Demosaicing and Denoising on Simulated Light Field Images

Trisha Lian
Stanford University
tlian@stanford.edu

Kyle Chiang
Stanford University
kchiang@stanford.edu

Abstract

Light field cameras use an array of microlens to capture the 4D radiance of as scene. Standard image processing techniques with light field data do not utilize all four dimensions to demosaic or denoise captured images. In this paper, we formulate demosaicing as an optimization problem and enforce a TV-prior on different dimensions of the light field. We apply our method on simulated light field data created from 3D virtual scenes. Because our data is simulated, we can use ground truth images to evaluate the effectiveness of our method. For certain combinations of dimensions, we achieve better overall PSNR values than the standard demosaicing technique described in Malvar et al. [1]. Despite the improvement in PSNR, we introduce more color artifacts in areas of high frequency in the image. Our method also improves PNSR values for scenes with low illumination levels.

1. Introduction

1.1. Background

Unlike standard cameras, light field cameras ("plenoptic" cameras) uniquely capture the 4D radiance information of a scene instead of just a 2D intensity image. This is achieved by inserting a microlens array between the camera's main lens and sensor. Each microlens separates incoming rays and allows the sensor to capture both the intensity of a ray as well as the angle from which it arrived (see Figure 1). Each ray can be characterized by its intersection with the microlens plane (s, t) and the main lens (u, v). These four coordinates make up the four dimensions of the light field data: $L(u, v, s, t)$. The 4D data can be post-processed to dynamically change the depth of field and focal plane of the image after it has been acquired. In this paper, we utilize all four dimensions to help improve the demosaicing and denoising steps in the image processing pipeline.

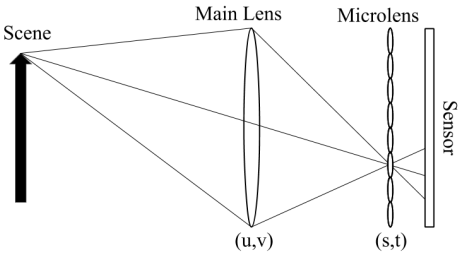


Figure 1: A schematic of a light field camera. Each ray can be uniquely characterized by its intersection with the main lens, (u, v) coordinates, and the microlens array, (s, t) coordinates.

1.2. Motivation

Standard demosaicing techniques demosaic the Bayer pattern output directly from the camera sensor. For a typical camera, this is the optimal strategy. However, for a light field camera, the microlens array encodes additional information in the sensor image. Demosaicing using traditional techniques ignores this additional information. The objective of our new optimization technique for demosaicing is to try to capture and use all four dimensions when generating the full-color light field.

Not much work has been done in utilizing this extra information in light field data. Some researchers [2] have proposed projecting samples of the microlens to the refocus plane before demosaicing. To avoid the random RGB sampling that results from this, the authors resample the radiance according to the parameters of the focal plane in order to achieve even samples for demosaicing. With this method, the authors claim to visually achieve reduced demosaicing artifacts and more detail. Other demosaicing methods use disparity [3] or machine learning [4] to improve color quality. Our method approaches this problem using optimization techniques and uses simulation data to quantify its effectiveness.

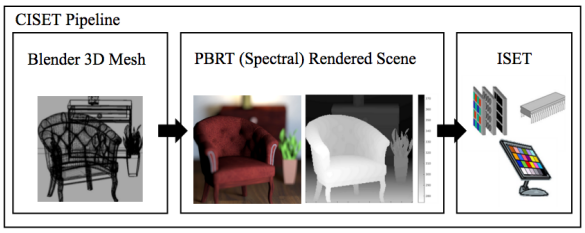


Figure 2: A diagram of our camera simulation pipeline. For the light field simulation, we model lenses in PBRT to match a light field camera. The rays are therefore traced through both a main lens and a microlens array.

2. Light Field Simulation

In order to test our method against a ground truth image, we use a light field camera simulation currently being developed by one of the authors. This simulation steps through the entire camera pipeline to generate realistic data: from a 3D virtual scene, through the optics of a light field camera, and onto a sensor. To generate the ground truth image, we sample the image with a simulated sensor that has RGB filters at every pixel and no noise parameters.

