓

view independent of other LF view. Thus we incorporate the view parallelization property.

Splitting the L3Fnet architecture into two stages is very beneficial. If all the SAIs are pushed into a CNN model, the model may be hard to train and test because of huge computational requirements. This is because LFs are much larger than even HD DSLR images. We, therefore, decomposed the L3Fnet architecture into two stages. Stage-I looks at all the SAIs and provides a latent representation to Stage-II, to aid individual restoration of each SAI. This architecture thereby prevents the parameter and computational complexity explosion.

Restoring low-light light fields subsumes the task of denoising, color restoration and preserving the LF geometry. The proposed GRB leverages information from multiple views to aid denoising and in preserving high-frequency details and the LF geometry. We shall verify these claims in the experimental section where we show that by using GRB depth estimation is benefitted, PSNR/SSIM improves, and the high-frequency details are preserved well.

### B. Network Architecture

The architecture of L3Fnet is shown in Fig. 3. We now describe each of its component in detail below.

**Stage-I Global Representation Block (GRB):** Given an input LF  $\mathcal{L}^{low} \in \mathbf{R}^{U \times V \times W \times H \times 3}$ , we first stack all the views across channels to obtain  $\hat{\mathcal{I}} \in \mathbf{R}^{W \times H \times 3UV}$ . Such a representation can be processed using a 2D Convolutional Neural Network (CNN) to obtain a low-dimensional LF representation useful for any down-stream task.

In the Global Representation Block (GRB) a convolutional layer  $\mathcal{H}^{GRB}$  is used to reduce the input channel dimensions of  $\hat{\mathcal{I}}$ , see Table II.

$$J_0 = \mathcal{H}^{GRB}(\hat{\mathcal{I}}) \quad (1)$$

To now extract useful information, we further process this representation using  $M$  (fixed at 4) residual blocks [48] to obtain the global representation

$$J_m = \mathcal{B}_m^{GRB}(J_{m-1}), \quad m \in \{1, \dots, M\}, \quad (2)$$

where  $\mathcal{B}_m^{GRB}$  denotes the  $m^{th}$  residual block in the GRB. This feature map is then fed to the final convolutional layer  $\mathcal{T}^{GRB}$  as a post-processing step

$$J_{M+1} = \mathcal{T}^{GRB}(J_M). \quad (3)$$

By processing all the views together using a CNN architecture the network captures the implicit LF structure such

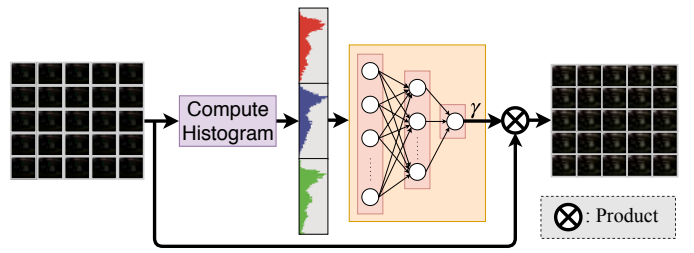


Fig. 4. The Histogram Module computes the RGB histogram of a low-light LF and outputs an amplification factor  $\gamma$ . Normalizing the dark LF with this module allows L3Fnet to process images of varying low-light levels mitigating the over/under saturation problem.

as disparity, sub-pixel information etc., relevant for the final task at hand. This global representation is then used by the view reconstruction block as an augmented information to reconstruct any of the input view.

**Stage-II View Reconstruction Block (VRB):** While the GRB representation is too coarse and common to all SAIs, for better reconstruction of each SAI we explicitly use its immediate neighbours. Formally, to reconstruct a particular view  $\mathcal{I}(u, v) \in \mathbf{R}^{W \times H \times 3}$ , its neighbours  $\{\mathcal{I}(u, v-1), \mathcal{I}(u-1, v), \mathcal{I}(u, v+1), \mathcal{I}(u+1, v)\}$  are stacked across channels to obtain  $\tilde{\mathcal{I}} \in \mathbf{R}^{W \times H \times 15}$ . The view restoration process begins by processing  $\tilde{\mathcal{I}}$  using a convolutional layer  $\mathcal{H}^{VRB}$  to obtain the feature map  $C_0$ :

$$C_0 = \mathcal{H}^{VRB}(\tilde{\mathcal{I}}). \quad (4)$$

The global feature map is concatenated with this before being fed to  $N$  (fixed at 6) residual blocks:

$$C_1 = \mathcal{B}_1^{VRB}(C_0 \oplus J_{M+1}) \quad (5)$$

$$C_n = \mathcal{B}_n^{VRB}(C_{n-1}), \quad n \in \{2, \dots, N\}, \quad (6)$$

where  $\mathcal{B}_n^{VRB}$  denotes the  $n^{th}$  resblock in the view reconstruction block and  $\oplus$  denotes the concatenation operation. The resblocks in GRB and VRB branches share the same structure. Finally, the output feature map is passed through a transposed convolution block  $\mathcal{T}^{VRB}$  followed by a long skip connection from the input SAI to restore the SAI

$$I_{out}(u, v) = \mathcal{T}^{VRB}(C_N) + I(u, v). \quad (7)$$

Many works [49], [50] have reported difficulty in training very deep networks by simply stacking residual blocks. Therefore, to stabilize the training and optimization of deep networks, adding a long/global skip connection has become a standard practice [49], [51]. Further, unlike the short skip connections which help in propagating finer details, long skip connection helps to transmit coarse level details which are crucial for image restoration tasks.

**Histogram Module:** The typical illuminance of the scenes in our dataset varies from 0.1 lux–20 lux. This causes a lot of variation in the input data distribution. A simple workaround is to scale the input image with an appropriate amplification factor  $\gamma$  which controls the brightness of the image. Chen *et al.* [19] requires a manual input for  $\gamma$ . Contrary to this, we automate the process by pre-processing the low-light LF with the Histogram Module shown in Fig. 4.

TABLE II

L3FNET ARCHITECTURE SUMMARY.  $K$  STANDS FOR KERNEL SIZE,  $S$  FOR STRIDE, OUT FOR NUMBER OF CHANNELS IN CONVOLUTIONAL LAYERS.

Branch	Name	Type	K	S	Out
Stage-I Global Representation Block	$\mathcal{H}^{GRB}$	Conv-2D	7	1	64
		Conv-2D	3	2	128
	$\mathcal{B}^{GRB}$	Res-Block	3	1	128
		Res-Block	3	1	128
		Res-Block	3	1	128
		Res-Block	3	1	128
$\mathcal{T}^{GRB}$	Conv-2D	1	1	64	
Stage-II View Reconstruction Block	$\mathcal{H}^{VRB}$	Conv-2D	7	1	15
		Conv-2D	3	2	64
	$\mathcal{B}^{VRB}$	Res-Block	3	1	128
		Res-Block	3	1	128
		Res-Block	3	1	128
		Res-Block	3	1	128
		Res-Block	3	1	128
		Res-Block	3	1	128
	$\mathcal{T}^{VRB}$	Transpose-2D	2	2	128
		Conv-2D	3	1	3
Histogram Module	$\mathcal{H}$	FC	-	-	200
		FC	-	-	100
		FC	-	-	50
		FC	-	-	1

To estimate the appropriate amplification factor  $\gamma$ , we make use of the RGB histogram of the input low-light LF image  $\mathcal{L}^{low}$ . Formally, let  $h^c \in \mathbf{R}^L$  be the normalized  $L$ -bin histogram of the input image corresponding to the color channel  $c \in \{R, G, B\}$ . This histogram is then fed to 3 fully connected layers  $\mathcal{H}$  to compute the amplification factor  $\gamma$ .  $\gamma$  is then used to appropriately scale the input low-light LF:

$$\gamma = \mathcal{H}(h^R \oplus h^G \oplus h^B) \quad (8)$$

$$\mathcal{L}^{norm} = \gamma \times \mathcal{L}^{low}, \quad (9)$$

where  $\oplus$  denotes concatenation. During the training phase, we only need to update the weights of the MLP to obtain the required  $\gamma$  and this requires back-propagation to happen only till the first layer of MLP and not beyond. Scaling the input image with the amplification factor acts as an important pre-processing step. Without this pre-scaling step the restored LF would sometimes tend to be over or under-saturated for light levels of different gradations.

**Loss Function:** For the proposed solution we try to minimise the following loss function:

$$\begin{aligned} Loss = & \alpha_1 \sum_{k=1}^K \|I^{out}(u_k, v_k) - I^{GT}(u_k, v_k)\|_1 \\ & + \alpha_2 \sum_{k=1}^K CX(I^{out}(u_k, v_k), I^{GT}(u_k, v_k)) \\ & + \lambda \|w\|_1, \end{aligned} \quad (10)$$

where  $CX(\cdot, \cdot)$  is the contextual loss [52],  $w$  denotes the model parameters and  $I^{GT}(u_k, v_k)$  is the ground-truth

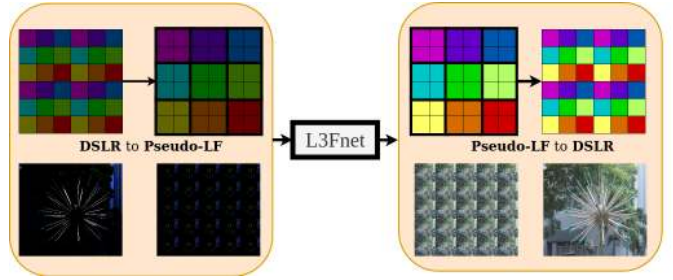


Fig. 5. Pseudo-LF: Extending the proposed L3Fnet to single-frame DSLR images. The proposed Pseudo-LF transformation covers a 2D DSLR image into a 4D pseudo-LF. Pseudo-LF transformation allows a CNN designed for LF to also process single-frame DSLR images.

$(u_k, v_k)^{th}$  view of the LF having  $K$  views. The first component in our loss function performs a dense matching (i.e., pixel-to-pixel) between the restored and the ground truth LF to recover the structure and color information. Since recovering the precise color information from a low-light signal is a hard problem, a small shift from actual hue information is expected, no matter how long the network is trained. So, in a bid to prevent our loss function from having large errors due to this mismatch, we chose the sum-of-absolute-deviation ( $L1$  loss) over other alternatives such as  $L2$  loss to perform the dense pixel matching. But as our dataset has small misalignment, directly using the  $L1$  loss would lead to blurred outputs. To prevent this we additionally used a low weighted contextual loss to handle this problem. Our ablation studies nicely demonstrate the importance of both components in our loss function.

