Remaining course logistics

- Optional homework assignment 7 is due on 12/18 at 23:59 ET.
  - Gradescope submission site available.

- Final project reports are due on Sunday 12/20 at 23:59 ET.
  - Gradescope submission site available.

- One upcoming computational photography talk:
  - Tuesday 12/22, 1 – 2 pm, Grace Kuo (lensless cameras, diffuserCam, AR/VR displays).

- Returning borrowed equipment:
https://docs.google.com/spreadsheets/d/1CVg7nUbI701pvZFPX3BR0uzKB76Y6PEl3tXmq1UF4AU/edit#gid=0

Class evaluation*s* – please take them!

- CMU's Faculty Course Evaluations (FCE): https://cmu.smartevals.com/
- TA evaluation: https://www.ugrad.cs.cmu.edu/ta/F20/feedback/
- 15-463/663/862 end-of-semester survey: https://docs.google.com/forms/d/e/1FAIpQLSew6XHzagr0KqWPFCmdoVmnHXQR5vfi4A6nEG0jMg6O1lBixA/viewform
- Please take all three of them, super helpful for developing future offerings of the class.
- Thanks in advance!

Using incident lightfield for simulating GelSight

Arpit Agarwal

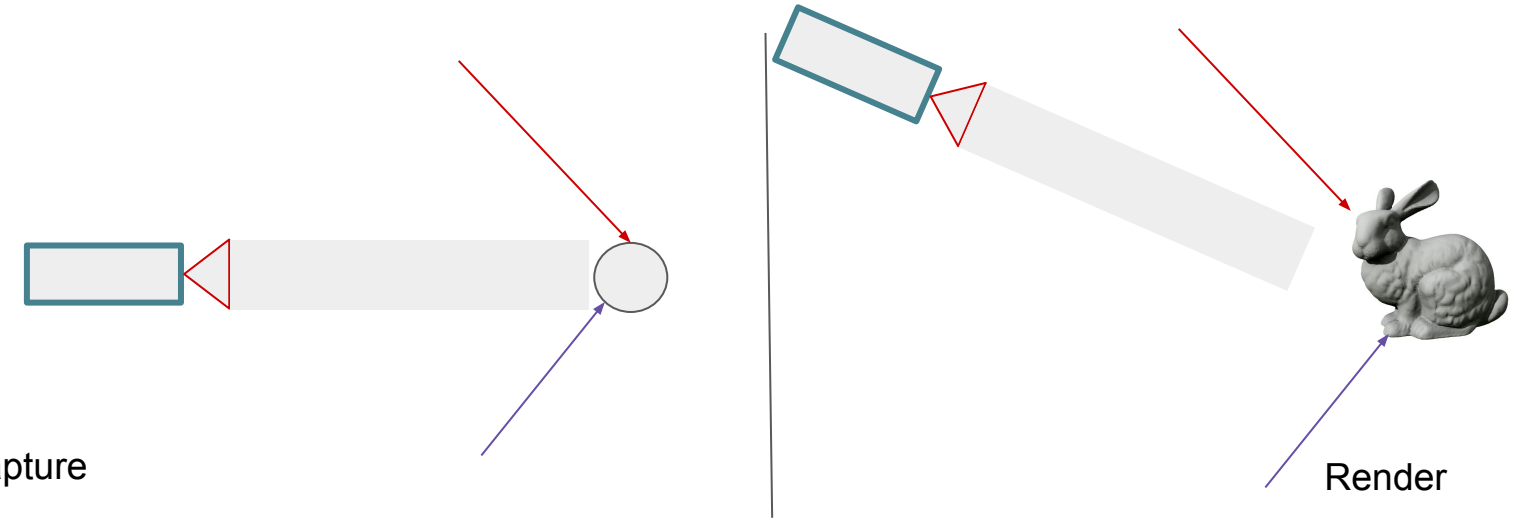
Objective

Capture spatially and directionally varying **illumination** and use this lighting to generate images with synthetic objects

Key Principle

1. Measured scene radiance and global illumination by placing light probe

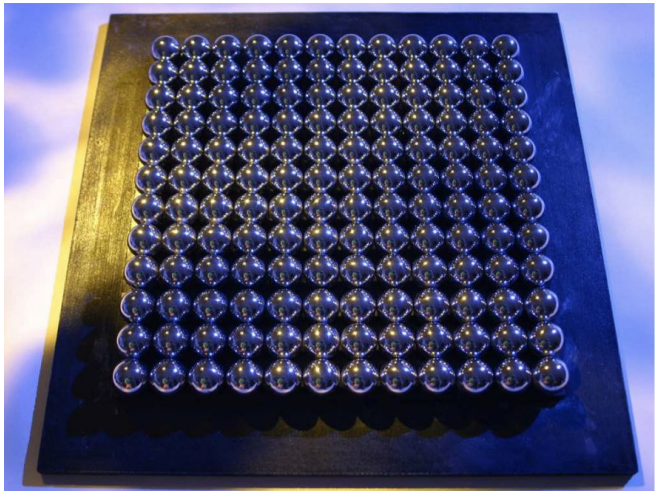
2. Use raycasting to find normals of surface at each pixel and interpolate colors



Debevec, Paul. "Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography." *ACM SIGGRAPH 2008 classes*. 2008. 1-10.

Spatially varying incident lightfield

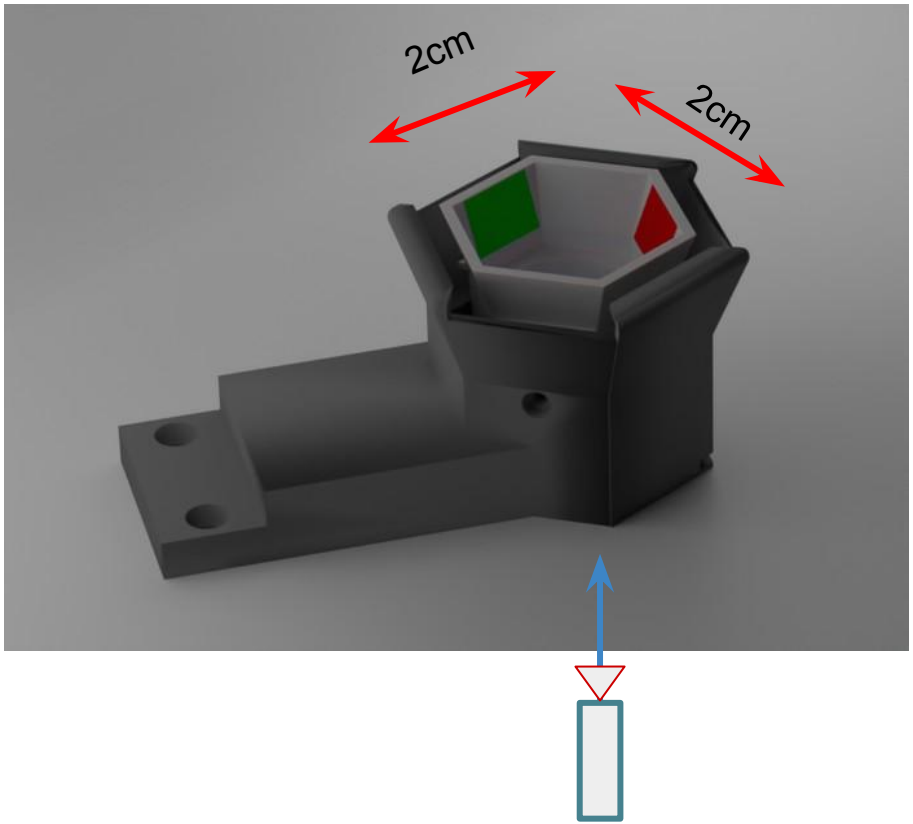
1. Previous method only works if the object is very small as compared to the captured image
2. Capture images with multiple probe locations



Unger, Jonas, et al. *Capturing and rendering with incident light fields*. UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY CA INST FOR CREATIVE TECHNOLOGIES, 2003.