2.1. Simulation Pipeline

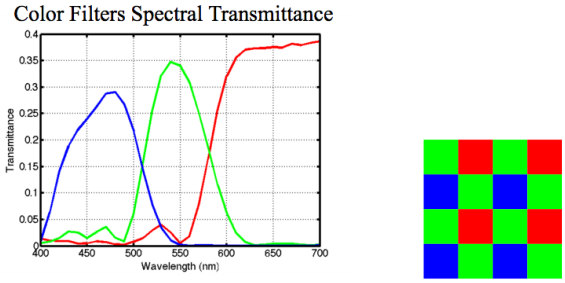
Figure 2 summarizes the main steps of the simulation. The simulation starts with a virtual scene created in a 3D modeling program such as Blender or Maya. This scene includes the geometry and material properties of the objects as well as the positions and shapes of lights. Next, a modified version of PBRT [5] is used to trace rays from the sensor, through light field optics (microlens and main lens), and into the scene. PBRT has been modified in this simulation to apply full-spectral rendering. During the ray-tracing step, the user specifies simulation parameters such as lens types, spectral properties of the light sources, film distance, aperture size, and field of view. The simulation also accounts for realistic lens properties such as chromatic aberration and potential diffraction limited systems.

Once all these parameters are specified, the resulting "optical image" is passed on to ISET (Image Systems Engineering Toolbox) [6]. ISET takes the incoming light information and captures it with a realistic sensor. The user can specify the sensor parameters, such as the Bayer pattern, pixel size, sensor size, exposure time, noise parameters, and illumination levels. The sensor data we obtain from the end of this pipeline is our raw data.

2.2. Simulation Parameters

For the data obtained in this paper, we simulated a camera with a 50 mm double gaussian main lens with an aperture setting of f/2.8. The camera had a 500 x 500 microlens array in front of the sensor. The location and size of the array was automatically calculated to cover as many sensor pixels as possible without overlap [7], and therefore has an f-number that matches the main lens. Each microlens covers a 9x9 block of sensor pixels, so we capture 81 different angular views of our scene.

The sensor size was 6.7 mm x 6.7 mm with a pixel size of 1.7 x 1.7 um. The resolution of the raw sensor image was 4500 x 4500 pixels and the resolution of the final image was equal to the number of microlenses (500 x 500). The exposure time was set to 1/90 s. Our Bayer pattern had an "grbg" configuration. See Figure 3 for the transmittance of the three color filters on our simulated sensor. ISET also included shot and electronic noise in the simulated sensor.



(a) (b)

Figure 3: a) Transmittance plots of the color filters on our simulated sensor. b) Bayer pattern used to obtain our raw data.

We render two different scenes. Both are lit with D65 illuminant area lights. One scene contains a chair and a house plant, while the other contains three planar resolution charts at varying distances. The objects in the scene are around 0.5 to 1.5 m away from the camera.

3. Methods

3.1. Baseline - Malvar et al.

As a baseline comparison to determine the effectiveness of our new method, we implemented a standard demosaicing algorithm on the raw sensor image. The method we decided to use as a baseline is described in Malvar et al [1]. Because the method performs demosaicing using a linear transformation, it can be implemented using 2D convolutions and computed very efficiently. Furthermore, this method produces very few color artifacts for a typical image. These artifacts only show up in areas of high frequency.

3.2. Optimization Problem

For our optimization problem, we wanted to find the most likely 4D light field image that would produce the Bayer filtered image captured by the camera. However,

due to the loss of information when sampling the scene, there are an infinite number of images that could produce the same Bayer filtered image. To choose the most likely image, we note that real world images tend to have sparse gradients and assume an anisotropic TV prior on the 4D image. The optimization problem can then be formulated as follows:

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|Dx\|_1$$

where A is the sampling matrix that generates a Bayer filtered image from the scene, b is the Bayer filtered image captured by the camera, D is the gradient function, and λ is a parameter chosen to weight the TV prior. This approach was inspired by the techniques described in Heide et al. [8].