## V. PSEUDO-LF: EXTENDING L3FNET TO SINGLE-FRAME DSLR IMAGES

The proposed L3Fnet has been specifically engineered for low-light Light Fields and thus cannot enhance low-light single-frame DSLR images. On the contrary, we will show that (see Fig. 6) methods designed for single-frame DSLR images are also inappropriate for low-light LF. Thus, we introduce a very simple and light-weight transformation called *pseudo-LF transform*, which allows L3Fnet to enhance even DSLR images. This is illustrated in Fig. 5.

The first step is to convert the the DSLR image into *pseudo-LF*. To do this, the high-resolution DSLR image is divided into blocks of  $B \times B$  pixels. The  $i^{th}$  pseudo SAI, where  $i \in [1, \dots, B^2]$ , is then obtained by collecting together the  $i^{th}$  pixel from each  $B \times B$  block. We use the term *pseudo* because the resulting pseudo-LF SAIs have no real disparity and only happen to be shifted subsampled versions of the input high-resolution DSLR image. The converted pseudo-LF, and not the original DSLR image, is then processed by the L3Fnet to obtain well-lit pseudo-LF. The resulting well-lit LF is finally converted back to well-lit DSLR image by reversing the sampling process described just now. Recently Gu *et al.* [39] and Shi *et al.* [53] have also used a similar pixel shuffling technique for better performance on DSLR images but do not allow CNNs designed for LF to process DSLR images.

The proposed pseudo-L3Fnet pipeline may appear a workaround to fit L3Fnet in the DSLR image framework, but

TABLE III

PSNR(dB)/SSIM COMPARISON OF THE PROPOSED L3FNET WITH RECENT WORKS ON THE L3F-20, L3F-50 AND L3F-100 DATASETS. THE EFFECTIVENESS OF THE PROPOSED L3FNET IS MORE EVIDENT FOR THE EXTREMELY LOW LIGHT IMAGES IN THE L3F-100 DATASET. **BOLD** INDICATES BEST VALUE AND UNDERLINE INDICATES SECOND-BEST.

	L3F-20	L3F-50	L3F-100
<b>LFBM5D</b> [32]	24.48/ <u>0.79</u>	20.94/0.64	18.14/0.46
<b>PBS</b> [42]	20.80/0.68	16.48/0.53	13.94/0.38
<b>RetinexNet</b> [41]	21.82/0.72	18.98/0.59	17.8/0.41
<b>DID</b> [40]	24.09/0.78	<u>22.63/0.68</u>	<u>20.68/0.61</u>
<b>SGN</b> [39]	24.10/0.76	22.18/0.67	20.70/0.59
<b>SID</b> [19]	<u>24.53/0.76</u>	<u>22.87/0.66</u>	<u>20.75/0.58</u>
<b>Our L3Fnet</b>	<b>25.25/0.82</b>	<b>23.67/0.74</b>	<b>22.61/0.70</b>

it has some inherent advantages. It allows a large receptive field which helps the restoration by gathering more contextual information. For example, modern CNN architectures mostly use a  $3 \times 3$  convolution kernel that operates directly over a DSLR image and hence it has a receptive field of just  $3 \times 3$  for one convolution layer. However, we first convert the DSLR image into  $B^2$  pseudo-SAIs. Stage-I of L3Fnet then collects all the  $B^2$  pseudo-SAIs and subsequently performs  $3 \times 3$  convolution with a stride of 1. This is equivalent to a receptive field of  $3B \times 3B$  and with a stride of  $B$  in the DSLR image. In our experiments, we use  $B = 10$  and so the effective receptive field is  $30 \times 30$ . To have the same effect by convolving directly on the DSLR image, requires a  $30 \times 30$  convolution kernel with a stride of 10. However, such large kernel sizes and strides have become obsolete in modern CNN architectures and deep learning libraries are mostly optimized for  $3 \times 3$  kernel size [54]. Thus, even though L3Fnet has been designed for LF, by using the pseudo-LF transformation it can be effectively used for single image enhancement with the advantage of having a large receptive field.

## VI. EXPERIMENTAL RESULTS

### A. Implementation details

**Data Pre-processing:** Recent works on single-frame low-light image restoration [19], [55] chose to work with raw format. This is because the raw data directly stores the signal measured by the sensor and so is immune to any loss of information due to the Image Signal Processing pipeline of the camera. However, there are few difficulties with directly working on raw LF obtained from Lytro Illum. The raw LF requires a significant amount of pre-processing [35] which involves tasks such as alignment rectification, hexagonal to orthogonal lattice conversion, de-vignetting, and demosaicing [35]. We wanted L3Fnet to focus on low-light restoration rather than learning these difficult geometric transformations for which already efficient techniques exist. So we chose to work with decoded JPEG images with the highest quality factor. From the training time and resource utilization perspective also this is beneficial because the JPEG compression reduced the decoded image size from massive  $400 - 500MB$  to  $40 - 50MB$ , which is still much larger than raw single-frame DSLR data.

Lytro Illum captures 225 views of a scene arranged in a  $15 \times 15$  grid. As the peripheral views are affected by ghosting and vignetting effect [25], [26], we restore only the central  $8 \times 8$  views. For the SAIs lying at the boundary of this  $8 \times 8$  grid, the adjoining SAIs are taken from the 9<sup>th</sup> row and column.

**Reducing Memory Requirement:** Stage-I computes the global representation by looking at all the 64 SAIs. This representation is then used by Stage-II to restore each SAI. But to speed-up the training and reduce the model size on GPU, we pass only 12 randomly chosen SAIs to Stage-II and the loss is computed only over these 12 SAIs. Note that during inference, however, we can together restore all the SAIs because during inference, the chain rule mechanism of PyTorch is disabled, which is responsible for most of the memory and computation.

**Loss Function:**  $L1$  loss turns out to be more crucial than the CX loss [52] in our ablation studies (see Table V). We thus set  $\alpha_1 = 5$  and  $\alpha_2 = 0.1$  for first  $20k$  iterations. After this  $\alpha_1$  is reduced to 1. For CX loss, we used the feature maps at the 9, 13 and 18<sup>th</sup> layer of VGG19 [56].  $\lambda$  was fixed to  $10^{-6}$  for all iterations. L3Fnet was trained for  $100k$  iterations.

**Data augmentation:** To augment the data in the training phase, we use horizontal flipping, vertical flipping, and color augmentation. Color augmentation is achieved by swapping the color channels in random order. The training is done on patches of size  $180 \times 180$  and full spatial resolution is used at the time of testing. In each iteration, the patch location is chosen randomly within each sub-aperture view. Since each view has a resolution of  $625 \times 433$  and we use randomly selected patches of  $180 \times 180$ , along with data augmentation techniques we had sufficient training data.

**Baseline:** We compare our L3Fnet with a LF denoising technique LFBM5D [32] and single-frame low-light enhancement methods SID [19], SGN [39], DID [40], RetinexNet [41] and PBS [42]. As LFBM5D is a denoising technique, we had to suitably scale it for better color restoration before proceeding for denoising. LFBM5D additionally requires an estimate of the noise variance. This was obtained from small texture-less patches of the low-light LF.

SID for single-frame images chose to work with raw format and performed an end to end training. It is however not suited for raw Lytro LF images. For reconstruction using SID, it has to be independently operated over each SAI which is like a single-frame image. The SAIs are however not readily available in raw LFR format and needs to be decoded using the Light Field Matlab Toolbox [35].

The existing methods such as SID were trained and evaluated using their own loss function comprising of just the  $L1$  loss. We, however, also tried our loss function consisting of  $L1 +$  Contextual loss to train them. We found that our loss function gave slightly better results and so we use these results for comparison.

### B. Visual and Geometric Reconstruction Comparisons

The proposed L3F-20, L3F-50 and L3F-100 datasets have very different light illumination levels, see Sec. III. In our first experiment, we train and test our method and the existing methods for each of the three datasets independently. That

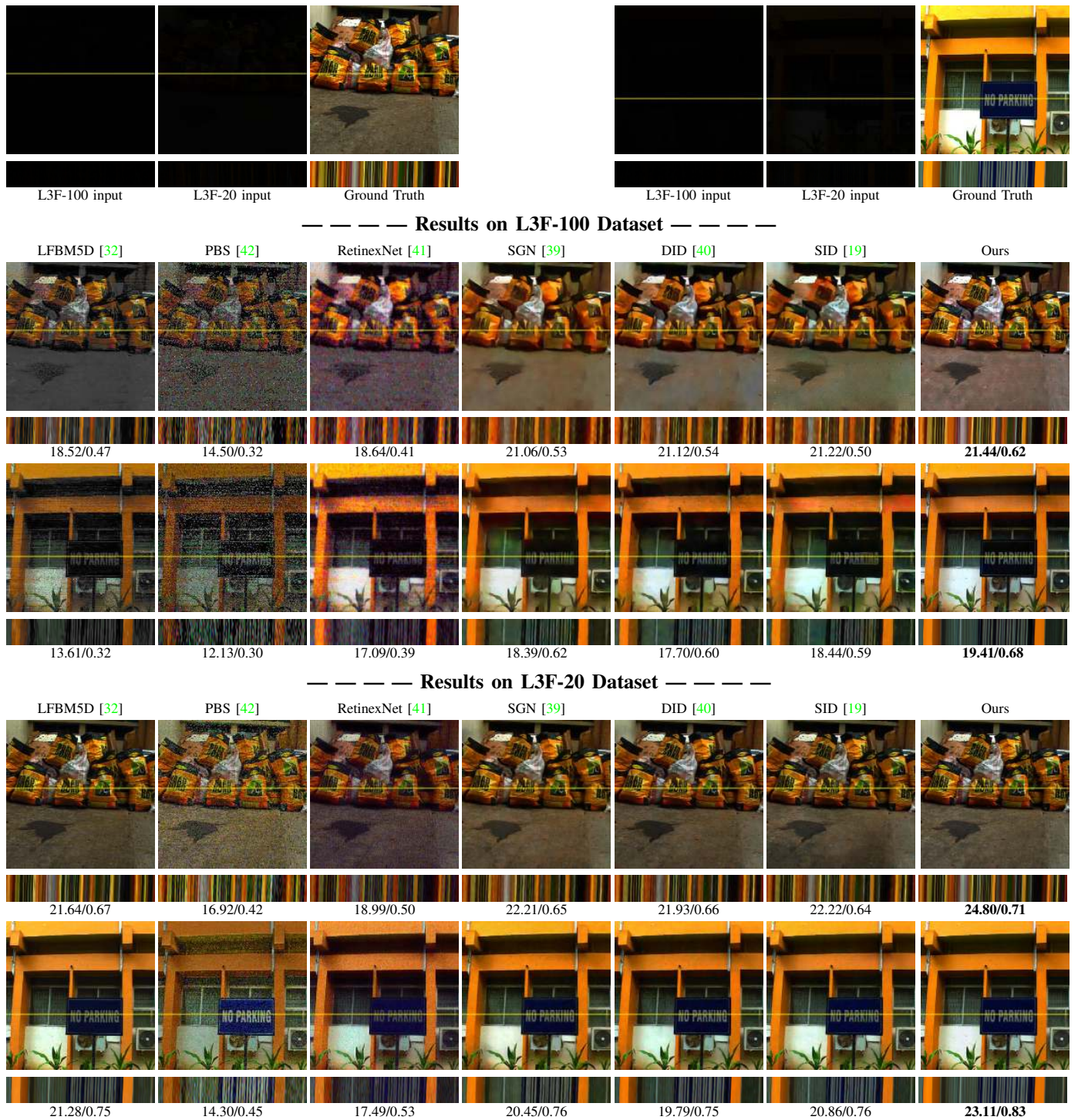


Fig. 6. (Best viewed when zoomed in) Visual and epipolar comparisons of the restored Light Fields. For the very low light case of the L3F-100 dataset, all the existing methods obtain very blurry restorations with much loss in details. Additionally, they sometimes have a large amount of noise. The proposed L3Fnet, however, overcomes these limitations to a good extent.