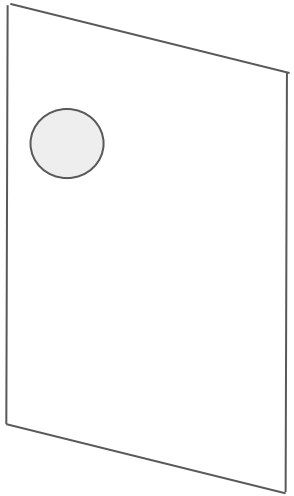
Motivation

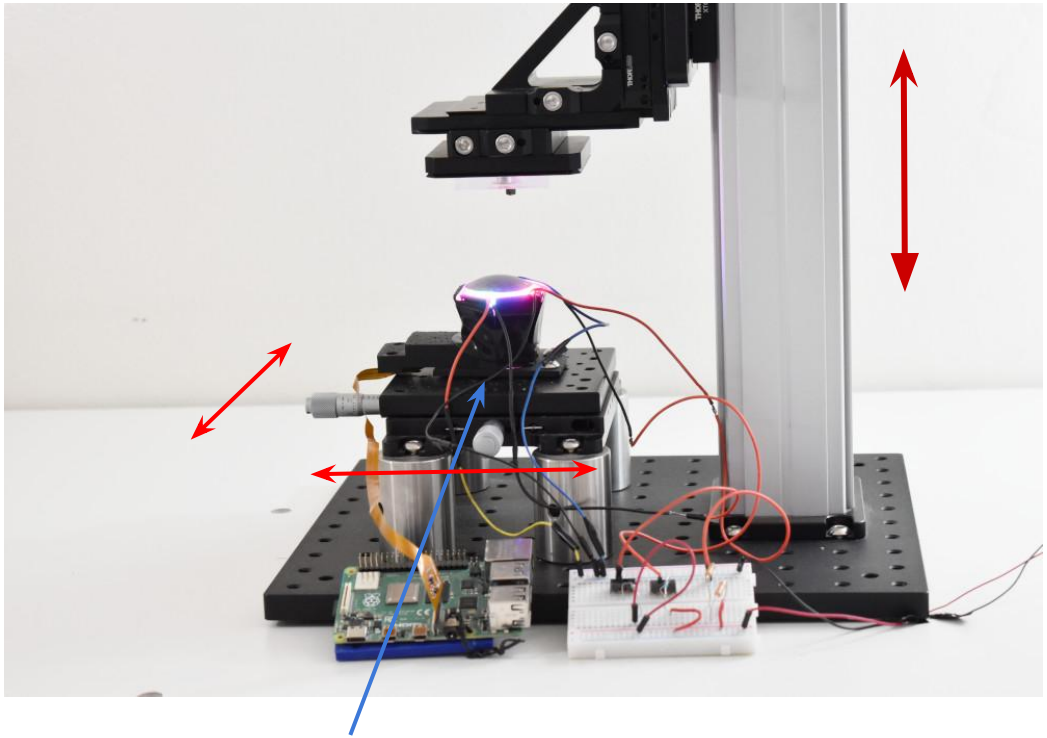
1. GelSight has small form factor
2. Camera can not be masked



