Disparity Estimation and Image Fusion with Dual Camera Phone Imagery

Rose Rustowicz
Stanford University
Stanford, CA
rose.rustowicz@gmail.com

Abstract

This project explores computational imaging and optimization methods in the image formation model. More specifically, disparity estimation and image fusion are performed on an input pair of small baseline stereo-images. Synthetic refocusing is also implemented to show an application of the outputs. A Huawei Mate 9 Pro mobile phone is used in experiments to demonstrate results.

1. Introduction and Related Work

The image signal processing (ISP) pipeline refers to the processing steps that are applied to a sensors raw output to yield a processed image. These steps may include some combination of illumination correction, demosaicing, image sharpening, depth estimation, and so on. ISPs for dual or multi-camera modules integrate information from multiple sensors to construct these processed images.

Although this standard implementation of the image signal processing pipeline is widely used to process images from today's sensors, it does have its drawbacks. The error introduced at each step of the pipeline is propagated through to the final image. Depending on the accuracy of each processing step, the output image may contain significant sources of error. Additionally, ISPs may become very complex and cumbersome. Each processing step adds to the complexity of the pipeline and may contribute error to the final image.

Heide *et al*. [10] showed that computational imaging and optimization methods could be used in lieu of standard ISPs to solve for a system's output processed image(s) by formulating the complicated ISP pipelines within an optimization problem. This idea can be seen in figure 2, taken from the FlexISP paper. Concepts from [10] were implemented into ProxImaL [9], a domain-specific language and compiler for solving image optimization algorithms, which was shown to generalize to a variety of tasks. Tang *et al*. [17] extend this work for image construction in RGB-IR sensors, where they jointly address channel deblurring, channel separation, and pixel demosaicing. Recent approaches explore deep learning for learning image priors for image classification [7] and in denoising, deblurring, and image reconstruction [6].

In this project, computational imaging and optimization methods are employed to implement different parts of the ISP pipeline for a dual camera phone. In particular, point spread function (PSF) estimation and deconvolution, disparity map estimation, and image fusion are explored. These processes take in a stereo image pair and yield an output disparity map and fused RGB image. Prior information can be incorporated into the optimization problem in order to constrain the output. For example, the PSF of the lenses can provide helpful information for deblurring (used in image reconstruction), while extrinsic parameters between the cameras can be used to rectify the stereo image pair onto parallel image planes (used in disparity estimation). Natural image priors such as smoothness and sparse gradients can also be used to constrain the output. To explore applications of these outputs, a synthetic refocusing algorithm is also implemented.

2. Mobile Phone Sensor Description

Raw imagery from the Huawei Mate 9 Pro dual camera phone is used in the project. At this stage, processing is performed externally on a computer. The dual camera phone simultaneously captures images from two sensors. The first sensor, a panchromatic sensor, integrates wavelengths over the visible spectrum (from 400nm to 700nm) and has a high spatial resolution of 5120×3840 pixels. The second sensor, a color sensor, uses a Bayer color filter array that captures long (red), middle (green), and short (blue) wavelengths in the visible spectrum. The spatial resolution of this sensor is slightly less (3968×2976 pixels), but the aspect ratio of the two sensor is the same. Because the sensors have a different number of pixels, different wavelength sensitivities, and different orientations in world coordinates, alignment is not trivial. Thus, alignment is incorporated into the optimization procedure to yield a final color image that is a fused result of the original two. An image of the dual camera phone sensors is shown in figure 3.