is, we train all the methods on the training-set of L3F-20 dataset and then test it on the testing-set of L3F-20. The same is repeated for L3F-50 and L3F-100 datasets. Our proposed L3Fnet is equipped with an amplification module (which we call as the Histogram module) which can handle varying illumination levels. However, existing methods lack this feature. Thus, for a fair comparison with the existing methods, we switched OFF the Histogram Module of L3Fnet

for the above experiments. Anyway, since we are training the methods for a particular light level (for example L3F-20) and testing it for the same light level (L3F-20), there is no need for estimating the amplification factor.

The results are shown in Fig. 6 and Table. III. The proposed L3Fnet does better restoration for all three datasets. However, the difference between L3Fnet and the other methods is most evident for the extreme low-light case of L3F-100 dataset. For



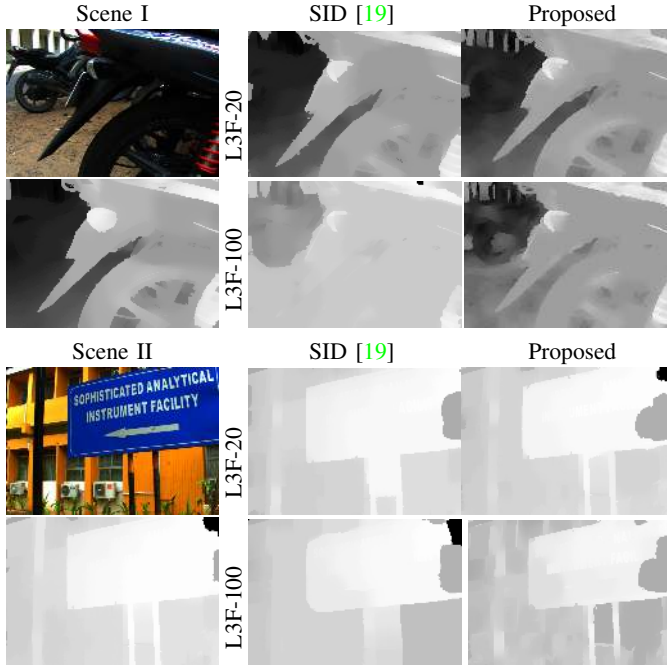


Fig. 7. Depth estimation from reconstructed low-light LFs for low-light (L3F-20) and extreme low light (L3F-100) cases: Depth estimates from LFs restored using SID are good for L3F-20 but very poor for the case of L3F-100 dataset because it is hardly able to preserve the epipolar geometry. On the other hand, our method is able to preserve the epipolar geometry even in the extreme low-light case.

this dataset, all the existing methods struggle to restore the finer details and the results are very blurry. L3Fnet harnesses information from all the views and hence is able to better restore each of the views. The single-frame methods such as SID [19], SGN [39] and DID [40], however, restore each SAI independent of other SAIs which causes performance degradation, including detail loss and blurry results. We further observe that for PBS [42] and RetinexNet [41], which are also single-frame methods, the restoration additionally has a lot of noise. This is because these methods have been designed for enhancing under-exposed images, which have sufficient color information with moderate level of noise. However, for the challenging case of extreme low light imaging, color information is almost lost and noise is very high.

The restored image may look aesthetically pleasing but the LF geometry might be destroyed. We thus additionally show the depth estimates by L3Fnet and SID for L3F-20 and L3F-100 dataset in Fig. 7. The method proposed by Jeon *et al.* [7] is used for depth estimation. Similar to visual restorations, depth estimation becomes more and more challenging for lower light levels and the depth estimates of L3Fnet are closer to the ground truth.

For the L3F-20 dataset, depth estimates from LFs restored using SID are good but lack the fine details. For example the spokes of the bike’s tyre in scene I of Fig. 7 are clearly demarcated in L3Fnet’s depth estimates but not for the case of SID. Likewise, for scene II the sharpness of the board corners is much better preserved in our results. Similar observations hold true for the L3F-100 dataset.

TABLE IV  
VARIATION CAUSED IN  $\gamma$  VALUES BY THE HISTOGRAM BLOCK ON THE TEST DATASET. THE PREDICTED AMPLIFICATION FACTOR INCREASES MONOTONICALLY WITH DECREASING LIGHT LEVEL.

Dataset	$\gamma$ range
L3F-20	0.3 – 0.4
L3F-50	0.8 – 0.9
L3F-100	1.4 – 1.7

TABLE V  
ABLATION STUDY TO EXAMINE THE CONTRIBUTION OF STAGE-I IN L3FNET MODEL AND THE TWO LOSS FUNCTIONS: L1 AND CX.

	Stage-I	Stage-II	L1 loss	CX loss	PSNR/SSIM
Net-I	✓	✓	✓	✗	21.71/0.63
Net-II	✓	✓	✗	✓	13.68/0.18
Net-III	✗	✓	✓	✓	22.21/0.65
Proposed	✓	✓	✓	✓	<b>22.61/0.70</b>

### C. L3Fnet for LF captured in Wild

In this section, we evaluate the performance of L3Fnet on the L3F-wild dataset. This dataset was collected under near 0 lux conditions and hence neither the ground-truth is available nor the optimal exposure is known. Therefore, it is crucial to estimate an appropriate amplification factor to restore these LF images. To facilitate this, the Histogram Module of L3Fnet was switched ON.

Since ground truth is not available for these images it cannot be used for training L3Fnet. We instead re-train L3Fnet with L3F – 20, 50, and 100 datasets merged together but pre-processed with the Histogram Module to estimate the desired amplification factor  $\gamma$ . The weights of the Histogram Module and L3Fnet were learnt together in a end-to-end fashion. We call this trained network as L3Fnet- $\gamma$ . In Table X we report the range of  $\gamma$  values predicted by L3Fnet- $\gamma$  for different light levels. For ease in notation and brevity, we refer L3Fnet trained on L3F – 20, 50, and 100 datasets as L3Fnet-20, L3Fnet-50, and L3Fnet-100, respectively.

In Fig. 15, we show a night time scene captured using different ISO and exposure settings and try to restore it using L3Fnet-20, L3Fnet-50, L3Fnet-100 and L3Fnet- $\gamma$ . We can easily notice the over/under saturation artifacts in LFs restored by L3Fnet-20, L3Fnet-50 and L3Fnet-100. The reason is that these networks are agnostic to image statistics and hence can not adapt to different illumination condition. But L3Fnet- $\gamma$  avoids this problem to a large extent by estimating an appropriate amplification for input LF. We also tried using Gamma correction instead of linear amplification, but the restoration exhibited a lot of artifacts. More results and discussion can be found in the supplementary.

### D. Ablation Study

**Effect of different loss functions:** We trained 2 variants of the proposed L3Fnet on the L3F-100 dataset called Net-I and Net-II as shown in Table V. Net-I, which was trained only with L1 loss, gave slightly blurry results. In contrast, Net-II was trained only with the Contextual Loss. While Net-I showed a small dip in performance due to blurriness, Net-II performed very poorly. The reason is without the L1 loss, the network could not learn the correct colors. We therefore have

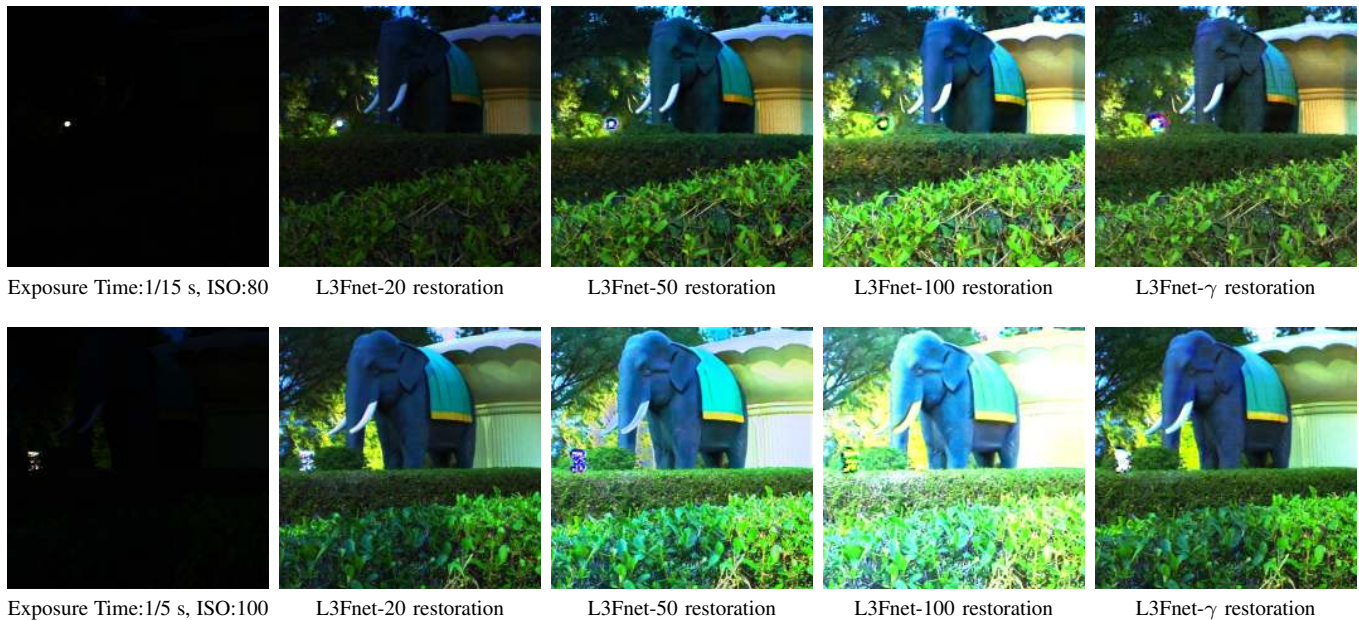


Fig. 8. Results on the L3F-wild dataset. The dark LF images have been pre-processed with the Histogram Module allowing L3Fnet to adapt to different levels of low-light. For example, in the top row, L3Fnet trained on L3F-20 dataset (denoted as L3Fnet-20) fails to achieve ambient brightness. Likewise, when the light levels are increased for test image in the second row, L3Fnet-100 oversaturates. L3Fnet- $\gamma$ , however, adjusts to both lighting conditions using the Histogram Module. Check supplementary for more visual results.