Capture Method



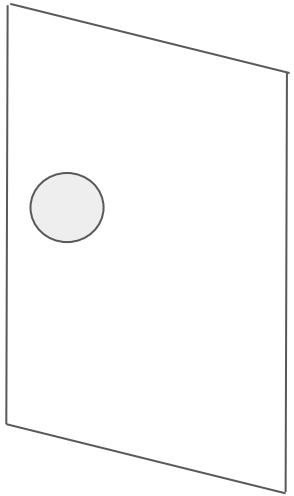


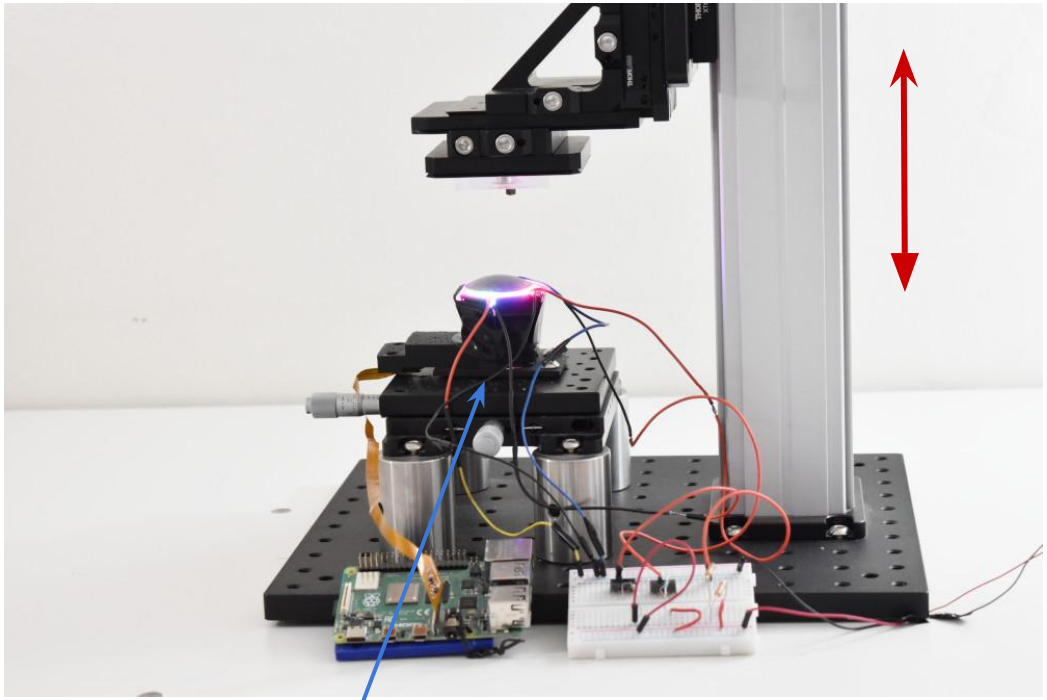


GelSight

Capture Method



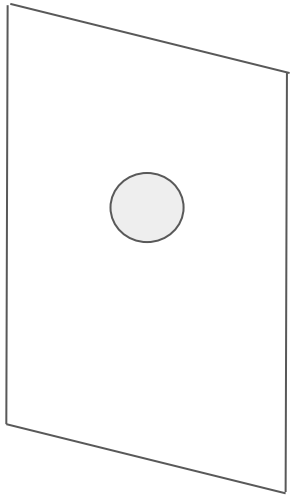


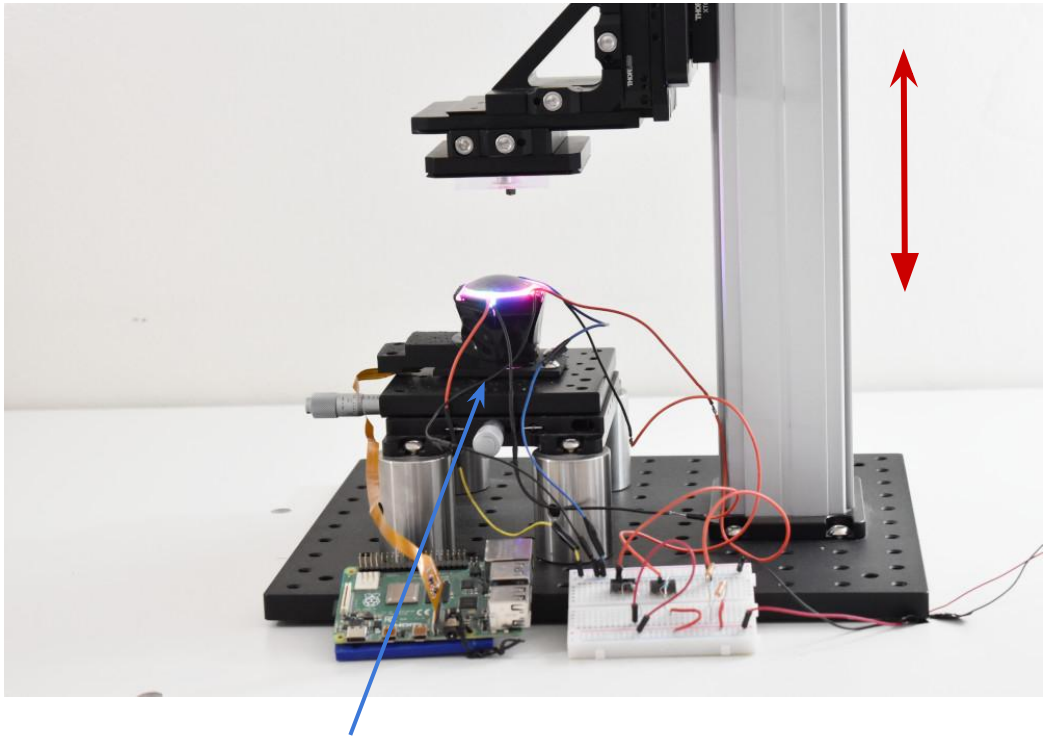


GelSight

Capture Method







Rendering Method

$$E(P,N) = \int_{\Omega} L'(P,\omega)\langle P,N\rangle d\omega$$

Diffuse material

$$E(P,N) = \langle N, \int_{\Omega} L'(P,\omega) N d\omega \rangle$$

$$E(P,N) = \langle N, C_P \rangle$$

Precomputed

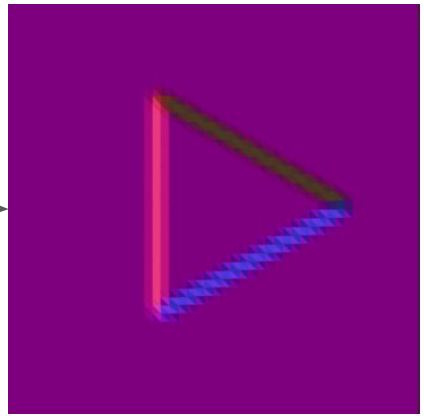
Rendering Method

$$E(P,N) = \int_{\Omega} L'(P,\omega)\langle P,N\rangle d\omega$$

$$E(P,N) = \langle N, \int_{\Omega} L'(P,\omega) N d\omega \rangle$$

$$E(P,N) = \langle N, C_P \rangle$$

Diffuse material

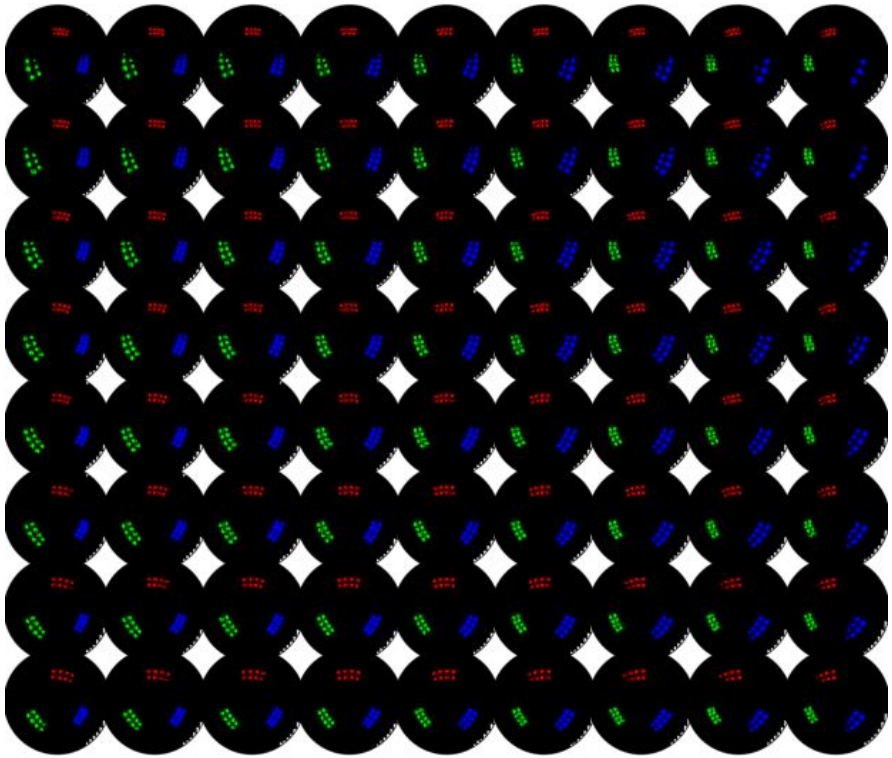


Scaled to [0,1]

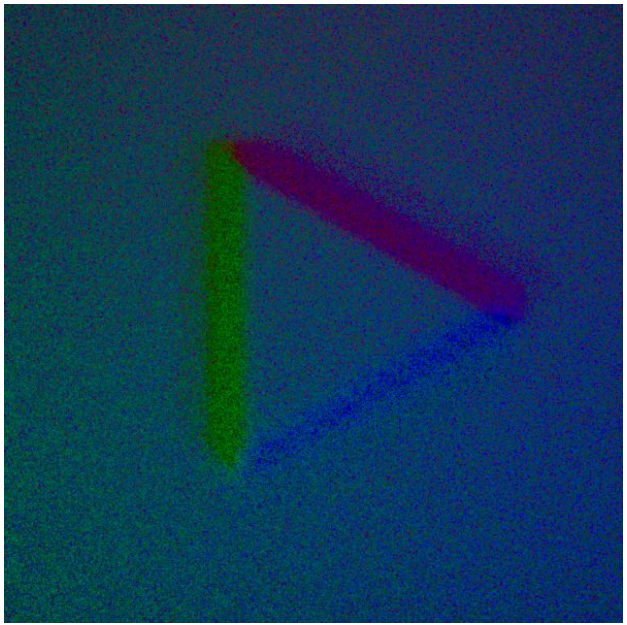
Simulation setup

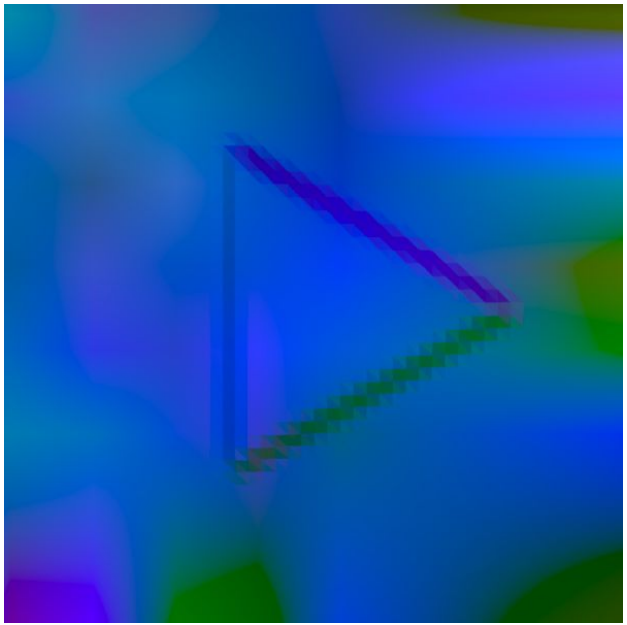
1. Setup scene in Mitsuba 2
2. Probe location along grid 8x9 with