3.3. Choice of Gradients

For a 2D image, the TV prior would be the sum of the gradient in the X direction and the gradient in the Y direction. However, for our 4D light field data, the TV prior has some ambiguity. There are 2 assumptions made for sparse gradients. The first being that each image captured from slightly different angles should be nearly identical. This would be enforced in the TV prior by setting the gradient function D to the gradient in s and t. The second assumption is that corresponding pixels in images seen through each microlenses should also very similar. To enforce this in the TV prior, D would be set to the gradient in the u and v directions. We chose to investigate these two assumptions both separately and together by looking at 3 cases: sparse gradients in u and v only, sparse gradients in s and t only, and sparse gradients in u, v, s and t.

3.4. ADMM

We solved this optimization problem using an iterative ADMM method. To implement this method, we first reformulated the optimization problem in the form:

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|z\|_1$$

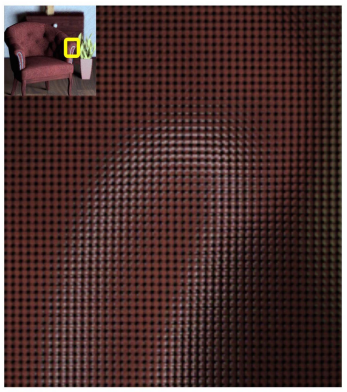
$$\text{subject to } Dx - z = 0$$

Using the ADMM strategy we then form the augmented Lagrangian

$$L_\rho(x, y, z) = \frac{1}{2}\|Ax-b\|_2^2+\lambda\|z\|_1+y^T(Dx-z)+\frac{\rho}{2}\|Dx-z\|_2^2$$

The iterative ADMM updates can then be derived as follows:

$$x \leftarrow \left(A^TA + \rho D^TD\right)^{-1}\left(A^Tb + \rho D^T(z-u)\right)$$



(a)

(b)

Figure 4: (a) An enlarged portion of the image captured by the sensor. (b) By restructuring the sensor data, we can display the image in this tiled form. Each image corresponds to a single *(u,v)* index.

$$z \leftarrow \begin{cases} v-\kappa & v > \kappa \\ 0 & |v| \le \kappa \\ v+\kappa & v < -\kappa \end{cases} \text{ for } v = Dx + u \text{ and } \kappa = \lambda/\rho$$

$$u \leftarrow u + Dx - z$$

These update rules are repeated until convergence or until the maximum number of iterations is reached.

3.5. Image Processing Pipeline

It is important to note that we only carry our simulation past the demosaicing portion of the image processing pipeline. We do not perform any gamut mapping, white balancing, or illuminant correction. We chose to do this in order to target the effectiveness of demosaicing and denoising with our method and to not confound our results with processes further down the pipeline. As a result of this purposefully incomplete processing, our images look tinted compared to the original scene.

Figure 4 shows a visualization of the 4D ground truth light field. As described earlier, we produce these ground truth images by capturing the rendered "optical image" with a full array sensor in ISET. This sensor has color filters for every pixel and has its noise parameters turned off. This image will serve as the reference for all PSNR calculations.

3.6. Gradients in Ground Truth

In Figure 5, we calculate and plot the gradients of one of our ground truth image in each of the different light field dimensions. The gradients in these images are mostly dark, which indicates that the gradients are indeed sparse and that the TV assumptions should improve the resulting image.The gradients are more sparse in (u, v), than in (s, t). This is particularly true for the in-focus plane in the center of the image. We would therefore expect our method to perform the best when we assume sparse gradients in the (u, v) dimension.