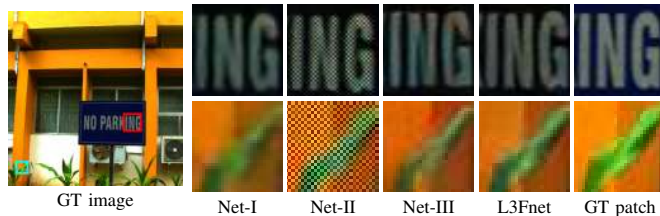


Fig. 9. Visual reconstruction results for ablation study on L3Fnet. Net-I produces blurry results since it uses only  $L1$  loss. Net-II restores the sharpness by instead using the contextual loss but introduces a large amount of color artifacts. Net-III solves both problems by incorporating both loss functions, but because of the absence of Stage-I it could not capture the LF geometry leading to inferior results than those obtained by the proposed L3Fnet. The difference is better corroborated by comparing depth estimations results in Fig. 10. Refer Table V for PSNR/SSIM values.

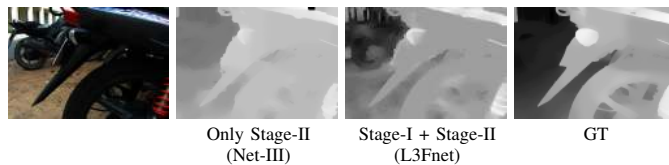


Fig. 10. Depth estimates from reconstructed LF with and without Stage-I of the L3Fnet model. Stage-I captures the LF geometry and hence it helps in producing better depth maps.

both  $L1$  and Contextual loss for the proposed L3Fnet. Some visual results are also given in Fig. 9. Since the exclusion of  $L1$  loss penalises the restoration far more than the Contextual loss, we first try to restore the color and basic geometry by giving higher importance to  $L1$  loss in the first  $20k$  iterations and subsequently reduce it to let contextual loss finetune the result. Besides the  $L1$  and Contextual loss we also tried to train with the SSIM loss along with  $L1 +$  Contextual losses. Using the SSIM loss, increases the SSIM value of the restored LF marginally, but the decrease in PSNR value is slightly more, see supplementary.

### Importance of Global Representation Block (Stage-I):

We trained a third variant of the proposed L3Fnet on the L3F-100 dataset by removing the Stage-I and instead added the 4 residual blocks of Stage-I to Stage-II so that the model parameters remains the same. Without Stage-I, L3Fnet was not able to preserve the epipolar constraints of LF. Fig. 10 shows that the depth map obtained from the LF restored by Net-III is inferior to that obtained from the LF restored when both stages are included in L3Fnet. Some more qualitative and quantitative results pertaining to Net-III can be found in Fig. 9 and Table V respectively.

### Effect of number of residual blocks and model parameters:

We have fixed the total number of residual blocks in L3Fnet to 10 for computational reasons. Out of the total 10 residual blocks, we allocate 4 residual blocks to Stage-I and 6 to Stage-II. We chose more residual blocks for Stage-II because our main goal is LF enhancement, which is being explicitly performed by Stage-II. Moreover, we want L3Fnet to give more weightage to the immediate neighbors of the SAI being restored and this is only possible in Stage-II. In the supplementary, we report several experiments that validate our intuition. In these experiments, we tried allocating a different number of blocks to Stage I and II and found that the proposed allocation is better. In the supplementary, we have also analyzed the performance of L3FNet by varying the model parameter count and found the performance to be quite robust to change in model parameters.

### E. Pseudo-LF for single-frame image reconstruction

In this section, we evaluate the proposed pseudo-L3F transformation for processing single-frame low-light DSLR images. The purpose here is not to surpass low-light DSLR

Ground Truth	Baseline	SID [19]	Pseudo-L3Fnet
BRISQUE, NIQE	47.68, 4.97	44.71, 5.10	<b>38.35, 4.84</b>
PSNR / SSIM	17.75/ <b>0.92</b>	20.48/0.85	<b>20.62/0.83</b>
	45.59, 5.07	48.84, 5.11	<b>41.24, 3.92</b>
	17.92/0.42	19.03/0.47	18.03/0.60
	45.65, 4.99	47.85, 5.26	<b>36.47, 4.36</b>
	<b>12.76/0.56</b>	12.16/0.58	12.45/0.64
	46.32, 5.22	46.40, 5.26	<b>40.03, 3.53</b>
	19.68/0.36	19.63/0.43	<b>20.77/0.57</b>

Fig. 11. Single-frame low-light DSLR restoration using the proposed pseudo-L3Fnet pipeline. BRISQUE and NIQE are recent perceptual metrics for single-frame DSLR images. Lower are these metrics, better is the perception.

reconstruction methods but to highlight the universality of L3Fnet to enhance both LF and single-frame images.

Similar to the L3F-100 dataset, we capture 50 extreme low-light DSLR images of which 14 were reserved for testing. We process JPEG images only as the raw format did not give much improvement but instead had much slower convergence because of additionally learning the Bayer demosaicing. We used all the data augmentation techniques mentioned for L3Fnet with patch-wise training.

We compare our pseudo-L3Fnet pipeline with SID [19]. We also create a new baseline with the same number of residual blocks present in pseudo-L3Fnet. Pseudo-L3Fnet has 10 residual blocks (4 in Stage-I and 6 in Stage-II), and so the new baseline has 10 residual blocks stacked end-to-end with a long skip connection from input to output. The new baseline has approximately the same capacity and number of layers present in pseudo-L3Fnet. The difference is that input to the new baseline is the actual DSLR image, while the input to pseudo-L3Fnet is DSLR image converted to pseudo-LF.

Some of the restoration results are shown in Fig. 11. The baseline and SID do good at denoising but very poorly on preserving the spatial smoothness, which is visible by the presence of color blobs spread all over the image. On the contrary, pseudo-L3Fnet obtains much gradual transitions mitigating the color blobs problem. This is because, as described in Sec. V, the pseudo-LF transformation awards L3Fnet with a large receptive field. To verify this, we experimentally computed the receptive field and found that L3Fnet has a large receptive field of  $830 \times 830$ , while for SID and Baseline it is just  $252 \times 252$  and  $83 \times 83$ , respectively. The large receptive field of L3Fnet helps gather more contextual information for better restoration. The average PSNR(dB)/SSIM values for SID is 17.21/0.61, for baseline is 17.08/0.56 and for the proposed Pseudo-LF is 19.04/0.62. Further, the average NIQE [57] values for SID is

5.24, for baseline is 5.01 and 4.29 for pseudo-LF. The average BRISQUE [58] values for SID is 48.30, for baseline is 46.48 and 41.45 for the proposed pseudo-LF. A lower value for these metrics is considered better.

## VII. CONCLUSION AND DISCUSSION

The primary objective of this work was to enhance LF captured in low-light conditions. We showed that the existing single-frame low-light enhancements methods find it hard to preserve the LF geometry because they reconstruct each LF view independently. To this end, we proposed a low-light L3F dataset and a two-stage L3Fnet architecture. The effectiveness of L3Fnet was shown on LFs for varying levels of low-light by conducting experiments on L3F-20, L3F-50 and L3F-100 datasets. Additionally, we used the Histogram Module to automatically tune the amplification factor  $\gamma$ . With this pre-processing module, L3Fnet could now automatically adapt to different light levels, which was substantiated by showing results on the L3F-wild dataset.

We additionally showed that while single-frame methods are not conceptually suited for LF related tasks, our L3Fnet can be used for decent enhancement of single-frame low-light images also. This was achieved by converting the DSLR images into pseudo-LF and vice-versa. Of course, L3Fnet is better optimized for LF and may be modified in the future to equally suit both LF and DSLR images simultaneously.

As a future work, another interesting direction would be to explore which camera is more suited for low-light reconstruction: a single-frame DSLR camera or a Light Field camera? While DSLR cameras have high resolution, LF cameras may be helpful because of complementary information present in various LF views. Besides, depth estimation is a clear advantage for LF cameras.

## VIII. SUPPLEMENTARY

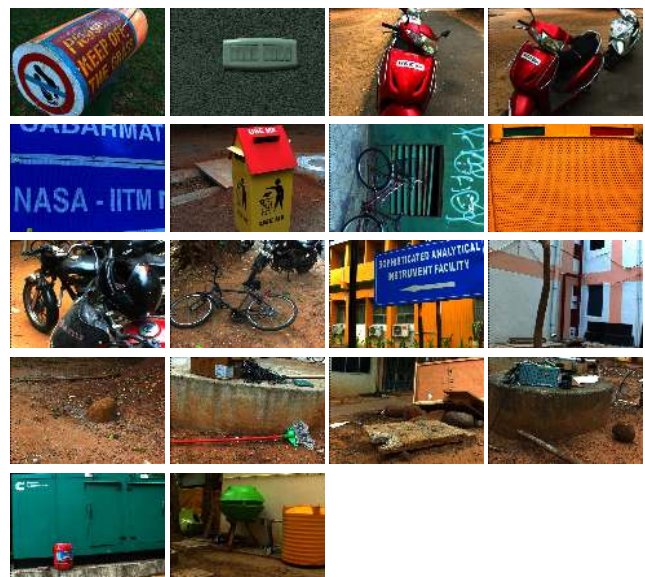


Fig. 12. Central SAI of the ground truth LFs present in the training set of the proposed L3F dataset. This does not include the L3F-Wild dataset.



Fig. 13. Central SAI of the ground truth LFs present in the test set of the proposed L3F dataset. This does not include the L3F-Wild dataset.