   0.5mm spacing



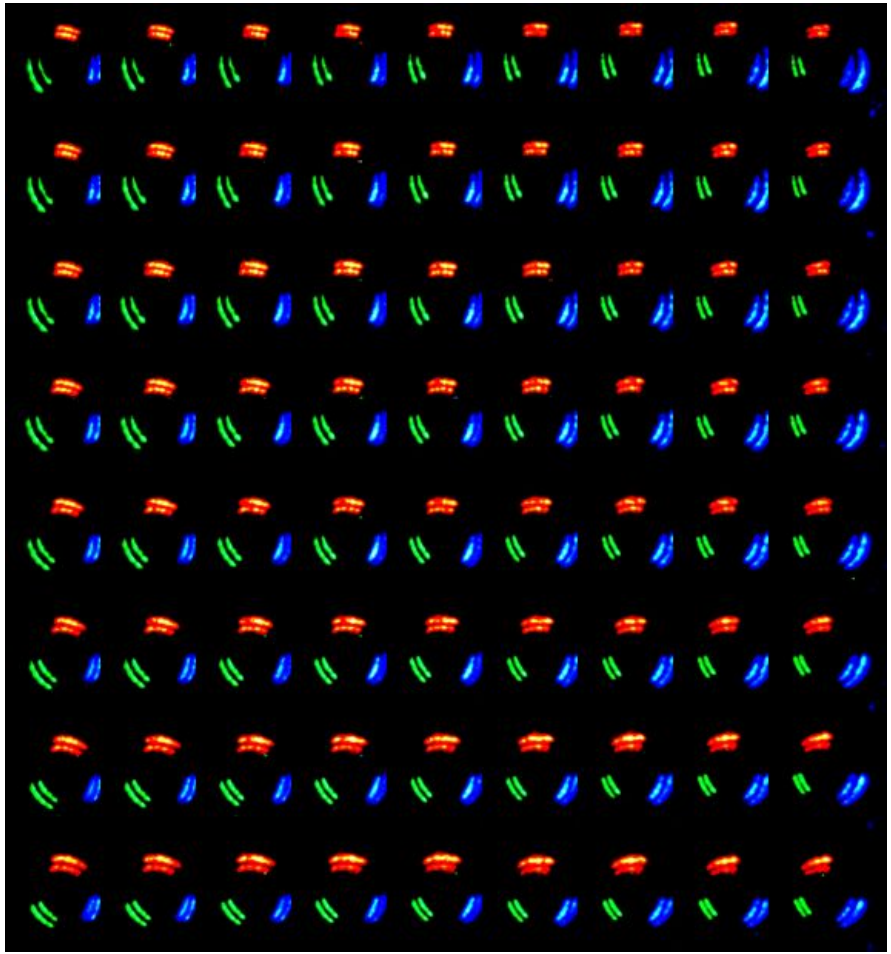
Simulation Results



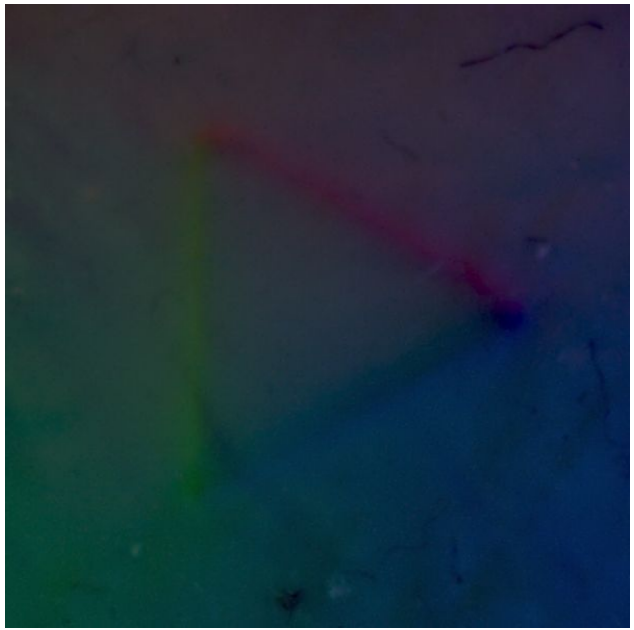


GelSight data collection

1. Collected low dynamic range raw images at 5 exposure settings
2. Merged LDR images into HDR using MergeDebevec function



GelSight Results





Summary

1. Increasing the number of probe locations might result to better image
2. Implement the method for general material
3. The method could be used to place fake light sources for global illumination

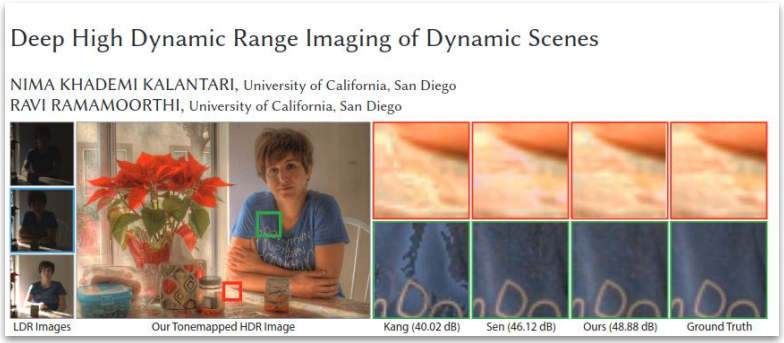
Carnegie Mellon University

Deep High Dynamic Ranging of Dynamic Scenes

Presenter: Uma Arunachalam
Andrew ID: uarunach

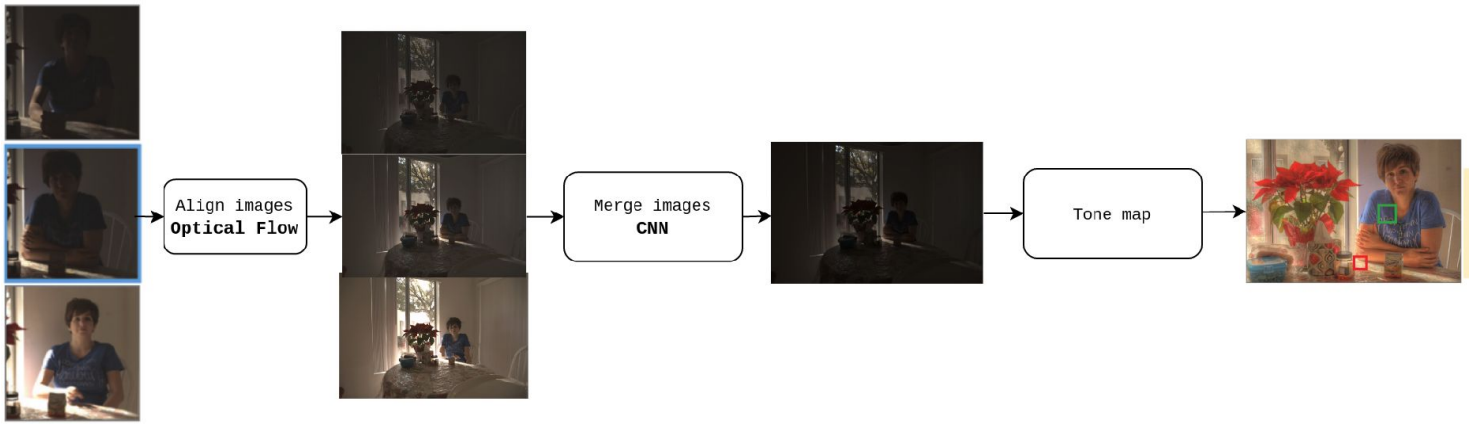
Brief Overview

- To merge LDR images that are not necessarily aligned or based on scenes that are not necessarily static, using deep learning (DL) and produce a HDR image as output
- Motivation-
  - Reduce the setup-requirements on capture process
    - *Can be applied to any random scene captured with a handheld camera instead of using tripods, gphoto for capture*
  - Reduce the processing time required for merging
    - *Time is spent only on aligning images and inference time of the DL network.*
- Mimic and extend on Baseline reference [1]



Carnegie Mellon University

Approach



3 misaligned images

Carnegie Mellon University

Optical Flow



3 misaligned images

Carnegie Mellon University

CeLiu's Classical Method

- Explores brightness constancy assumption, and uses iterative re-weighted least squares to minimize pixel intensity values between reference image and the 'estimated' flow guided pixel of another image