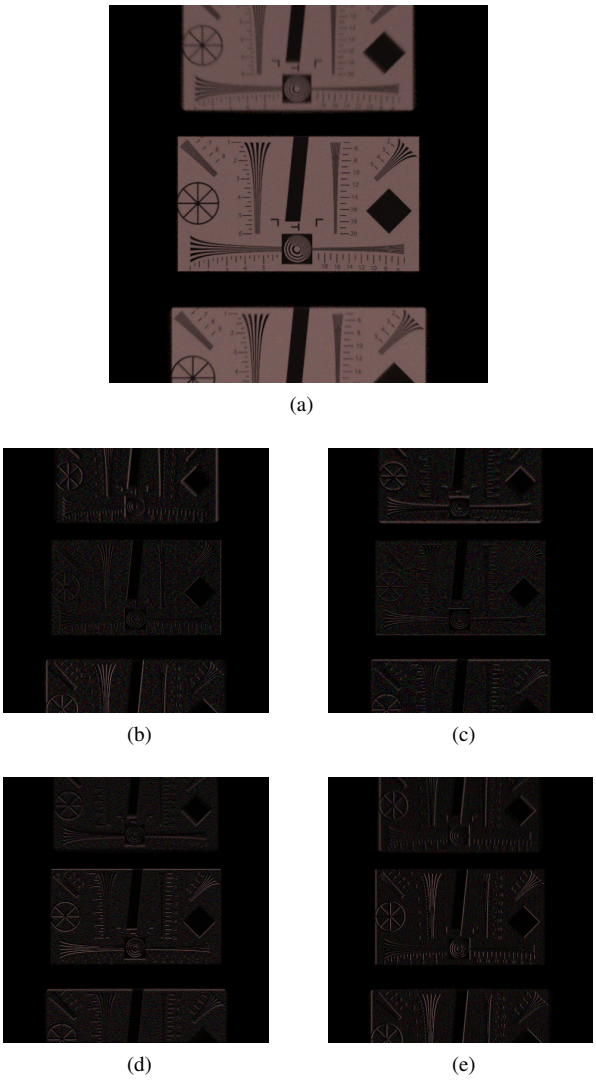
4. Results

For all results, we demosaic our raw data using 1) Malvar et al.'s method and 2) our optimization method. For our method, we try three different TV-priors as described above: a) gradients over (u, v), b) gradients over (s, t), and c) gradients over (u, v, s, t).

4.1. Average Illumination

For the following results, we set the mean illuminance level to be 12 lux. The maximum illuminance for each image is roughly 70 lux, which is equivalent to standard room lighting. The images shown are taken from the center sub-aperture (u = 0, v = 0). In other words, it is equivalent to the center tile when you display the data as shown in Figure 4b. By shifting and adding these different tiled images, the user obtains different depth of fields. We calculate PSNR values for both the center sub-aperture image and the mean image (average over all (u, v)).

Figure 6 and Figure 7 shows the demosaiced images of our two scenes. The differences (averaged across the color channels) between each image and the ground truth are shown as well. We can see that most errors are centered around the high frequency components of the image, and these errors are higher for our method compared to Malvar et al. Although these errors are difficult to see in the full image, enlarging high frequency sections of the image (Figure 8 and Figure 9) reveals color artifacts for our optimization method. Despite introducing color artifacts, our (u, v) and (u, v, s, t) methods result in higher overall PSNR values than Malvar et al (see Tables 1 and 2).



(a) (b) (c) (d) (e)

Figure 5: Gradients taken in each of the four light field dimensions. (a) Ground Truth (b) u (c) v (d) s (e) t

	Malvar	(u,v)	(s,t)	(u,v,s,t)
Center	35.40 dB	37.26 dB	34.69 dB	36.55 dB
Mean Image	34.65 dB	37.56 dB	32.97 dB	35.38 dB

Table 1: PSNR values for the "Chair" scene.

	Malvar	(u,v)	(s,t)	(u,v,s,t)
Center	28.87 dB	28.07 dB	28.38 dB	28.88 dB
Mean Image	28.05 dB	29.01 dB	26.18 dB	28.00 dB

Table 2: PSNR values for the "Resolution Charts" scene.

4.2. Changing Illumination Levels

Because we assume sparse gradients in the image, our method should perform better under noisier conditions.To