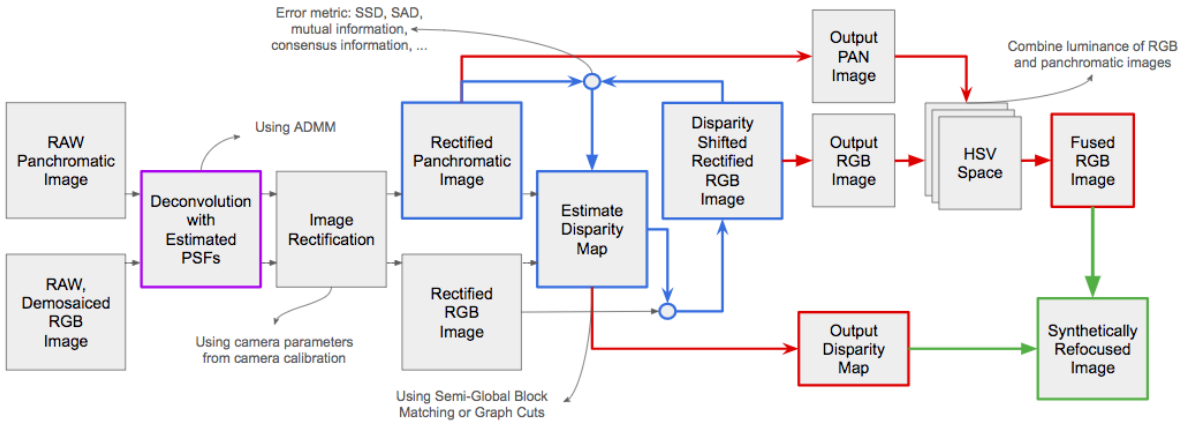

Figure 1. The high level overview of the approach. The three main parts of the implementation are 1.) PSF estimation, 2.) disparity estimation and image fusion, and 3.) synthetic refocusing. See the 'Technical Approach' section for more information on each of these components.

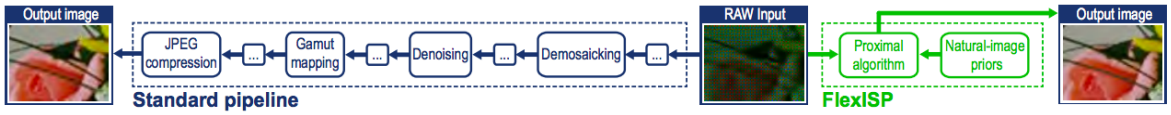

Figure 2. This figure is taken from the FlexISP paper [10]. Standard ISPs (left) perform image reconstruction in a step-by-step pipeline that accumulates error along the way. FlexISP (right) shows that this pipeline can be formulated as an optimization problem, where the entire image formation process and latent image is solved in one step.

Figure 3. The Huawei Mate 9 Pro dual camera phone, composed of two vertically aligned sensors.

3. Technical Approach

To obtain prior information on the system, the dual cameras are geometrically calibrated following the method in [11] and [18]. This work was completed last quarter, and utilized the Matlab Stereo Calibration toolbox. Spatially-varying Point Spread Functions (PSFs) are also found for each color channel, following the approach by Mosleh *et al*. [15], which was completed as part of this project. Within this method, an alignment between two image spaces is implemented, and an additional optimization problem is used to solve for the PSFs. To solve for a disparity map and fused color image, an iterative approach with graph cuts is deployed, and other disparity map estimation techniques are also explored. The output disparity map and fused image are used as inputs to a synthetic refocusing algorithm, where each pixel is blurred based on it's disparity value. More details on each of the project components are outlined below. Figure 1 provides an outline of the entire approach.

3.1. PSF Estimation and Deconvolution

PSFs of the camera lenses can be used in image reconstruction for deblurring through deconvolution. In a naive approach, the unblurred, latent image can be solved for using inverse filtering (1):

$$u = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(b)}{\mathcal{F}(k)}\right) \tag{1}$$

, where u is the unblurred, latent image, b is the blurred image, k is the PSF kernel, $\mathcal{F}$ denotes the Fourier transform, and $\mathcal{F}^{-1}$ denotes the inverse Fourier transform. When there is noise in the system, divisions close to zero in the Fourier domain drastically amplify noise. To suppress this effect, Wiener filtering can be used for deconvolution (2):

$$u = \mathcal{F}^{-1}\left(\frac{|\mathcal{F}(k)|^2}{|\mathcal{F}(k)|^2 + 1/SNR} \cdot \frac{\mathcal{F}(b)}{\mathcal{F}(k)}\right) \quad (2)$$

, where SNR is the signal to noise ratio. Other methods, such as ADMM [3], can also be used for deblurring / deconvolution. This approach alternates between three different optimization steps to solve for the deconvolved image. In each deconvolution method mentioned here, we need an estimate of the point spread function (PSF), k in these equations, to perform the deblurring.

Following the method in [15], we are able to estimate spatially varying PSFs in a two step process that includes 1.) alignment and 2.) PSF optimization. To perform alignment, four calibration images are captured, shown in figure 4. Pristine versions of the same targets are created on the computer itself, shown in figure 5. The targets (Bernoulli noise, checkerboard, white, and black images) are displayed on a laptop screen and captured by the camera. Display of the targets on the screen prevents the need for target registration.

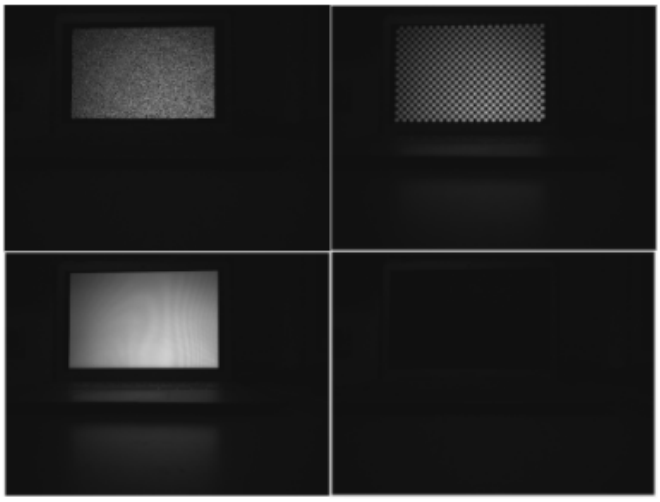

Figure 4. Calibration targets, captured with the camera.