TABLE VI  
PSNR (DB)/ SSIM COMPARISON FOR THE L3F TEST IMAGES SHOWN IN FIG. 13

Image	L3F-100 dataset	
	SID [19]	Our L3Fnet
Forest	22.03/0.38	<b>24.79/0.67</b>
Generator	28.01/0.80	<b>31.30/0.85</b>
Cement	21.22/0.50	<b>21.44/0.62</b>
SignBoard	18.44/0.59	<b>19.41/0.68</b>
Ramanujam	19.89/0.72	<b>22.86/0.79</b>
Bikes	22.63/0.68	<b>23.16/0.80</b>
PaintBox	11.84/0.42	<b>12.79/0.53</b>
Crane	20.40/0.42	<b>23.47/0.61</b>
Grass	22.27/0.65	<b>24.22/0.70</b>

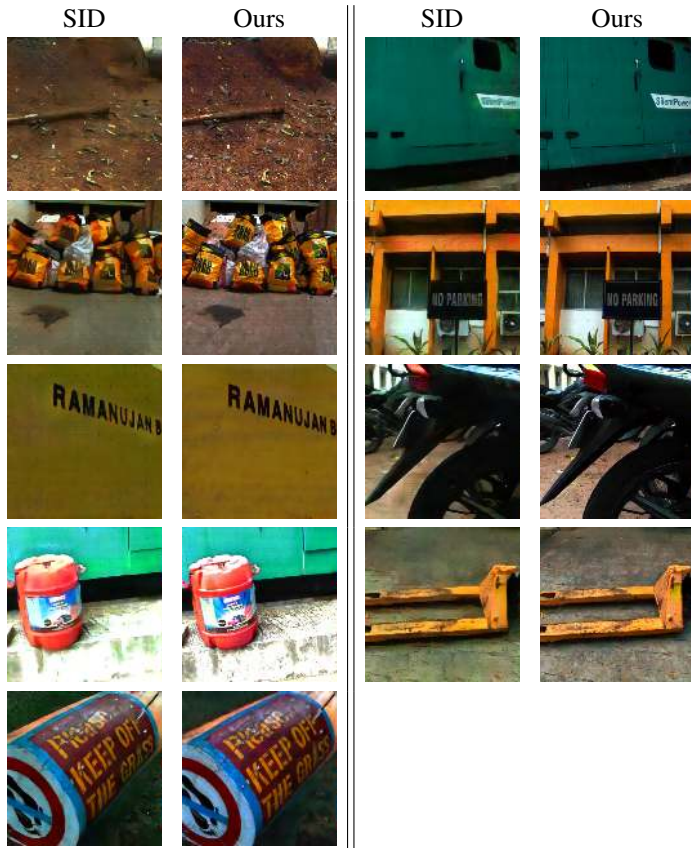


Fig. 14. L3F-100 Restoration results for the test data shown in Fig. 13.

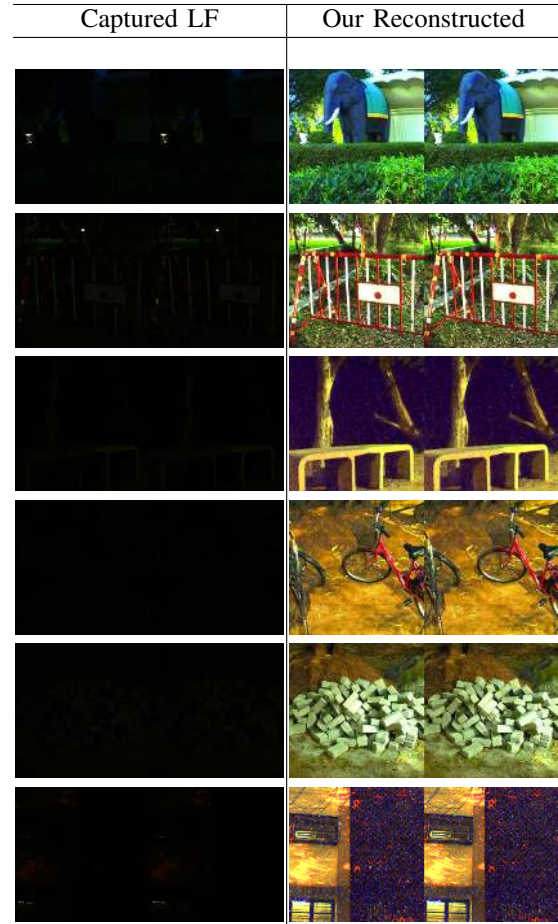


Fig. 15. Some L3F-Wild Reconstruction Results [Trained on L3F-20+L3F-50+L3F-100, see Fig. 12] with Histogram Module Pre-processing. The figure shows two consecutive SAIs.

TABLE VII

VARIATION IN PERFORMANCE BY CHANGING THE PARAMETER COUNT IN THE PROPOSED L3FNET. THESE NETWORKS HAVE BEEN RE-TRAINING ON THE L3F-100 DATASET. EVEN WITH MUCH LOWER PARAMETER COUNT, WE OUTPERFORM SID [19] WHICH HAS MORE THAN 7 MILLION PARAMETERS. NOTATION:  $S1$  = NUMBER OF RESIDUAL BLOCKS IN STAGE-I,  $S2$  = NUMBER OF RESIDUAL BLOCKS IN STAGE-II,  $C$  = NUMBER OF CHANNELS IN EACH RESIDUAL BLOCK,  $CT$  = NUMBER OF CHANNELS IN THE FINAL TRANSPOSED CONVOLUTION LAYER.

Model description	Parameter count	PSNR(dB) / SSIM
Proposed. $S1=4$ , $S2=6$ , $C=128$ , $CT=128$	3.7 Million	22.61/0.70
Reduce number of Residual Blocks. $S1=2$ , $S2=2$ , $C=128$ , $CT=128$	2.2 Million	22.60/0.69
Increase the number of Residual Blocks $S1=6$ , $S2=9$ , $C=128$ , $CT=128$	5.2 Million	22.57/0.69
Reduce the channels in Residual Blocks. $S1=4$ , $S2=6$ , $C=64$ , $CT=128$	1.4 Million	21.87/0.68
Increase the channels in Residual Blocks. $S1=4$ , $S2=4$ , $C=256$ , $CT=128$	13.4 Million	22.50/0.69
Increase the richness of final activation layer. $S1=4$ , $S2=6$ , $C=128$ , $CT=1024$	4.2 Million	22.60/0.70

#### A. Effect of parameter count on performance

The proposed L3Fnet with 3.7 Million parameters, outperforms the best performing baseline in our experiments SID [19] which has more than 7 Million parameters. To further understand the effect of parameter count on model performance, we conducted additional experiments. In these experiments, the parameter count in L3Fnet were varied and then re-trained on the L3F-100 dataset (see Table VII). The details are given below:

- **Reduce the number of residual blocks:** Currently, L3Fnet has 4 residual blocks for Stage-I and 6 for Stage-II. We reduce this to half and thus Stage-I has 2 residual blocks while Stage-II has 3. This reduces the parameter count from 3.7 Million to 2.2 Million. The performance is almost the same as the original network.
- **Increase the number of residual blocks:** Currently, L3Fnet has 4 residual blocks for Stage-I and 6 for Stage-II. We increase this to 6 and 9 respectively. This increases the parameter count from 3.7 Million to 5.2 Million. The performance is almost the same as the original network.
- **Reduce the channels in Residual Block:** Currently L3Fnet has 10 Residual Blocks each using 128 channels for each convolution. We reduced the number of channels to 64, which in turn reduced the parameter count from 3.7 Million to 1.4 Million. We observe a drop in performance.
- **Increase the channels in Residual Block:** We increased the number of channels in the Residual Blocks from 128 to 256. This increased the parameter count from 3.7 Million to 13.4 Million. With this huge increase in the

TABLE VIII

PSNR (dB) / SSIM METRICS BY SWAPPING THE NUMBER OF RESIDUAL BLOCKS IN STAGE-I AND STAGE-II. IN BOTH CASES L3FNET WAS RE-TRAINED AND TESTED ON THE L3F-100 DATASET.

Number of Residual Blocks in Stage-I	Number of Residual Blocks in Stage-II	PSNR(dB)/SSIM
4	6	22.61/0.70 (Proposed)
6	4	22.30/0.69
3	7	22.28/0.69
7	3	22.27/0.68

number of model parameters the performance decreases a bit possibly due to over-fitting.

- **Increase the richness of final Transposed Convolution:** To obtain the final 3 channel RGB image it is always good to average out a large feature map with richer activations. We thus increase the number of output channels in the Transpose-2D convolution from 128 to 1024. This increased the parameter count from 3.7 Million to 4.2 Million but with no change in performance.

The results are summarized in Table VII. Overall the performance of the proposed L3Fnet remains stable with the change in parameter count. Save and except for the case of 1.4 Million parameters, change in parameter count causes a variation of maximum 0.1 dB in the PSNR and 0.01 in the SSIM metric. A closer inspection of Table VII reveals that our results corroborate the ‘U-shaped’ curve of the famous bias-variance trade-off. Initially when model parameters are less (close to 1 Million) the performance increases with parameter count. However, after some *sweet spot* it starts to decline marginally with increase in model parameters.

#### B. Given a fixed number of Residual Blocks, how to distribute them among Stage-I & Stage-II ?

For the proposed L3Fnet we have used 4 residual blocks for Stage-I and 6 for Stage-II. We chose more residual blocks for Stage-II because our main goal is LF enhancement, which is being explicitly performed by Stage-II. Stage-I looks at all the 64 SAIs and computes a Global Representation. In the restoration process however, it should not happen that to restore a pixel from a particular SAI  $\phi$ , the network is looking at all the SAIs without any discrimination. *Rather to restore a pixel from SAI  $\phi$  the network should gather more contextual information from  $\phi$  and its immediate neighbors.* To allow this, we designed Stage-II which looks at the Global Representation of Stage-I but at the same time gives exclusive attention to the particular SAI (and its 4-nearest neighbors) which is being restored. Given that we want to give more weightage to the immediate neighbors, we allocated more residual blocks to Stage-II.

We conducted more experiments by varying the number of residual blocks in Stage I and II, see Table VIII. For all these experiments, training and testing was carried out on L3F-100 dataset. We kept the total number of residual blocks to 10 so that the parameter count of L3Fnet is low (it is less than half of

TABLE IX  
PSNR/SSIM METRICS OF THE PROPOSED L3FNET ON THE L3F-100 DATASET FOR VARIOUS LOSS FUNCTIONS.

L1 loss	Contextual loss	SSIM loss	PSNR(dB) / SSIM
✓	✗	✗	21.71/0.63
✓	✓	✗	22.61/0.701
✓	✗	✓	22.40/0.711
✓	✓	✓	22.45/0.710

what is found in SID [19] and yet outperform it). We observe that flipping the number of residual blocks between Stage I and II, results in a small drop in performance. We also tried 7 residual blocks for stage-I and 3 for stage-II and the other way around, see Table VIII. For both the cases, the results are lower than the proposed L3Fnet.