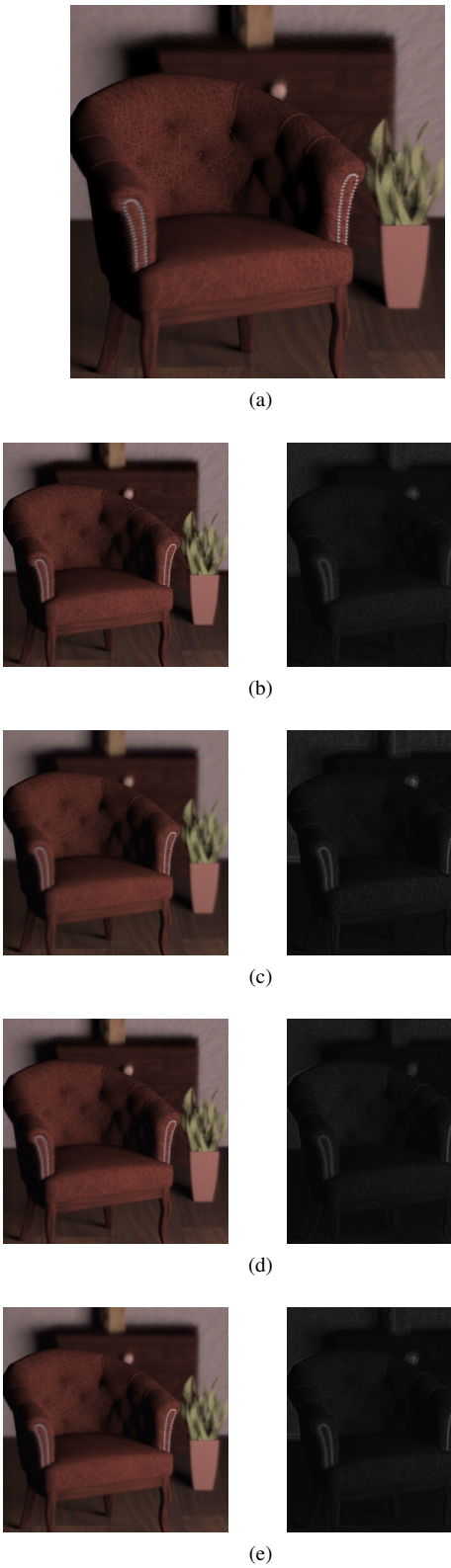


Figure 6: Demosaiced scene of chair [left] along with a visualization of error relative to ground truth [right] (a) Ground truth image. (b) Malvar et al. (c) (u, v). (d) (s, t). (e) (u, v, s, t).

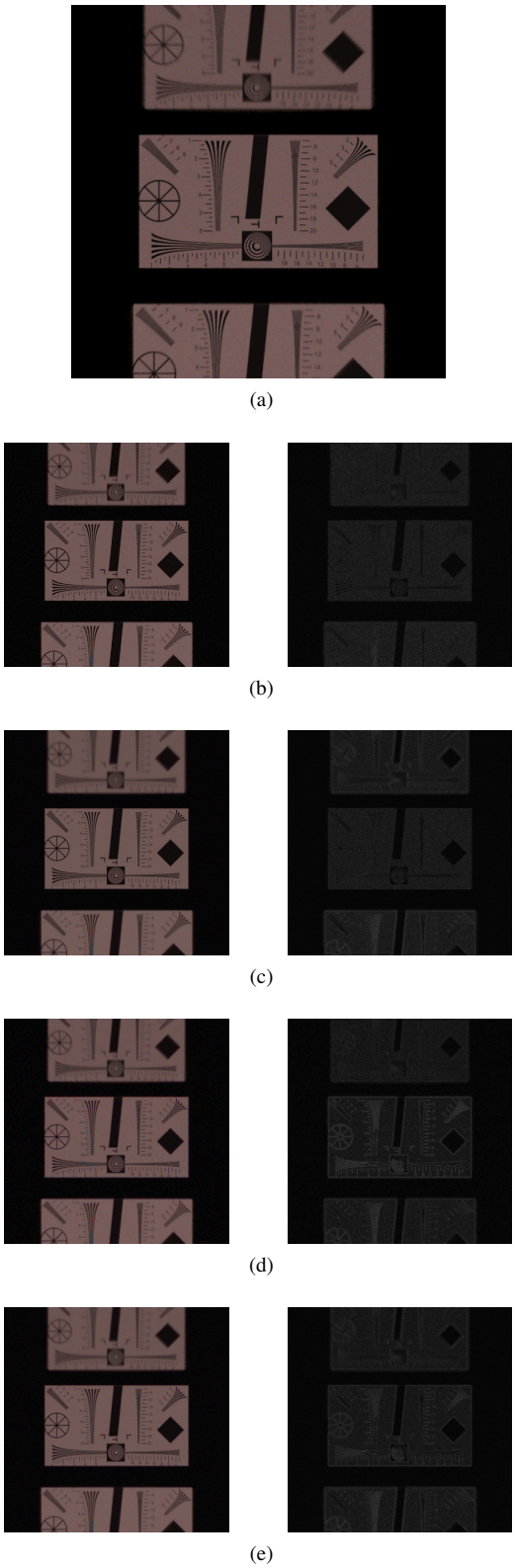


Figure 7: Demosaiced image of resolution charts [left] along with a visualization of error relative to ground truth [right] (a) Ground truth image. (b) Malvar et al. (c) (u, v). (d) (s, t). (e) (u, v, s, t).