Z2, reference image



Z3

Adjust Exposure

$$Z_{2,3} = \text{clip}(Z_2 \Delta_{2,3}^{1/\gamma})$$

$$\Delta_{2,3} = t_3 / t_2$$

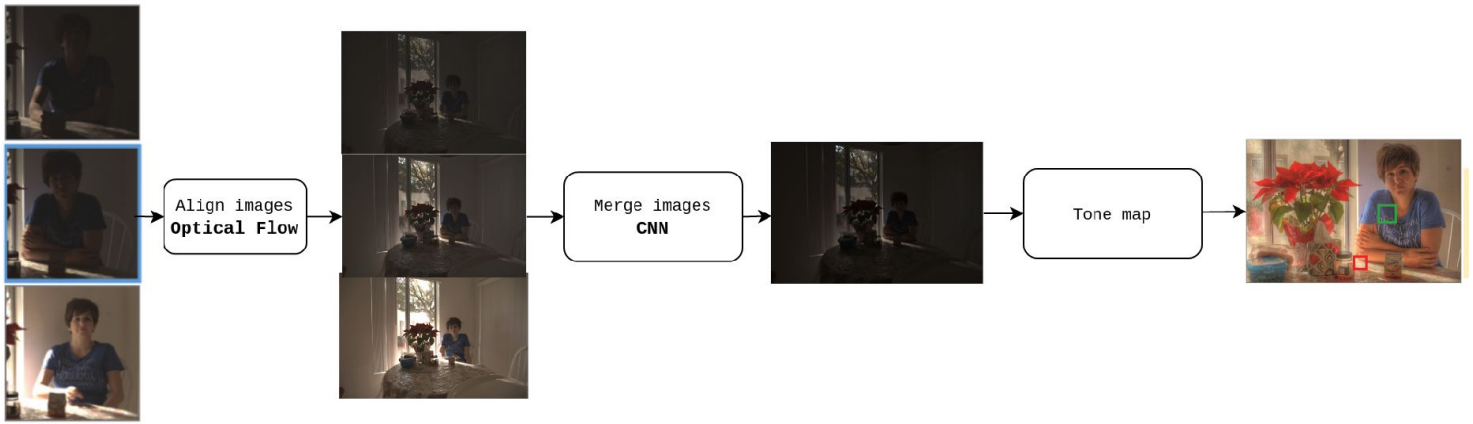
CeLiu Optical Flow + Restore exposure





negie
on
versity

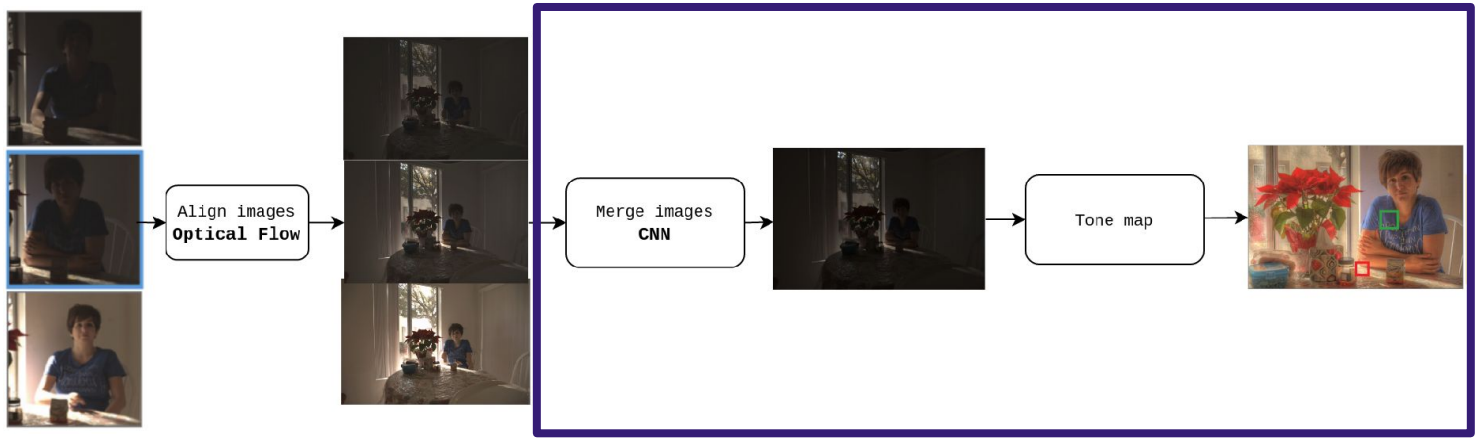
Approach



3 misaligned images

Carnegie Mellon University

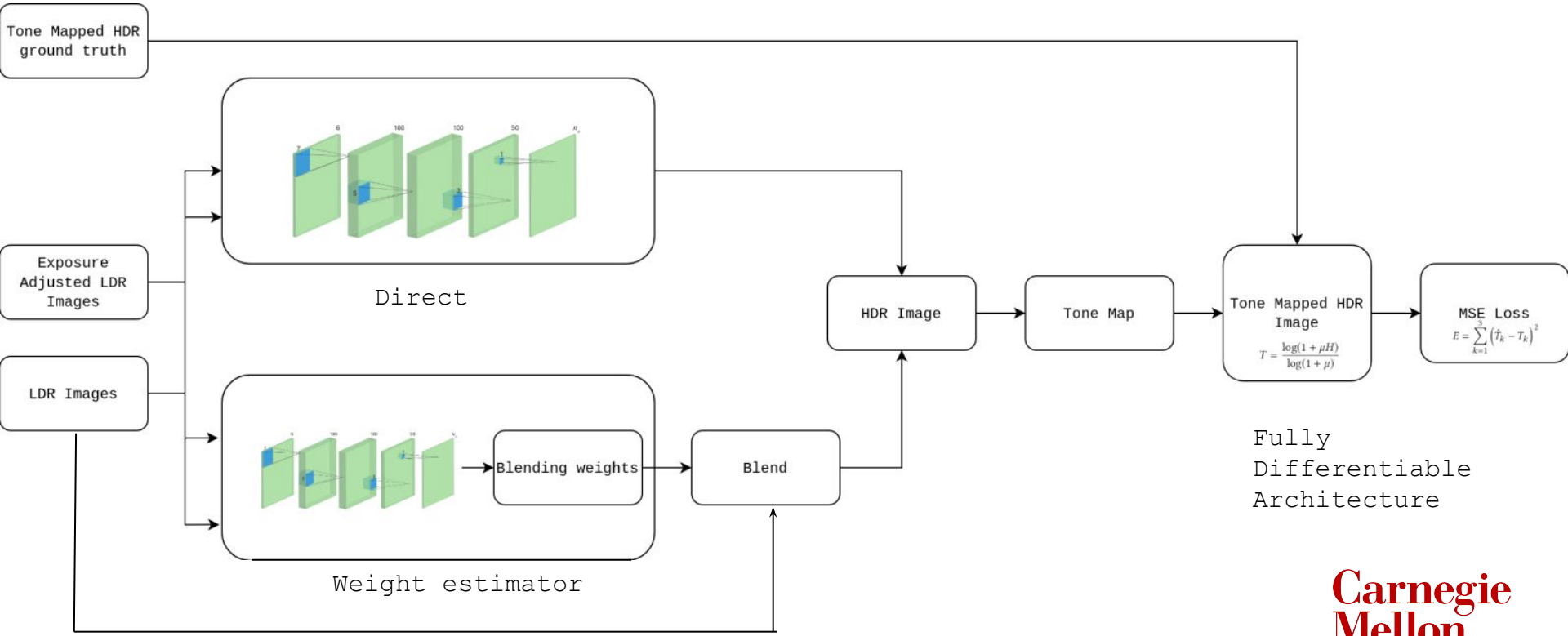
Learning based LDR merging



3 misaligned images

Carnegie Mellon University

Network Architecture

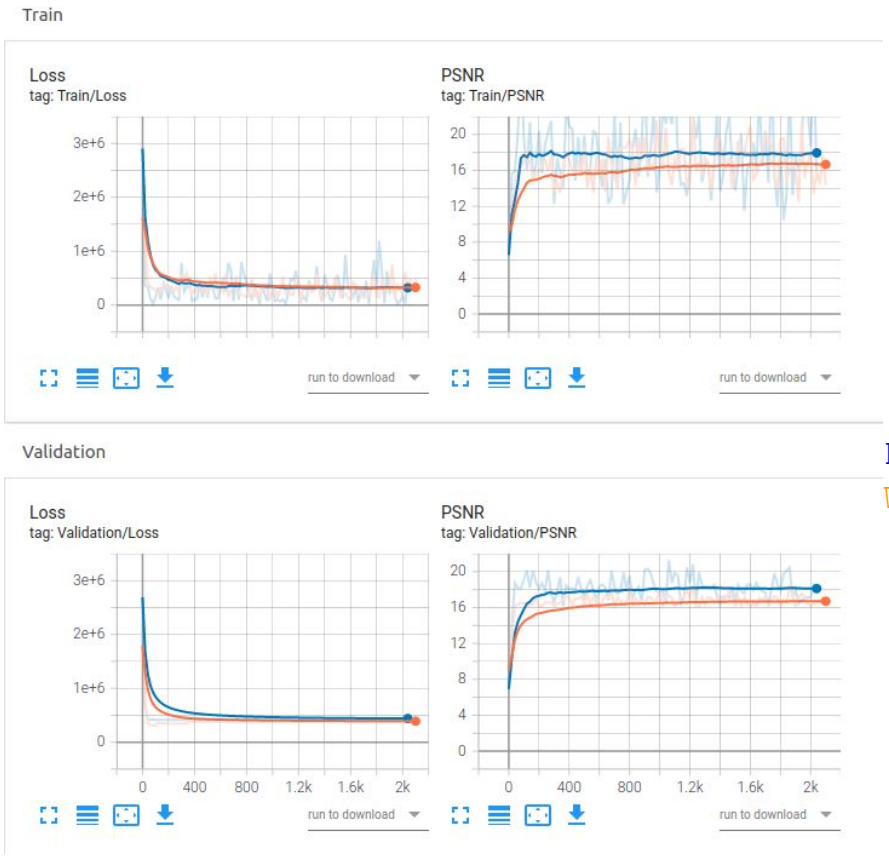


Carnegie Mellon University

Experimental Setup

- Default optical flow parameters, as suggested in [2]
- For training Direct and WE architectures:
  - Used online available datasets plus few custom captures
  - Xavier uniform initialization
  - Adam optimizer, Learning rate = 0.0001
  - Trained images over ~100 epochs, logging every 20 iterations
- Trained for about 3 hours before achieving convergence per method on NVIDIA Tesla v100
- Torch MSE Loss, with *reduction to 'sum'*