C. Trying different combinations of L1, Contextual and SSIM loss

The main reason why we chose Contextual Loss was to handle any minor camera shake that would have occurred during the dataset collection. But we also tried using the SSIM loss. We do not find much difference in performance in either using SSIM or Contextual Loss, see Table IX. Using the SSIM loss, SSIM increases marginally, but the decrease in PSNR is slightly more. And since recent methods such as Gu et al. [39] and recently popular NTIRE challenges (<https://data.vision.ee.ethz.ch/cvl/ntire20/>, <https://data.vision.ee.ethz.ch/cvl/ntire19/>, <https://data.vision.ee.ethz.ch/cvl/ntire18/>) prefer PSNR over SSIM for deciding the best performance, we do not include the SSIM loss.

D. On the use of Linear amplification vs Gamma Correction

**Range of  $\gamma$ :** Table X shows the range of  $\gamma$ . The learned  $\gamma$  values follow our intuition that it should be higher for darker images.

**Can  $\gamma$  cause over-saturation?** : No, it is very unlikely that the Histogram Module can cause oversaturation. The Histogram Module is applied at the beginning of L3Fnet and not on the final restored image. As everything is learned end-to-end, the supervision provided by the GT images would modify the model weights such that the restored image is within the desired range. Besides this, the pixel intensities of the captured low-light LF are very small and the learned  $\gamma$  values are less than 2 (see Table X). Thus multiplying the input low-light LF with  $\gamma$  does not cause oversaturation and the pixel intensities still remain in the range [0,1]<sup>1</sup>. Some restoration results are shown in Fig. 15.

**Ablation study on linear amplification vs. Gamma correction:** In an ablation study we re-trained the Histogram Module with the proposed L3Fnet in an end-to-end fashion with the linear multiplication replaced with gamma correction. Specifically,  $\mathcal{L}^{norm} = \gamma \times \mathcal{L}^{low}$  was changed to  $\mathcal{L}^{norm} =$

<sup>1</sup>In our experiments all images are normalized in the range [0,1]. Thus the maximum pixel intensity in our experiments is 1.

TABLE X  
RANGE OF  $\gamma$  VALUES LEARNED FOR DIFFERENT DATASETS. WE ALSO COMPUTE THE MEAN AND MAXIMUM PIXEL INTENSITY FOR EACH TEST IMAGE PRESENT IN THE DATASET, AND THEIR RANGE IS SHOWN IN THE TABLE. THE LEARNED  $\gamma$  VALUES FOLLOW OUR INTUITION THAT  $\gamma$  SHOULD BE HIGHER FOR DARKER IMAGES.

Dataset	Mean pixel intensity	Max pixel intensity	$\gamma$ range
L3F-20	0.0097 – 0.0257	0.06 – 0.77	0.3 – 0.4
L3F-50	0.0043 – 0.0107	0.04 – 0.41	0.8 – 0.9
L3F-100	0.0026 – 0.0084	0.04 – 0.16	1.4 – 1.7

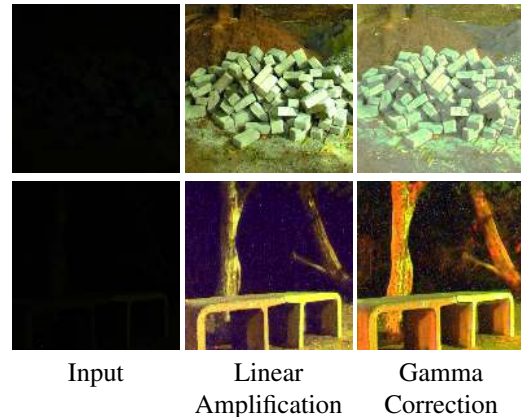


Fig. 16. Restoration results using L3Fnet by first normalizing the input LF using the proposed linear amplification or by using gamma correction. Using gamma correction for normalizing the images, oversaturates the bright pixels in order to boost the dark pixels and exhibits artifacts in the vicinity of bright pixels.

$(\mathcal{L}^{low})^\gamma$ . In Fig. 16 we find that using gamma correction for normalizing the images, oversaturates the bright pixels in order to boost the dark pixels and exhibits some artifacts in the vicinity of bright pixels. A possible reason for this could be the fact that Gamma correction introduces higher order noise terms which will be more difficult to remove by the later CNN stages. We elaborate on this in the next paragraph.

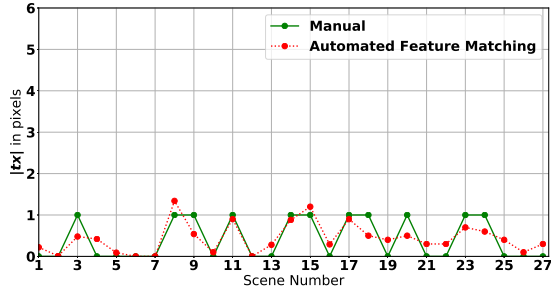
Whenever we capture images in low light, it is affected by a large amount of noise. For simplicity, we can assume  $y = x + n$ , where  $y$  is the captured image in low light,  $x$  is the actual signal we intended to capture and  $n$  is the large amount of noise present due to low-light conditions. Now if we use a linear amplification we obtain,  $\gamma y = \gamma x + \gamma n$ . If however, we use the power law (gamma correction), we obtain,

$$\begin{aligned}
 y^\gamma &= x^\gamma \left(1 + \frac{n}{x}\right)^\gamma \\
 &= x^\gamma + \frac{\gamma n x^{\gamma-1}}{1!} + \frac{(\gamma)(\gamma-1)n^2 x^{\gamma-2}}{2!} \\
 &\quad + \frac{(\gamma)(\gamma-1)(\gamma-2)n^3 x^{\gamma-3}}{3!} + \dots \quad (11)
 \end{aligned}$$

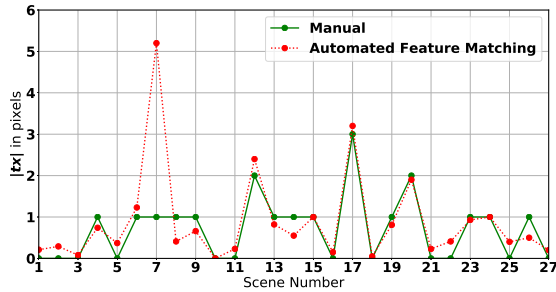
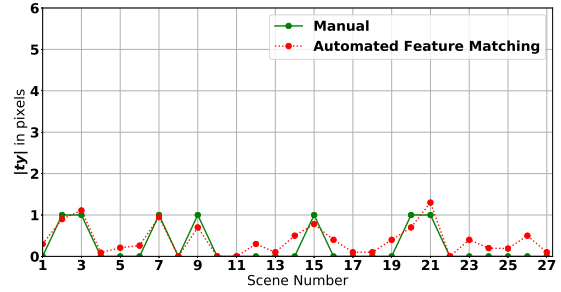
As for low-light  $n$  is very large, the CNN will have a tough time to remove all the higher order terms.

E. Quantifying the misalignment in the proposed Dataset

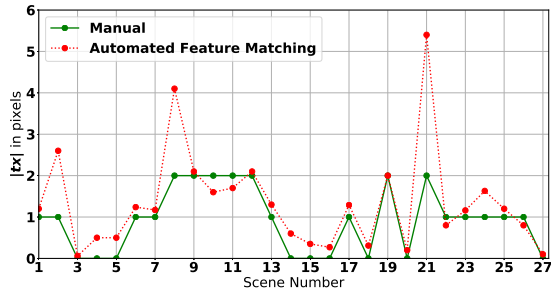
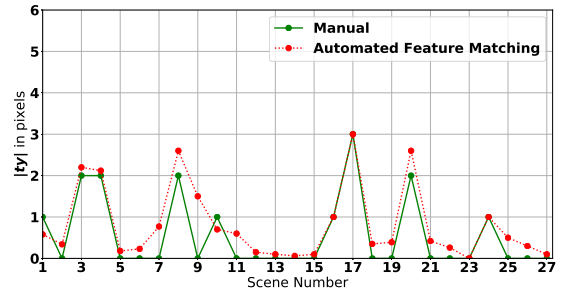
To quantitatively examine the small misalignment in our data, we have adopted two approaches: (a) Use automated



(a) L3F-20 dataset



(b) L3F-50 dataset



(b) L3F-100 dataset

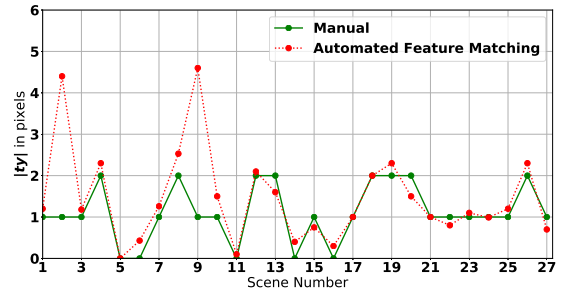


Fig. 17. The figure shows the misalignment in the horizontal (tx) and vertical (ty) direction. Data for rotation is not plotted as both manual and automated feature matching revealed almost zero rotation. The first nine images on the x-axis are the test images and the remaining are the images used for training.

feature correspondences to estimate the transformation required to warp the low-light image onto the GT image. The estimated transformation thereby describes the misalignment quantitatively. (b) Manually inspecting the images to detect and quantify misalignment. Before describing these methods in detail, we summarise the results in Fig. 17. The figure shows the misalignment in the horizontal and vertical directions for all the scenes in our dataset. The average misalignment is  $< 1$  pixel. We now describe the two approaches in detail and then report the results in Sec. VIII-H.

#### F. Using automated feature matching

Employing matching algorithms for extreme low-light images often results in wrong correspondences. This is especially true for the proposed L3F-100 dataset, where we had a tough time finding reliable matched pairs. This is because

the input extreme low-light LF has very poor visibility and SNR, marked by an arbitrarily high amount of noise. A more reliable alternative is to manually inspect the image pairs for misalignment. But as manual inspection for all the image pairs across the entire dataset is virtually impossible, we use a good number of precautionary measures to find a reliable set of feature correspondences using the SIFT descriptor. They are described below:

- The GT image is well-illuminated, and we faced no trouble in finding good descriptors in them. However, we tried several descriptors such as SIFT, SURF, ORB, etc, but they could not detect features in our extreme low-light images. We thus linearly amplified them. The L3F-20 dataset was amplified by  $10\times$ , L3F-50 by  $30\times$  and L3F-100 by  $50\times$ . Although a larger amplification gives better overall brightness level, but also increases the

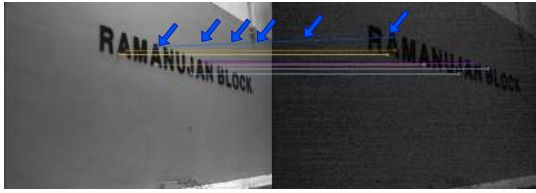


Fig. 18. Here we show the matched feature pairs between a well-lit GT scene (on the left) and extreme low-light scene taken from the L3F-50 dataset. The low-light image was linearly amplified by  $30\times$ . These pairs were obtained after the Cross-Check, Ratio Test and RANSAC, checks were made. But despite all these precautions, we had to also manually check the matched pairs for inconsistencies. The blue arrows point to an inconsistent matched pair that was deleted after manual inspection.

noise — causing inconsistent pairs. We thus tried using the minimum amplification factor required for detecting features in the low-light image. These amplification factors were chosen after empirically experimenting with different amplification factors. The grayscale version of the color images are used for feature matching.