The pristine version of the noise target (that from the computer itself) is mapped into the same space as the image of the target taken by the camera. The result is shown in the left image of figure 4. To perform this warping procedure, the following linear transformation is applied to each point in the pristine noise target (3):

$$\begin{bmatrix} 1 & u' & v' & u'v' \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} \quad (3)$$

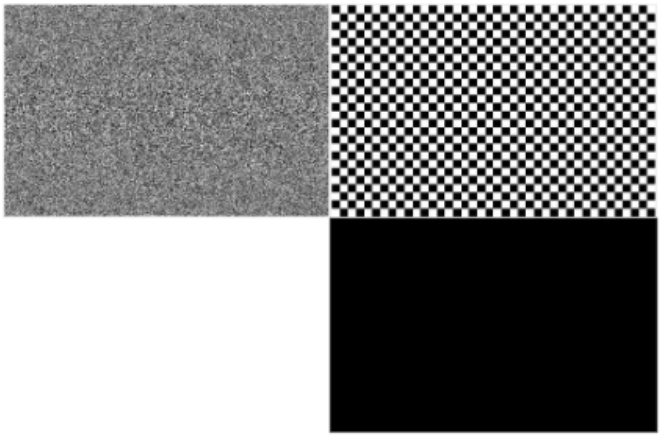

Figure 5. What is referred to here as the pristine calibration targets. The pristine Bernoulli noise target is warped into the camera image space of the noise target in figure 4 so that the PSFs can be estimated using the two targets (one blurred noise target and one warped 'pristine' noise target).

Here, (u', v') are the (x, y) coordinates in pristine space of the point being transformed, scaled between 0 and 1 within the corresponding pristine checkerboard block. x_0, y_0, x_1, y_1, x_2, y_2, and x_3, y_3, are the upper left, upper right, lower left, and lower right coordinates of corners of the checkerboard block in the camera image space. The result of this linear transformation gives the $[x, y]$ position of each pristine pixel transformed into the camera image space. For more details, please refer to [15], which also provides an outline for the warping algorithm. Note that there was a disagreement in the transformation provided in the paper, which was re-worked here to yield the results shown in this report, and that line 8 in the algorithm provided in the paper [15] should be outside of the for loops, exchanged with line 11.

After warping, the white and black target camera images are used to incorporate color effects from the image space such as vignetting into the mapped pristine image – this result is shown in the center image in figure 4. This is unconventional, as color targets are typically used to remove color effects. However, incorporating the shading into the warped pristine target makes the image as similar as possible to the noise camera image itself (which is shown in the right image of figure 4), except for the blur introduced by the lens. This process allows the pristine target to be directly compared with the lens-blurred target, perfect for PSF estimation. A zoomed in patch of the mapped and actual noise targets are shown in figure 4, in the center and left images, respectively. The central image, known as $\hat{u}$, and the right image, known as $\hat{b}$, are used to solve for the PSFs.

The second half of the PSF estimation approach is the

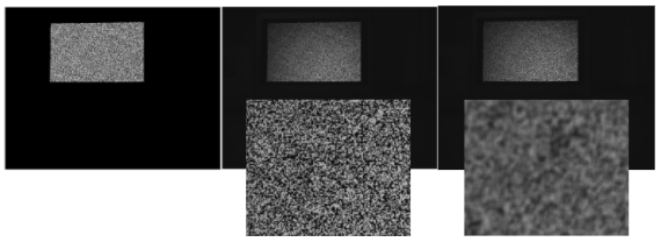

Figure 6. The left image shows the pristine image after warping into the camera image space. The central image shows the pristine image after warping and color adjustment. The right image shows the actual noise target image taken with the camera. The lower row of images show a zoomed in patch of the warped, color-corrected pristine image and the camera image, respectively. The central image, known as $\hat{u}$, and the right image, known as $\hat{b}$, are used to solve for the PSFs.

optimization procedure, in which the following objective function (4) is minimized to solve for the blur kernel, k.

$$\begin{aligned} \underset{k}{\text{minimize}} \quad & E(k) = ||\hat{u}k - \hat{b}||^2 + \lambda||k||^2 + \mu|| \bigtriangledown k||^2 \\ \text{subject to} \quad & k >= 0 \end{aligned} \tag{4}$$

The objective function is constructed with a data fidelity term (the first term), a smoothness prior (the second term) with constant λ, and a gradient prior (the third term), tuned by constant μ. The last element in the objective function from [15] was not implemented and is not shown here. Each PSF was minimized using the objective function and its derivative in the `minFunc()` function in Matlab [1].