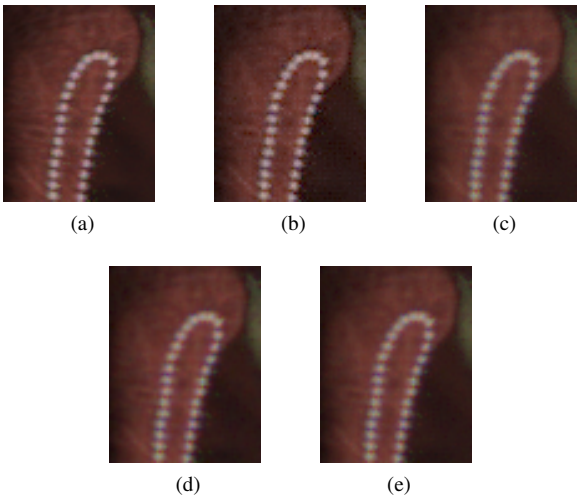


Figure 8: A enlarged section of arm of the chair. (a) Ground truth image. (b) Malvar et al. (c) *(u,v)*. (d) *(s,t)*. (e) *(u,v,s,t)*.

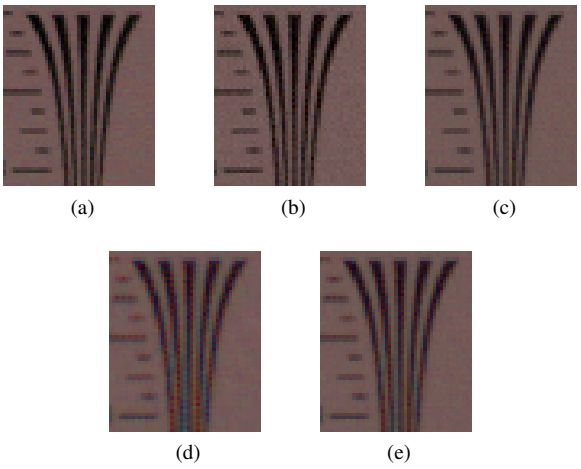


Figure 9: A enlarged section of resolution chart. (a) Ground truth image. (b) Malvar et al. (c) *(u,v)*. (d) *(s,t)*. (e) *(u,v,s,t)*.

test this, we rendered our raw data under different sensor illumination levels in ISET. Lower illumination results in noisier images. Figure 10 and Figure 11 show our results.

From the plot, we can see that our (u, v) and (u, v, s, t) methods perform better than Malvar et al. for very low illumination. This is because this baseline technique performs no denoising, while our assumption of sparse gradients automatically smooths out noise. Linear demosaicing (such as Malvar et al.'s method) is greatly affected by noise, which is why many image processing pipeline perform denoising before demosaicing.

As illumination levels increase, our (u, v) technique continues to outperform Malvar et al.'s method in terms of overall image PSNR. (s, t) performs poorly regardless of the illumination. The assumption of sparse gradients in this dimension may not be very strong, which is supported by the number of gradients seen in Figure 5.