- We then matched the features from the GT image and the low-light image by computing the L1 distance between the feature descriptors. To now remove the inconsistent pairs we employed a technique called ‘*Cross-Check*’. Let  $\mathcal{G} = \{g_1, g_2, \dots, g_n\}$  and  $\mathcal{D} = \{d_1, d_2, \dots, d_m\}$  be the set of detected features in the GT and the input dark images, respectively. Now for each descriptor  $g_i \forall i \in [1, n]$ , we find the descriptor  $d_j \in \mathcal{D}$  such that,

$$\|g_i - d_j\| < \|g_i - d_k\| \forall d_k \in \mathcal{D} \text{ but } d_k \neq d_j. \quad (12)$$

This pair  $(g_i, d_j)$  is called a consistent feature pair, if additionally,

$$\|g_i - d_j\| < \|g_k - d_j\| \forall g_k \in \mathcal{G} \text{ but } g_k \neq g_i, \quad (13)$$

otherwise the pair is called inconsistent and is not used for subsequent calculations.

- To further ensure the matched pair’s consistency, they were subjected to the ‘*Ratio Test*’ mentioned by David Lowe in his seminal paper on SIFT [59]. The threshold ratio was set to 70%.
- After these pre-eliminary checks were made, the RANSAC algorithm was used to remove the outliers with a confidence of 0.99.
- Despite all these precautions, we also randomly checked the matches returned after employing these automatic subroutines and once a while had to manually delete the incorrect correspondences as illustrated in Fig. 18.

Some examples of reliable matched pairs between the low-light and GT SAI are shown in Fig. 19. Here the visualisation is shown for the (7,7) SAIs lying in a  $15 \times 15$  grid. Once we had the matched pairs we estimated the translation and rotation matrix required to warp the low-light SAI onto the GT SAI. Our dataset’s misalignment is not because of casual photography but because of a very small but unavoidable camera shake due to pressing of camera button and movement of the camera’s internal mirrors. Thus it is safe to assume that

the captured dataset is free from scaling, skew and projective distortions. This is also evident from Fig. 20 in which we show the GIF animation of toggling between the GT and low-light SAI. One can hardly see any scaling or skew distortions in this animation. We thus estimate the following transformation,

$$\begin{bmatrix} x_g \\ y_g \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & tx \\ \sin\theta & \cos\theta & ty \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}. \quad (14)$$

Here  $x$  and  $y$  denote the spatial coordinates and the subscripts  $g$  and  $d$  denote GT and dark input image, respectively.  $\theta$  is the rotation angle and  $tx, ty$  are the translation along the horizontal and vertical direction.

### G. Manual Inspection

Automated feature matching is a quick way of estimating the misalignment. Still, occasionally we had to see failure cases because, in our low-light images, the noise significantly dominates the scene visibility. This is especially true for the L3F-100 dataset, where localizing the features is very hard. We thus manually inspected the image pairs to estimate the misalignment. For this, we overlapped the SAIs with 50% transparency and quickly toggled between the GT and dark SAIs to detect the regions showing some displacement. Then with a good amount of zoom-in, we noted the pixels’ spatial coordinates in those regions, which we could uniquely identify in both images. This is a very slow and tedious task but returned a couple of matched pairs with a very high probability of correct correspondence.

### H. Results and conclusion

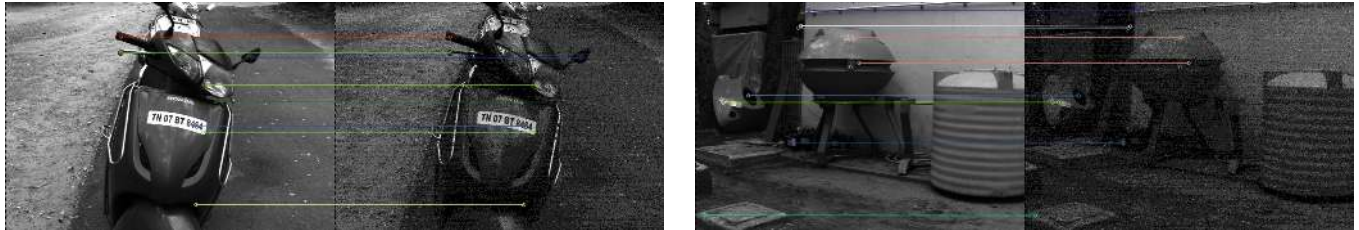
As each SAI present in a LF image captured using Lytro Illum has pixels from all parts of the LF sensor, we focus on analyzing the (7,7) SAIs. The results are shown in Fig. 17. In this figure, we plot  $|tx|$  and  $|ty|$  for the (7,7) SAIs obtained from each scene present in our dataset. The figure shows the estimates obtained using both manual inspection (denoted as ‘*Manual*’ in the legend) and using automated feature matching (denoted as ‘*Automated Feature Matching*’ in the legend). The first nine scenes denoted on the x-axis are the scenes reserved for testing and the remaining scenes were used for training. We do not plot  $\theta$  because the rotation angle was  $< 0.01$  degrees.

We observe that for the L3F-20 dataset the estimation via both methods is in almost perfect harmony. But as the noise increases, especially for the L3F-100 dataset, we sometimes see sporadic peaks for the automated feature matching method. This is because in the L3F-100 dataset the noise dominates the true signal by a large margin. This makes precise localization very difficult, resulting in wrong estimates. Thus, manual inspection of misalignment in such cases is very essential. Please note that, while manually inspecting the L3F-100 dataset, it was quite challenging for us also to do precise localization. At several instants, we found misalignment to be  $< 1$  pixel. But given the huge dominance of noise over the true signal, we decided not to report a conservative estimate and thus, for some scenes upto 2 pixel misalignment is reported. Still, overall both the manual and automated methods estimate

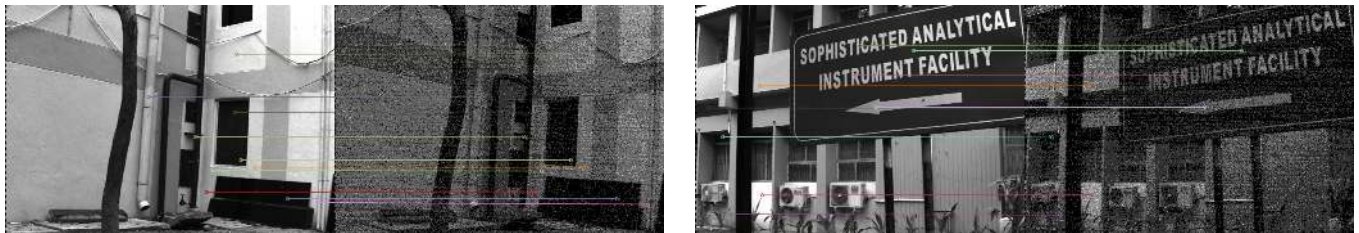




(a) L3F-20 dataset, amplified 10×

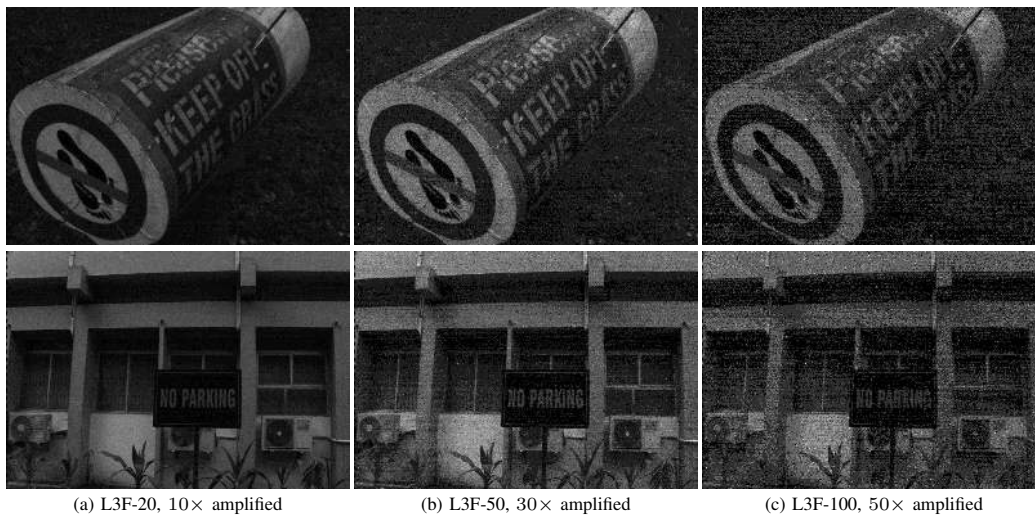


(b) L3F-50 dataset, amplified 30×



(c) L3F-100 dataset, amplified 50×

Fig. 19. Some examples of reliable matched pairs between the low-light and GT SAI. Here the visualisation is shown for the (7,7) SAIs lying in a  $15 \times 15$  grid.



(a) L3F-20, 10× amplified

(b) L3F-50, 30× amplified

(c) L3F-100, 50× amplified

Fig. 20. GIF animation of the toggling operation between the GT and the input extreme low-light SAI. View in Adobe PDF Reader or any other PDF viewer with JavaScript engine enabled.

nearly the same value for most scenes. On an average, the misalignment in both directions is  $< 1$  pixel. We also repeated this experiment for all the SAIs present in the central  $9 \times 9$  grid and the automated feature matching estimated roughly the same misalignment of 1 pixel on an average.