3.2. Disparity Map Estimation and Image Fusion

Within this framework, we find a disparity map, use the disparity map to shift one of the stereo images by the disparity value at each pixel location, and then fuse the shifted image with the reference (unshifted) image. The error in the image fusion can be fed back through the pipeline to obtain an improved disparity map, which in turn yields an improved fused image. In this formulation, the disparity map is estimated in order to minimize the alignment error of the panchromatic image and the RGB image according to the estimated disparity map.

To get an initial disparity map estimate, we use the calibrated camera parameters to rectify two stereo images to the same parallel plane. Many standard disparity estimation algorithms assume input rectified images. We explore methods for disparity estimation that assume a rectified input stereo pair, including Block Matching [14], Semi-Global Block Matching [12], and Graph Cuts with alpha-expansion [5]. In particular, graph cuts is a global approach that minimizes a global energy function. Sum of absolute differences (SAD) and mutual information (MI) were used for distance metrics in graph cuts. Mutual information specifically is invariant to illumination changes, which was important given two sensors with varying spectral sensitivities. To utilize mutual information within a graph-based approach, a mutual information cost array was formed, following the approach in [16].

Here, we focus on graph cuts. The graph cuts approach consists of two main steps: 1.) filling the graph with weights and 2.) finding the minimum cut (or equivalently the maximum flow) through the weights of the graph. To construct and fill the graph, the RGB image is shifted by a discrete number of disparity values to yield an image cube, where each channel corresponds to the entire image shifted by some disparity. The average over the color channels of the disparity shifted RGB image cube is then compared to the reference panchromatic image by an error metric, such as SAD or the MI cost. This cost cube, which gives the cost value at each pixel location and for each disparity, is used to fill the graph as the weights of the data term, shown in figure 7, from [4]. The data weights are represented as the green connections in the figure, connecting the disparity labels. The smoothness term must also be specified, which yields the weights between neighboring pixels, shown in figure 7 as the orange weights between neighboring pixels. In the case of this implementation, a linear weighting scheme is used, where neighboring pixels are penalized with increasing value depending on the difference in disparity value. If neighboring pixels have the same disparity value, there is no penalty. This constrains the output map to be smooth.

In the second part of this algorithm, the minimum cut through the graph is found using the alpha-expansion algorithm [5]. The global energy function (5) to be minimized is as follows:

$$E = E_{data} + E_{smoothness} \tag{5}$$

, where E_{data} refers to the data term, and $E_{smoothness}$ refers to the smoothness term, as mentioned above. For more information on the graph cuts algorithm for stereo matching, the author of this report found [13] to be particularly helpful.

Once the final disparity map is estimated, one image is mapped into the space of the other image by shifting each pixel by it's disparity value, given by the values at each pixel of the estimated disparity map. Holes left over from the shift in disparity value can be filled with the underlying reference image values. Following this step, the shifted RGB image is transformed to hue-saturation-value (HSV) space, and the panchromatic image is fused with the value (V) image by averaging. The resulting HSV image is transformed back to the RGB color space to yield the final fused result.

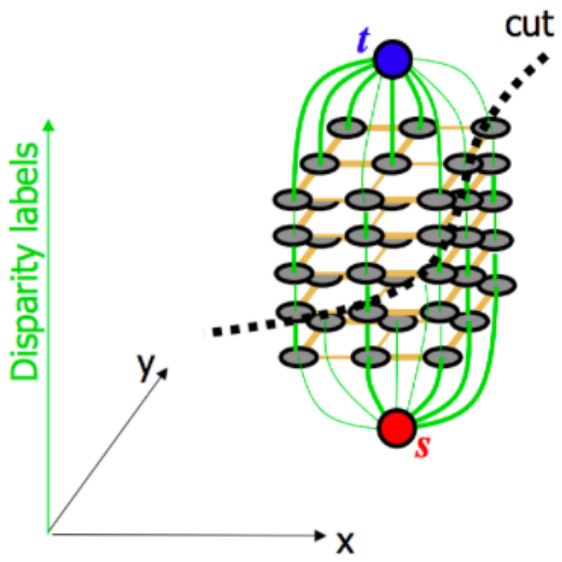

Figure 7. A visual of the graph constructed in graph cuts for multi-label energy minimization, from [4]. The green connections in the graph represent the data term cost weights, while the orange connections in the graph represent the smoothness cost weights of neighboring pixels. The smoothness cost gives incentive for neighboring disparities to be equivalent, which makes the resulting disparity estimation smooth. Costs can be formulated to be robust to object edges, where disparities should not be constant.