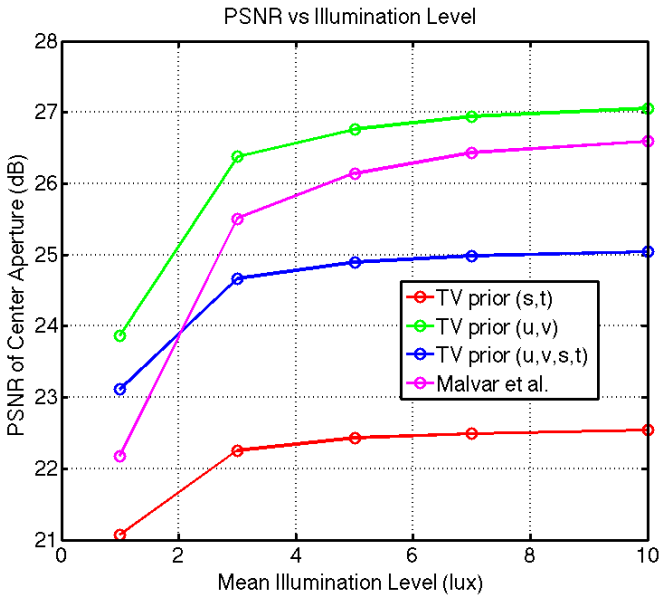


Figure 10: PSNR values for different illumination levels.

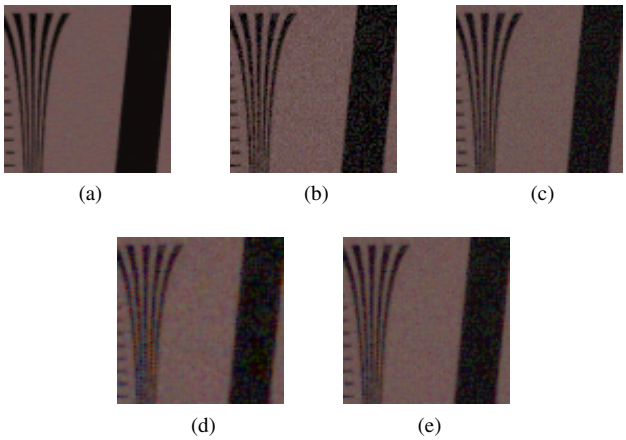


Figure 11: A comparison of how each technique performs on a noisy image (mean illumination = 1 lux). (a) Ground truth image .(b) Malvar et al. (c) (u, v). (d) (s, t). (e) (u, v, s, t)

.

5. Conclusion

In conclusion, our demosaicing method, which solves an optimization problem, results in an image with better PSNR values than the traditional method when we assume sparse gradients in the (u, v) or (u, v, s, t) dimensions. However, for images with good lighting, we ended up with more color artifacts than demosaicing with traditional methods in areas of high frequencies. We suspect that this is due to the fact that while the TV prior helps create a truer overall image, it is specifically avoiding high frequency signals in the image, resulting in color artifacts in these regions. However, for images with poor illumination or lots of noise, the advantages of running optimization with a TV prior shine, with the best results assuming sparse gradients across u and v.

5.1. Future Work

While the solution we investigated may not be the optimal demosaicing for a light field camera in all cases, there are several other possible directions to pursue to try to harness the information of the 4D light field to obtain the best demosaiced image. One possible improvement is to look at a cross channel prior that also penalizes the difference in gradients between the color channels. In most images sharp edges result in gradients in all 3 color channels, so enforcing this assumption could result in fewer color artifacts. Another possible route for investigation is to use the true image to train an optimal linear transform similar to the one presented in Malvar et al, extended to 4 dimensions.

References

[1] Malvar, Henrique S., Li-wei He, and Ross Cutler. "High-quality linear interpolation for demosaicing of Bayer-patterned color images." Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on. Vol. 3. IEEE, 2004.

[2] Yu, Zhan, et al. "An analysis of color demosaicing in plenoptic cameras." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

[3] Seifi, Mozhdeh, et al. "Disparity guided demosaicking of light field images." Image Processing (ICIP), 2014 IEEE International Conference on. IEEE, 2014.

[4] Huang, Xiang, and Oliver Cossairt. "Dictionary learning based color demosaicing for plenoptic cameras." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2014.

[5] Pharr, Matt, and Greg Humphreys. Physically based rendering: From theory to implementation. Morgan Kaufmann, 2004.

[6] Farrell, Joyce, et al. "A display simulation toolbox for image quality evaluation." Journal of Display Technology 4.2 (2008): 262-270

[7] Ng, Ren, et al. "Light field photography with a hand-held plenoptic camera." Computer Science Technical Report CSTR 2.11 (2005): 1-11.

[8] Heide, Felix, et al. "FlexISP: a flexible camera image processing framework." ACM Transactions on Graphics (TOG) 33.6 (2014): 231.