Results: Error metric



Direct

WE

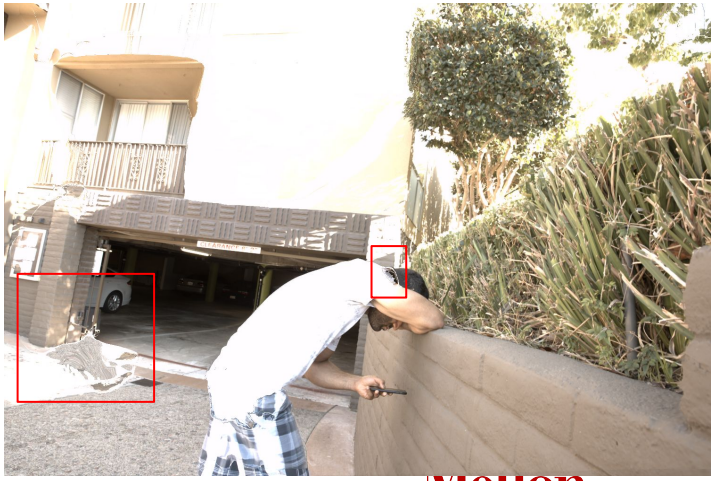
Carnegie Mellon University

Results: Optical Flow









Results: Deep merging vs Tent









Carnegie Mellon University

Results: WE vs Direct





Carnegie Mellon University

Results

Metric	Tent merging	Deep merging
Mean Square error	0.0594	0.0037
PSNR	12.27	24.31
Run time (merging)*	9.58 s	5.88 s

* not profiled, averaged over 2 runs

Carnegie Mellon University

Main References

[1] https://cseweb.ucsd.edu/~viscomp/projects/SIG17HDR/PaperData/SIGGRAPH17_HDR.pdf

[2] https://people.csail.mit.edu/celiu/OpticalFlow/

Thank you!
Questions?

Carnegie Mellon University

Real-time Cartoonization

Ricky Bao

What is NPR and cartoonization?

- Non-photorealistic rendering favors artistic styles instead of faithful recreation of a scene
  - Paintings and drawings
- Cartoonization is a form of NPR
  - Enhanced lines
  - Textures and colors smoothed out
- Use cases of cartoonization
  - Entertainment
  - Simplification of scenes

Example of cartoonization





How is cartoonization performed?

- Line extraction
  - Canny
  - Gradient
  - Difference of Gaussian
- Color reduction
  - Bilateral filter
  - Anisotropic diffusion filter
- Image enhancement
  - Combine line extraction and color reduction

Real-time cartoonization

	Average (ms)	Min (ms)	Max (ms)
Roof	56.2	50.3	62.7
Android	54.9	47.2	60.1
Pool	58.0	50.6	66.9
Globe	53.2	46.3	58.2
Sunset	54.7	49.8	61.0

Resources

- R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk, “Non-photorealistic camera: Depth edge detection and stylized rendering using multi-flash imaging,” eng, *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 679-688, 2004, issn: 0730-0301.
- A. B. Patankar, P. A. Kubde, and A. Karia, “Image cartoonization methods,” eng, IEEE, 2016, pp. 1-7, isbn: 9781509032914.

Multi-Flash Imaging for Depth Edge Detection and Photometric Stereo

Matthew Baron (mcbaron)

Setup





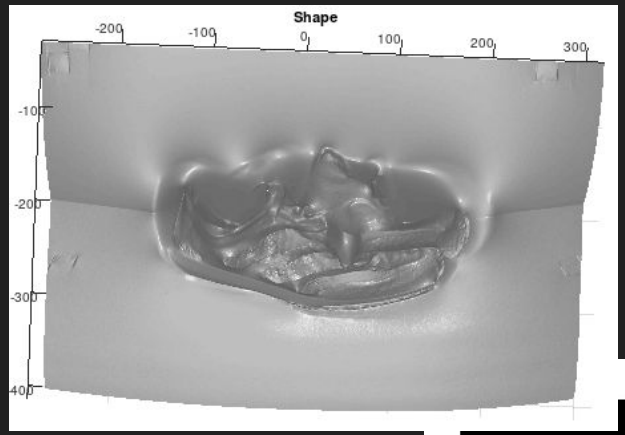
Depth Edge Detection

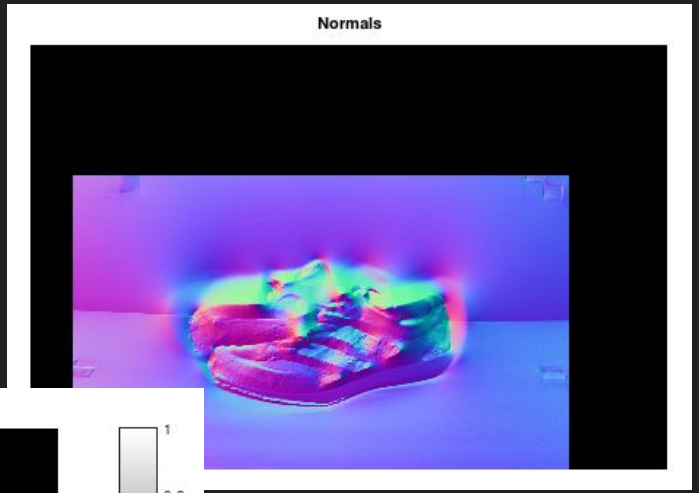


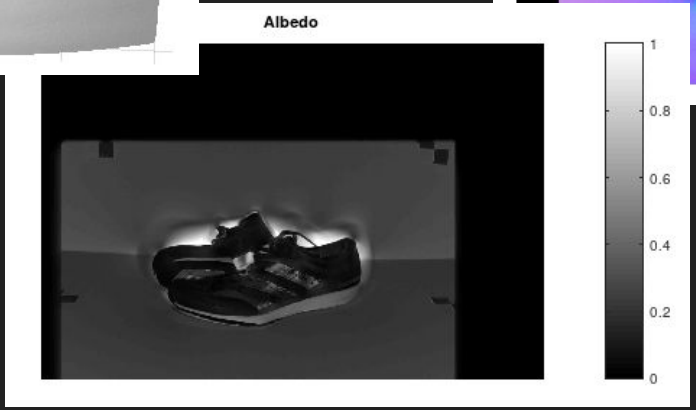




Photometric Stereo







References

[1] T. Papadhimitri and P. Favaro, “A New Perspective on Uncalibrated Photometric Stereo,” in 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, Jun. 2013, pp. 1474–1481, doi: 10.1109/CVPR.2013.194.

[2] R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk, “Non-Photorealistic Camera: Depth Edge Detection and Stylized Rendering Using Multi-Flash Imaging,” p. 12.