3.3. Synthetic Refocusing

Access to a fused color image and disparity map may enable other applications, such as synthetic refocusing. To implement this approach, the fused image is convolved with a disparity-dependent blur kernel at each pixel location.

The sigma parameter of the Gaussian is controlled by the following equation (6):

$$\sigma = C\frac{|d - d_f|}{d_f} \tag{6}$$

, where C is a constant, d_f is the disparity at which to focus and d is the disparity value at each pixel in the aligned disparity map.

In realistic images, background objects will never blur over objects that are in front of them. To handle this, a small addition can be incorporated into the synthetic refocusing algorithm. Starting at the smallest disparity level (furthest away scene objects), morphological operators can be used to dilate the pixels of the given disparity level by half the kernel size of the blur kernel. The disparity maps of all objects in front of the current disparities are then used to mask the morphological mask further, so that the objects in front of the current depth are not blurred with the convolutional kernel. This is repeated for each disparity level, and all of these images are added together to yield the final synthetically refocused image.

4. Results and Discussion

Results for PSF estimation, disparity estimation, image fusion, and synthetic refocusing are shown and discussed in the following subsections.

4.1. PSF Estimation

The PSF estimation implementation was validated by simulating results with known blur kernels and added zero mean Gaussian noise. Results from validation are shown in figure 8, where PSNR values are found for each reconstructed kernel. The first row shows the ground truth kernels, and the following rows show results outputted from the optimization procedure. The second row shows results from convolution with the kernels only, while the third and fourth rows show results from convolution and added zero mean Gaussian noise, with a variance of 0.01 and 0.1, respectively. Results are best without added noise, but become quite weary with increased noise variance, depending on the truth kernel used. To improve the approach, the SDP prior from the original paper should also be incorporated into the estimation.

To obtain blur kernels for the camera phone dual lenses, a PSF is estimated for every 128×128 block in the images from each sensor. For the Bayer sensor, PSFs are measured for each color channel after demosaicing. Calibration images displayed on the laptop screen were captured at twelve different locations to cover the entire field of view of the cameras. A subset of results from this procedure for the Bayer sensor at one of the twelve views is shown in figure 9.

Deconvolution was performed using ADMM [3], where each 128×128 camera-blurred image block was deconvolved with a corresponding 15×15 PSF kernel in the grid of estimated PSFs (as in figure 9). A small patch of the deconvolution result is shown in figure 10. The images have a green tint because they have not been white balanced here. Although the image does appear sharper, edge artifacts appear at the boundaries between image blocks. Edge handling must be implemented to improve these results. Another example of the deconvolution is shown in figure 11, where the deblurring is applied to the scene used throughout this paper. A small patch of the image is shown, as a function of ADMM parameters within the deconvolution, specified in the upper left corner of each image in the figure. The left most image is the image before deconvolution.

4.2. Disparity Estimation Methods

Prior to running the disparity estimation procedure, images were rectified using the Computer Vision system tool-

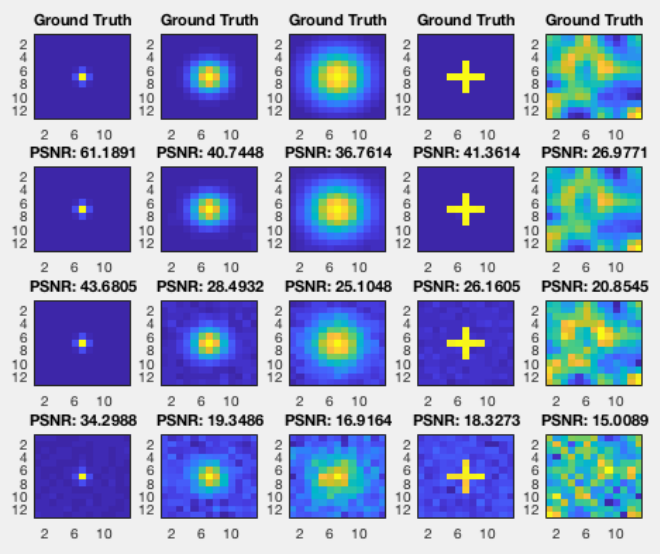

Figure 8. Validation of the PSF implementation, shown for simulated results on five ground truth blur kernels. The top row shows ground truth kernels, while the other rows show the outputs from the optimization procedure. The second, third, and fourth rows show results when the image is convolved with the kernel with no added noise, zero mean / 0.01 sigma added Gaussian noise, and zero mean / 0.1 sigma added Gaussian noise, respectively. Results degrade with increased noise, but appear relatively robust to low noise levels. Degradation also depends on the kernel itself, where more complicated kernels appear to degrade more than simple kernels. PSNR values give the PSNR of the ground truth kernel with the optimized kernel in each case.

box in Matlab, using camera calibration parameters acquired previously. A rectified image example is shown in figure 12, where the two images are shown overlapped with one another as a stereo anaglyph. The red and blue offsets show the shifts between the two views.