## REFERENCES

- [1] Y. P. Loh and C. S. Chan, “Getting to Know Low-Light Images with the Exclusively Dark Dataset,” *Comput. Vis. and Image Understanding*, vol. 178, pp. 30–42, 2019. 1
- [2] E. H. Adelson and J. Y. A. Wang, “Single Lens Stereo with a Plenoptic Camera,” *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol. 14, no. 2, pp. 99–106, 1992. 1
- [3] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, P. Hanrahan *et al.*,

- “Light Field Photography with a Hand-Held Plenoptic Camera,” *Comput. Science Technical Report CSTR*, vol. 2, no. 11, pp. 1–11, 2005. 1
- [4] M. Levoy and P. Hanrahan, “Light Field Rendering,” in *Conf. Comput. graphics and interactive techniques*. ACM, 1996. 1
- [5] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, “The Lumigraph,” in *Siggraph*, vol. 96, no. 30, 1996, pp. 43–54. 1
- [6] R. Ng, “Fourier slice photography,” in *ACM Trans. Graph.(TOG)*, vol. 24, no. 3, 2005, pp. 735–744. 1
- [7] H.-G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y.-W. Tai, and I. So Kweon, “Accurate depth Map Estimation from a Lenslet Light Field Camera,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015. 1, 2, 9
- [8] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, “Occlusion-Aware Depth Estimation using Light-Field Cameras,” in *IEEE Int. Conf. Comput. Vis.*, 2015. 1, 2
- [9] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, “Depth from combining defocus and correspondence using light-field cameras,” in *IEEE Int. Conf. Comput. Vis.*, 2013. 1, 2
- [10] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, “Learning-Based View Synthesis for Light Field Cameras,” *ACM Trans. Graph. (TOG)*, vol. 35, no. 6, p. 193, 2016. 1
- [11] S. Nousias, M. Lourakis, and C. Bergeles, “Large-Scale, Metric Structure from Motion for Unordered Light Fields,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 1
- [12] C. Jia, F. Shi, Y. Zhao, M. Zhao, Z. Wang, and S. Chen, “Identification of Pedestrians from Confused Planar Objects using Light Field Imaging,” *IEEE Access*, vol. 6, pp. 39 375–39 384, 2018. 1
- [13] T. Li, D. P. Lun, Y.-H. Chan *et al.*, “Robust Reflection Removal based on Light Field Imaging,” *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1798–1812, 2018. 1
- [14] P. Christian and W. Lennart, “Raytrix: Light Field technology,” <https://raytrix.de>, [Accessed 19-July-2019]. 1
- [15] X. Guo, Y. Li, and H. Ling, “LIME: Low-Light Image Enhancement via Illumination Map Estimation,” *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 982–993, 2017. 2, 3
- [16] S. Park, S. Yu, B. Moon, S. Ko, and J. Paik, “Low-light image enhancement using variational optimization-based retinex model,” *IEEE Trans. Consumer Electronics*, vol. 63, no. 2, pp. 178–184, 2017. 2, 3
- [17] R. Wang, Q. Zhang, C.-W. Fu, X. Shen, W.-S. Zheng, and J. Jia, “Underexposed Photo Enhancement Using Deep Illumination Estimation,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2, 3
- [18] W. Ren, S. Liu, L. Ma, Q. Xu, X. Xu, X. Cao, J. Du, and M.-H. Yang, “Low-light Image Enhancement via a Deep Hybrid Network,” *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4364–4375, 2019. 2, 3
- [19] C. Chen, Q. Chen, J. Xu, and V. Koltun, “Learning to See in the Dark,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 2, 3, 5, 7, 8, 9, 11, 12, 13, 14
- [20] K. G. Lore, A. Akintayo, and S. Sarkar, “LLNet: A Deep Autoencoder Approach to Natural Low-Light Image Enhancement,” *Pattern Recognition*, vol. 61, pp. 650–662, 2017. 2, 3
- [21] T. Remez, O. Litany, R. Giryes, and A. M. Bronstein, “Deep Convolutional Denoising of Low-Light Images,” *arXiv:1701.01687*, 2017. 2, 3
- [22] Y. Guo, X. Ke, J. Ma, and J. Zhang, “A Pipeline Neural Network for Low-Light Image Enhancement,” *IEEE Access*, vol. 7, pp. 13 737–13 744, 2019. 2, 3
- [23] T. Plötz and S. Roth, “Benchmarking Denoising Algorithms With Real Photographs,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2
- [24] S. Wanner and B. Goldluecke, “Spatial and Angular Variational Super-Resolution of 4D Light Fields,” in *Eur. Conf. Comput. Vis.* Springer, 2012, pp. 608–621. 2, 4, 5
- [25] Y. Wang, F. Liu, K. Zhang, G. Hou, Z. Sun, and T. Tan, “LFNet: A Novel Bidirectional Recurrent Convolutional Neural Network for Light-Field Image Super-Resolution,” *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4274–4286, 2018. 2, 4, 5, 7
- [26] S. Zhang, Y. Lin, and H. Sheng, “Residual Networks for Light Field Image Super-Resolution,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 2, 4, 5, 7
- [27] 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 *Int. Conf. Comput. Vis. Workshop*, 2015. 2, 4, 5
- [28] J. S. Lumentut, T. H. Kim, R. Ramamoorthi, and I. K. Park, “Fast and Full-Resolution Light Field Deblurring using a Deep Neural Network,” *arXiv:1904.00352*, 2019. 2
- [29] P. P. Srinivasan, R. Ng, and R. Ramamoorthi, “Light Field Blind Motion Deblurring,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2
- [30] D. Lee, H. Park, I. Kyu Park, and K. Mu Lee, “Joint Blind Motion Deblurring and Depth Estimation of Light Field,” in *Proceedings of the Eur. Conf. Comput. Vis.*, 2018. 2
- [31] K. Mitra and A. Veeraraghavan, “Light Field Denoising, Light Field Superresolution and Stereo Camera Based Refocussing Using a GMM Light Field Patch Prior,” in *IEEE Comput. Vis. Pattern Recog. Workshops*, 2012. 2
- [32] M. Alain and A. Smolic, “Light Field Denoising by Sparse 5D Transform Domain Collaborative Filtering,” in *Int. Workshop on Multimedia Signal Processing*, 2017, pp. 1–6. 2, 3, 7, 8
- [33] 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 *Computational Imaging XI*, vol. 8657. Int. Society for Optics and Photonics, 2013, p. 86570P. 2
- [34] A. Sepas-Moghaddam, P. L. Correia, and F. Pereira, “Light Field Denoising: Exploiting the Redundancy of an Epipolar Sequence Representation,” in *True Vis. Capture, Transmission and Display of 3D Video (3DTV-CON)*, 2016, pp. 1–4. 2
- [35] D. G. Dansereau, O. Pizarro, and S. B. Williams, “Decoding, Calibration and Rectification for Lenslet-Based Plenoptic Cameras,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013. 2, 3, 7
- [36] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering,” *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007. 2
- [37] E. H. Land, “The Retinex Theory of Color Vision,” *Scientific American*, vol. 237, no. 6, pp. 108–129, 1977. 3
- [38] Z. Ying, G. Li, Y. Ren, R. Wang, and W. Wang, “A New Low-Light Image Enhancement Algorithm using Camera Response Model,” in *IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3015–3022. 3
- [39] S. Gu, Y. Li, L. V. Gool, and R. Timofte, “Self-guided network for fast image denoising,” in *IEEE Int. Conf. Comput. Vis.*, 2019. 3, 6, 7, 8, 9, 14
- [40] P. Maharjan, L. Li, Z. Li, N. Xu, C. Ma, and Y. Li, “Improving extreme low-light image denoising via residual learning,” in *Int. Conf. Multimedia and Expo*, 2019. 3, 7, 8, 9
- [41] W. Chen, W. Wenjing, Y. Wenhan, and L. Jiaying, “Deep retinex decomposition for low-light enhancement,” in *British Machine Vision Conference*, 2018. 3, 7, 8, 9
- [42] Q. Zhang, G. Yuan, C. Xiao, L. Zhu, and W.-S. Zheng, “High-quality exposure correction of underexposed photos,” in *ACM Int. Conf. Multimedia*, 2018. 3, 7, 8, 9
- [43] X. Fu, D. Zeng, Y. Huang, Y. Liao, X. Ding, and J. Paisley, “A fusion-based enhancing method for weakly illuminated images,” *Signal Processing*, vol. 129, pp. 82–96, 2016. 3
- [44] Y. Zhang, J. Zhang, and X. Guo, “Kindling the darkness: A practical low-light image enhancer,” in *ACM Int. Conf. Multimedia*, 2019. 3
- [45] C. Li, J. Guo, F. Porikli, and Y. Pang, “Lightnet: a convolutional neural network for weakly illuminated image enhancement,” *Pattern Recognition Letters*, vol. 104, pp. 15–22, 2018. 3
- [46] H. Lee, K. Sohn, and D. Min, “Unsupervised low-light image enhancement using bright channel prior,” *Signal Processing Letters*, vol. 27, pp. 251–255, 2020. 3
- [47] C.-K. Liang and R. Ramamoorthi, “A Light Transport Framework for Lenslet Light Field Cameras,” *ACM Trans. Graph.*, vol. 34, no. 2, p. 16, 2015. 4
- [48] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 5
- [49] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, “Image Super-Resolution Using Very Deep Residual Channel Attention Networks,” in *Eur. Conf. Comput. Vis.*, 2018, pp. 286–301. 5
- [50] K. Zhang, M. Sun, T. X. Han, X. Yuan, L. Guo, and T. Liu, “Residual Networks of Residual Networks: Multilevel Residual Networks,” *IEEE Trans. Circuit Syst. Video Technol.*, vol. 28, no. 6, pp. 1303–1314, 2017. 5
- [51] Q. Yan, D. Gong, Q. Shi, A. v. d. Hengel, C. Shen, I. Reid, and Y. Zhang, “Attention-Guided Network for Ghost-Free High Dynamic Range Imaging,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 5
- [52] R. Mechrez, I. Talmi, and L. Zelnik-Manor, “The Contextual Loss for Image Transformation with Non-Aligned Data,” in *Eur. Conf. Comput. Vis.*, 2018. 6, 7
- [53] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 6

- [54] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, “Shufflenet v2: Practical guidelines for efficient cnn architecture design,” in *Eur. Conf. Comput. Vis.*, 2018. [7](#)
- [55] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy, “Burst photography for high dynamic range and low-light imaging on mobile cameras,” *ACM Trans. Graph. (TOG)*, vol. 35, no. 6, p. 192, 2016. [7](#)
- [56] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Int. Conf. Learn. Represent.*, 2015. [7](#)
- [57] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a “Completely Blind” Image Quality Analyzer,” *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2012. [11](#)
- [58] A. Mittal, A. K. Moorthy, and A. C. Bovik, “No-Reference Image Quality Assessment in the Spatial Domain,” *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4695–4708, 2012. [11](#)
- [59] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004. [16](#)