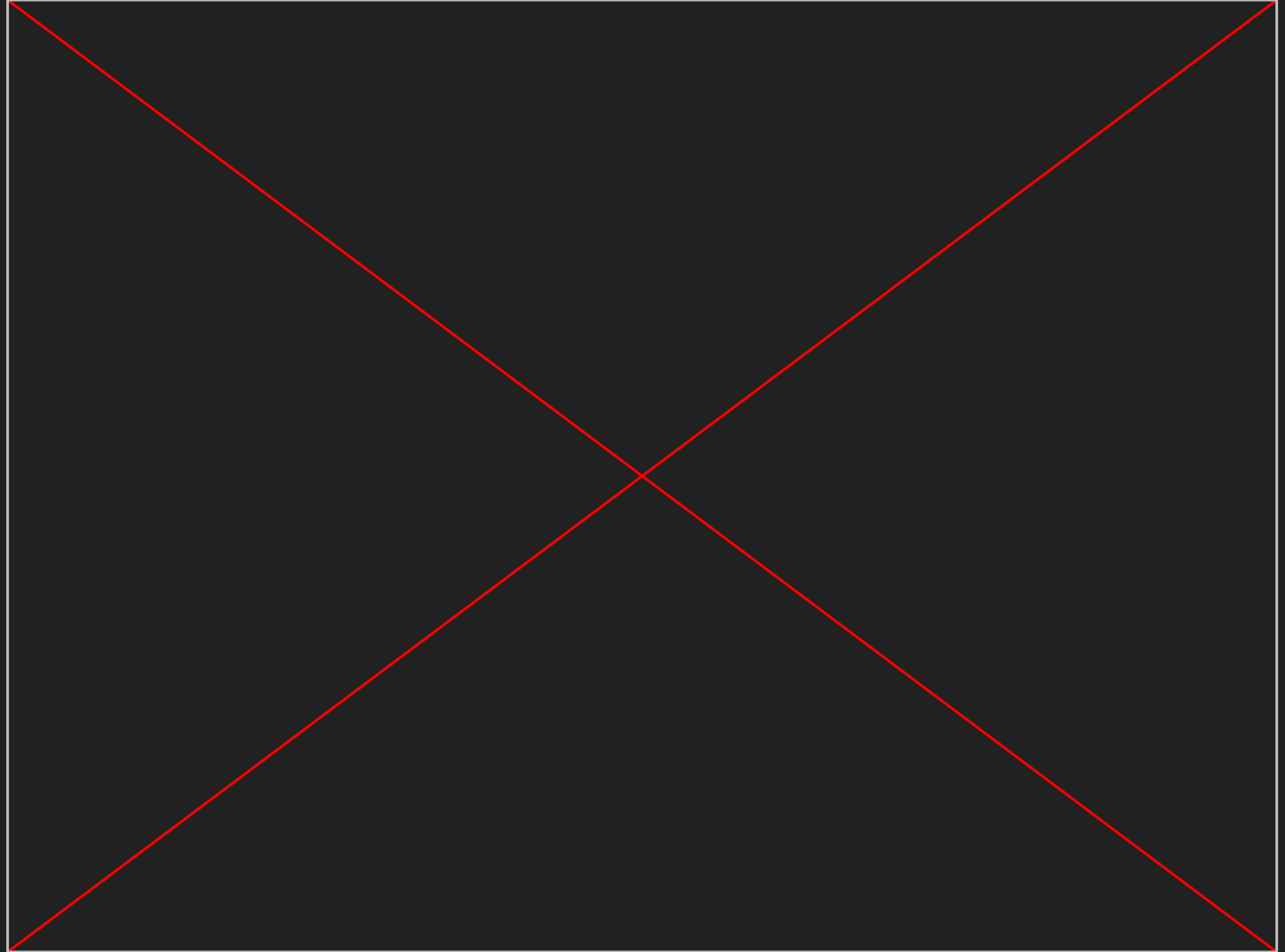
[3] Y. Taguchi, “Rainbow Flash Camera: Depth Edge Extraction Using Complementary Colors,” p. 16.

[4] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma, “Robust Photometric Stereo via Low-Rank Matrix Completion and Recovery,” in Computer Vision – ACCV 2010, vol. 6494, R. Kimmel, R. Klette, and A. Sugimoto, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 703–717.

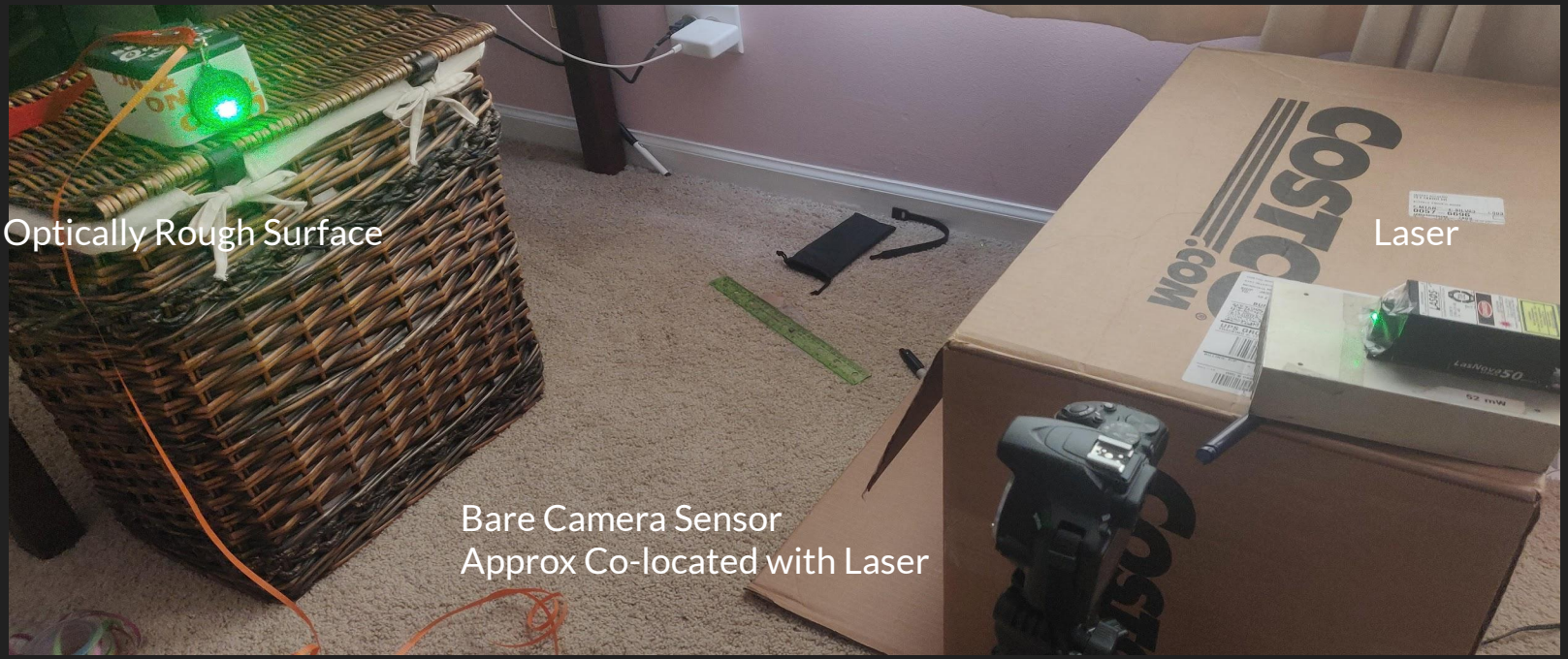
[5] Y. Quéau, F. Lauze, and J.-D. Durou, “Solving Uncalibrated Photometric Stereo Using Total Variation,” J Math Imaging Vis, vol. 52, no. 1, pp. 87–107, May 2015, doi: 10.1007/s10851-014-0512-5.