Disparity estimation results for block matching, semi-global block matching, and graph cuts are shown in figure 17. Block matching and semi-global block matching were implemented using OpenCV [8], while graph cuts was implemented with the GCO Python library [2].

Comparatively, block matching performs the worst on the stereo image set, with many holes in the disparity map. Semi global block matching (SGBM) performs rather well, although some holes are still present. Post processing can be performed on the SGBM result to yield a cleaned disparity map without holes. This technique uses views from both stereo images to perform consistency checks for disparity values, and was ultimately used for image fusion. The result of the post processed SGBM disparity map is shown in figure 13. This disparity map looks great, but does require the additional post processing. Other than the overestimated disparity regions in the graph cuts results, graph cuts look

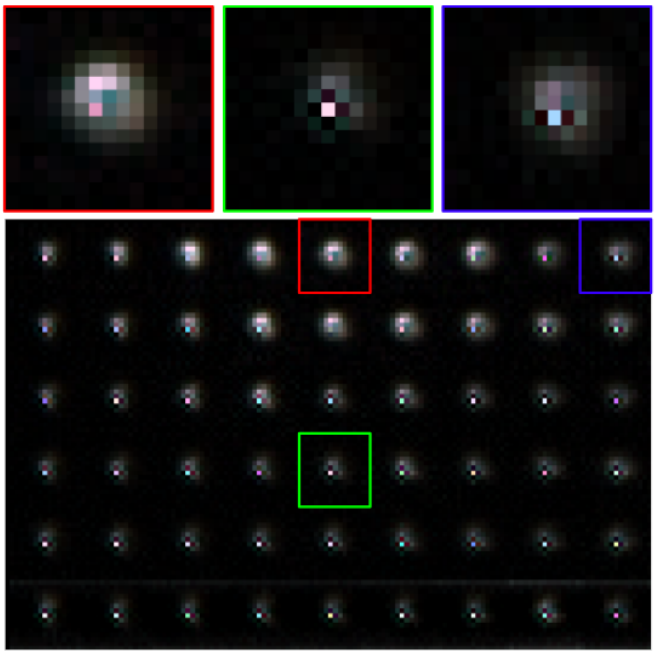

Figure 9. A subset of spatially varying point spread functions for the Bayer camera array. 15×15 pixel PSF kernels were generated for each 128×128 block in the image, across all color channels, and a subset of RGB PSFs is shown here. PSFs are normalized so that they can be better visualized.

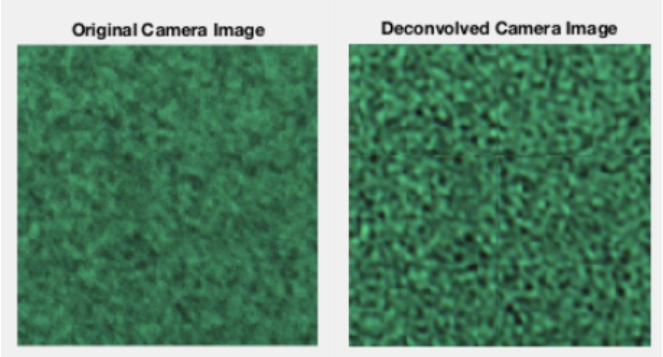

Figure 10. A subset of spatially varying point spread functions for the Bayer camera array. PSFs were generated for each 128×128 block in the image, across all color channels. The top left image shows the combined RGB PSFs, while the other three images show each color channel independently. PSFs are normalized here so that they can be better visualized.

best before post processing, as most of the image is smooth in constant depth areas, and appears robust to object edges and discontinuities. To improve results, it may be worth incorporating a confidence metric into the disparity estimation, and in exploring other distance metrics such as census information, which uses hamming distances to measure er-

Figure 11. A small patch of a deconvolved image is shown. The value in the upper left corner of each image in the figure is the value used for the ADMM parameters. The left most image is the image before deconvolution.

Figure 12. An overlapped result of two rectified images in a stereo pair. The red and blue offsets show the shifts between the two views.

ror between the shifted images.

Once the disparity map is estimated, the RGB image is shifted for image fusion. The resulting fused image is shown in figure 14, and a zoomed in patch of this image is shown in figure 15. The fused results show a decrease in noise level. Currently, the holes that remain after the disparity shift are filled with the average of the local RGB neighborhood of the pixel, since the underlying reference image is grayscale and does not provide color information. To improve upon these results, the panchromatic image could be shifted into the space of the RGB image instead, so that holes are filled with the underlying RGB image.

4.3. Synthetic Refocusing

Results from the synthetic refocusing algorithm are shown in figure 16. The top left image shows the refocused

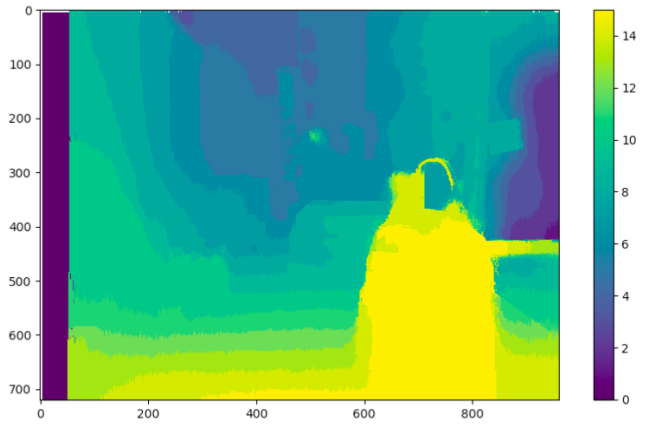

Figure 13. Post processing can be applied to the SGBM disparity map to clean the result and fill all holes, shown here.

Figure 14. Post processing can be applied to the SGBM disparity map to clean the result and fill all holes, shown here.

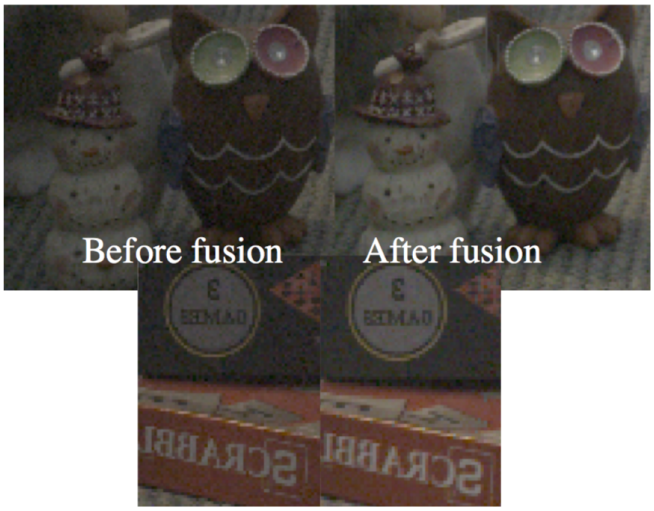

Figure 15. A zoomed in patch of the fused image from figure 14.

image at the front, where the lunch box in the image is in focus, whereas the top right image shows the pixels in the disparity map that correspond to this in-focus region. The bottom left image shows the refocused image at a middle

Figure 16. Synthetically refocused images and the corresponding in-focus pixels from the disparity map. The left column shows the refocused images (the top is focused toward the front of the scene (including the lunch box), and the bottom image is focused toward the center of the scene's depth (including the board games). The right column shows a mask of the pixels from the disparity map that were kept in focus, while all other pixels are blurred depending on their disparity value relative to the focused disparity value. The color of the scene is also blurred/focused in this algorithm, which the author left for an artistic effect.

depth in the scene, where the board games are in focus. The bottom right image shows the corresponding disparity map pixels that are in focus. This application could be used in mobile phones with dual cameras to synthetically focus an image after capture.

5. Conclusions and Future Work

Throughout this project, optimization and computational imaging techniques were explored to implement steps in the image signal processing (ISP) pipeline. PSF estimation was implemented to perform deconvolution. Disparity maps were estimated with a variety of techniques, with focus on a global energy minimization technique known as graph cuts. Images were fused with the estimated disparity results, yielding an improved, fused output image. One practical application known as synthetic refocusing was also demonstrated, which could be deployed on dual camera systems to enable a user to refocus an all-in-focus image after capture.

Several updates can be explored to improve results. With respect to PSF estimation, the SPD prior from the implemented paper can be incorporated into the approach for more noise-robust results. Edge cases must also be accounted for after deconvolution so that edges do not exist between image blocks. In regard to disparity estimation, a more sophisticated hole filling approach may be used, or the disparity map can be used to shift the panchromatic image rather than shifting the RGB image. This will naturally improve hole filling, where the underlying RGB image can be used to fill in the missing data.

6. Acknowledgements

I would like to thank Dr. Gordon Wetzstein and Dr. Don Dansereau for their advisement throughout this project. I would also like to thank Ashwini Ramamoorthy for helpful collaboration regarding implementation of the PSF estimation technique, and to Fei Xia and Kuan Fang for discussing graph cuts for disparity estimation. This project was completed for both EE367: Computational Imaging and Displays, and CS231A: Computer Vision - From 3D Reconstruction to 2D Recognition at Stanford University, during the 2018 Winter quarter. The project was completed by one student, and thus the author is responsible for the completed work.

References

[1] https://www.cs.ubc.ca/ schmidtm/software/minfunc.html.

[2] Python wrappers for gco alpha-expansion and alpha-beta-swaps. `https://github.com/amueller/gco_python`. Accessed: 2018-03-01.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers (admm). *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

[4] Y. Boykov, D. Cremers, and V. Kolmogorov. Eccv 2006 tutorial on graph cuts vs. level sets. *European Conference on Computer Vision (ECCV)*, 2006.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[6] S. Diamond, V. Sitzmann, F. Heide, and G. Wetzstein. Unrolled optimization with deep priors. *arXiv*, 2017.

[7] S. Diamond, V. Sitzmann, s. Boyd, G. Wetzstein, and F. Heide. Dirty pixels: Optimizing image classification architectures for raw sensor data. *arXiv*, 2017.

[8] A. K. G. Bradski. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Sebastopol, CA, 2008.

[9] F. Heide, S. Diamond, M. Niessner, J. Ragan-Kelley, W. Heidrich, and G. Wetzstein. Proximal: Efficient image optimization using proximal algorithms. 2016.

[10] F. Heide, M. Steinberger, Y. Tsai, M. Rouf, D. Pajak, D. Reddy, J. Liu, O. Gallo, W. Heidrich, K. Egiazarian, J. Kautz, and K. Pulli. Flexisp: A flexible camera image processing framework. *ACM Transactions of Graphics, SIGGRAPH ASIA*, 33(6), 2014.

[11] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. *IEEE International Conference on Computer Vision and Pattern Reognition*, 1997.

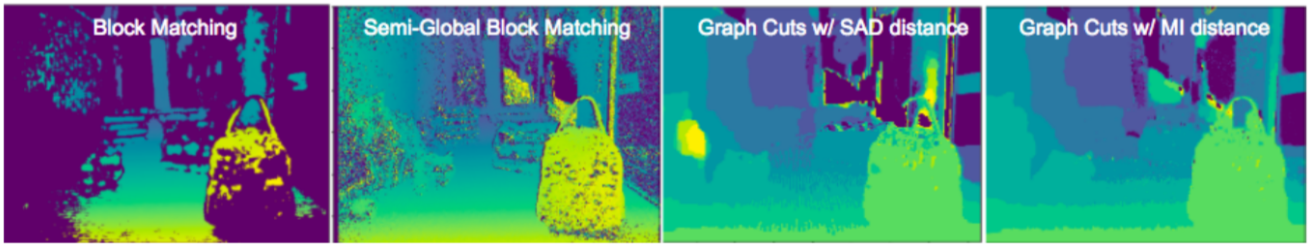

Figure 17. Disparity map estimation techniques explored within this project which include block matching, semi-global block matching, and graph cuts (with both a sum of absolute differences (SAD) distance and mutual information (MI) distance metric). Incorrect, high disparity values are labeled as zero in the graph cuts technique (seen in the purple abrupt areas in the background of the scene) to show the range of correct disparity estimates.

[12] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. *International Conference on Computer Vision and Pattern Recognition*, 2005.

[13] V. Kolmogorov, P. Monasse, and P. Tan. Kolmogorov and zabih's graph cuts stereo matching algorithm. *Image Processing On Line (IPOL)*, 4:220–251, 2014.

[14] K. Konolige. Small vision systems: Hardware and implementation. *Proceedings of the 8th International Symposium in Robotic Research*, pages 203–212, 1997.

[15] A. Mosleh, P. Green, E. Onzon, I. Begin, and J. Langlois. Camera intrinsic blur kernel estimation: A reliable framework. *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2015.

[16] J. Mustaniemi. Image fusion algorithm for a multi-aperture camera. Master's thesis, University of Oulu, Dept. of Computer Science and Engineering, Finland, 2015.

[17] H. Tang, X. Zhang, S. Zhuo, F. Chen, K. Kutulakos, and L. Shen. High resolution photography with an rgb-infrared camera. *IEEE International Conference on Computational Photography (ICCP)*, 4:1–10, 2015.

[18] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.