Tracking Microscopic Motion Using Speckle Imaging

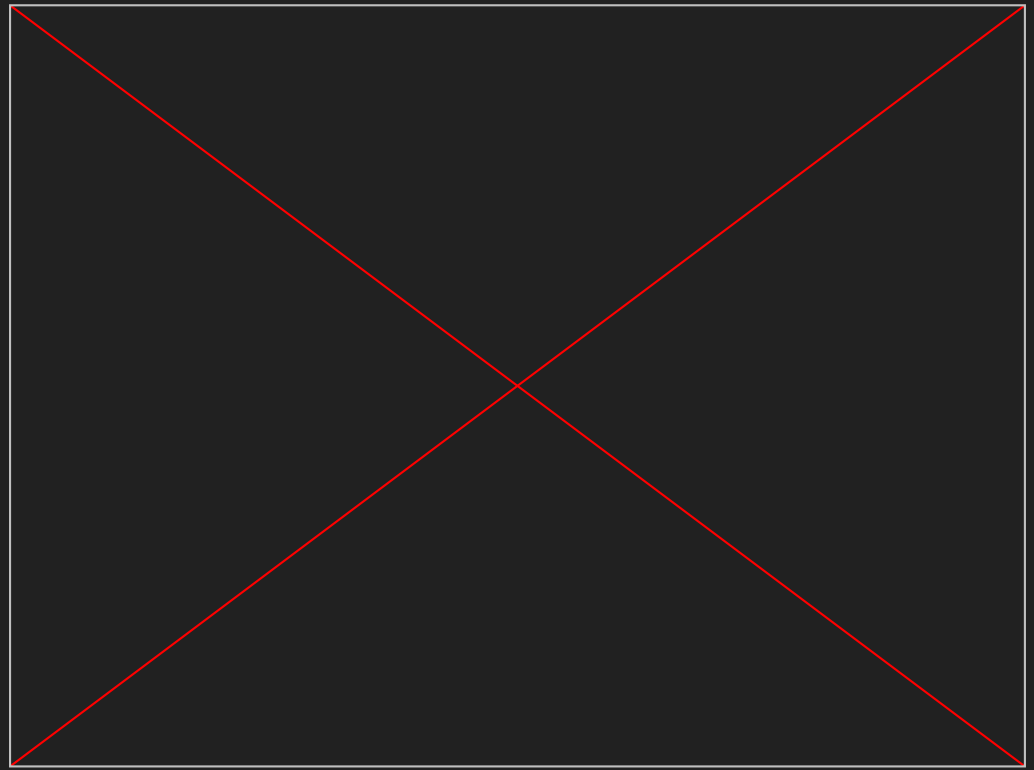
Yash Belhe



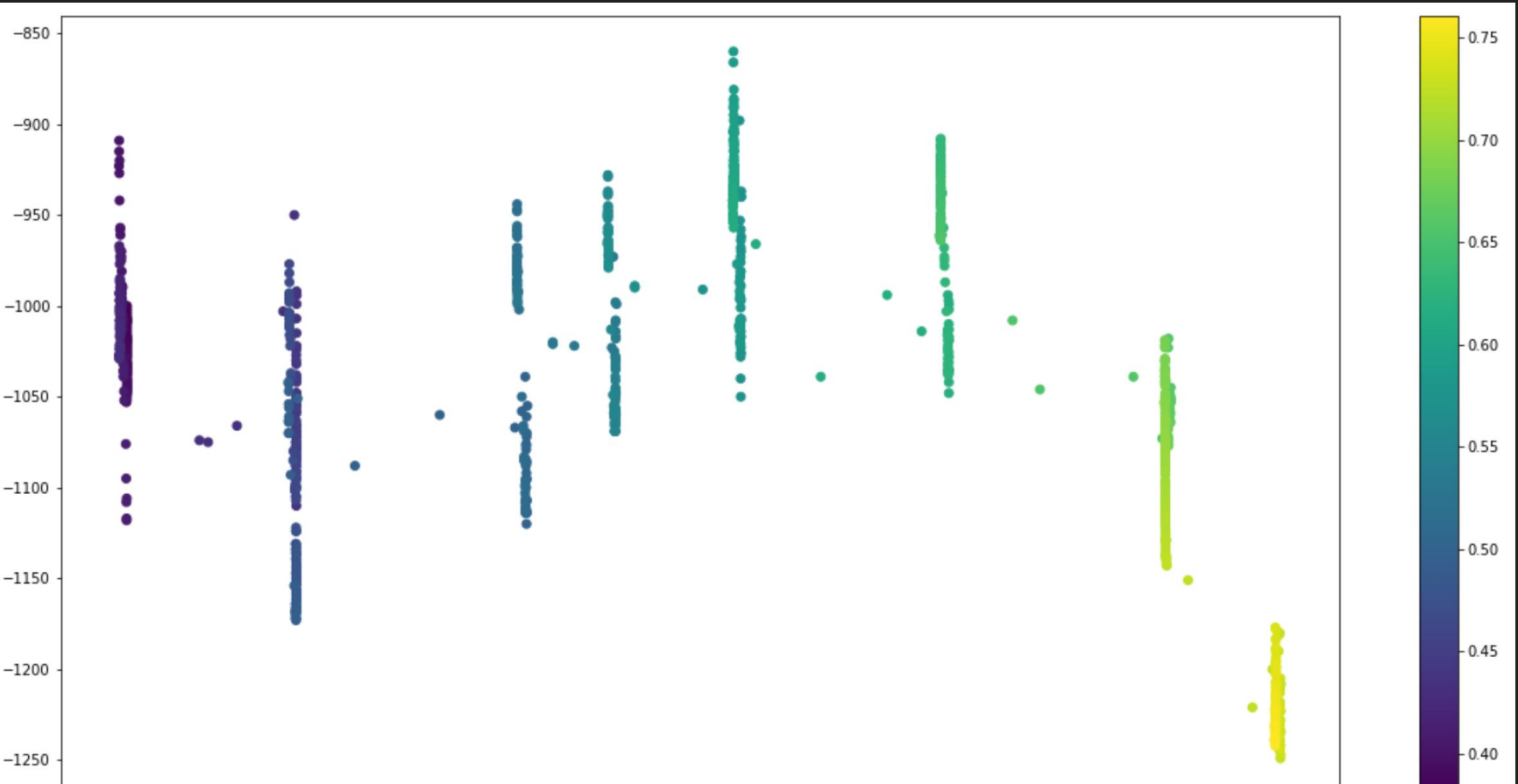
Setup


Optically Rough Surface
Laser
Bare Camera Sensor
Approx Co-located with Laser

Example with horizontal motion (with stutters)



Tracking Results



Carnegie Mellon University

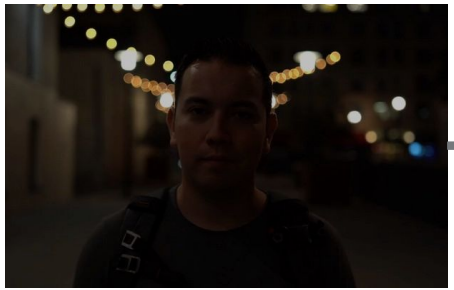
Learning to predict Depth and Autofocus using Focal and Aperture stack

Akankshya Kar and Anand Bhoraskar
{akankshk,abhorask}@andrew.cmu.edu

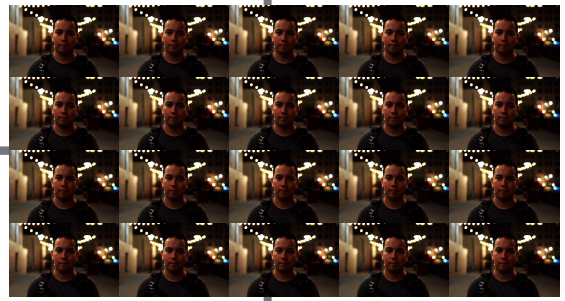
Contents

- Problem Statement
- Related Work
- Dataset
- Proposed Method
- Ground Truth generation for Autofocus
- Depth Map Generation
- Autofocus
- Future Work

Problem Statement



Low Light Image



AFI Image



Depth Map

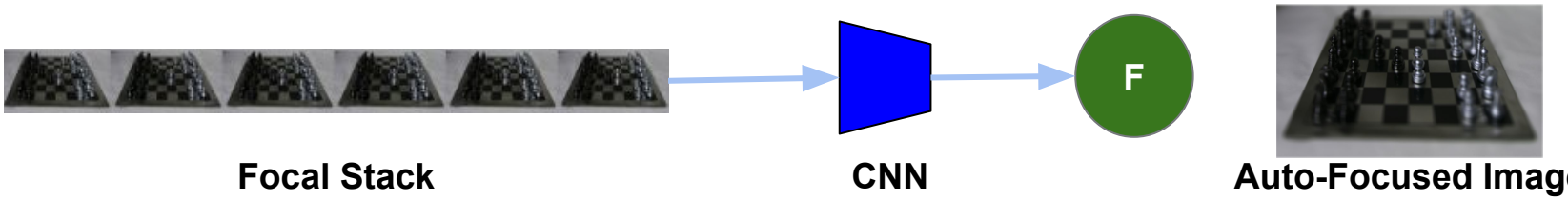


Bright Autofocus Image

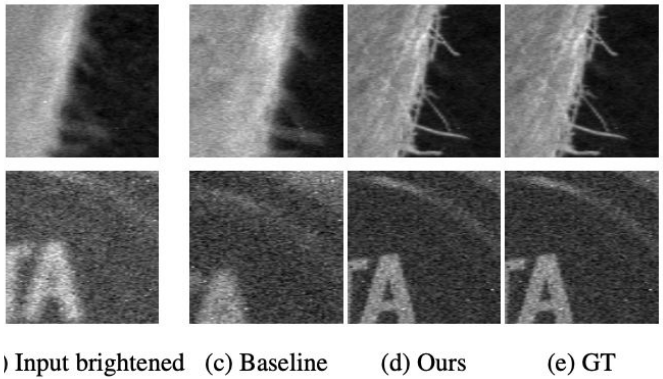
Carnegie Mellon University

Related Work

Learning to Autofocus[1]



- Uses Soft ordinal regression[4] for predicting F from 49 indices in focal stacks
- Captures its own images using 5 Pixel phones in High resolution
- Use MobileNetV2, to make the prediction very fast, to enable on-device use case

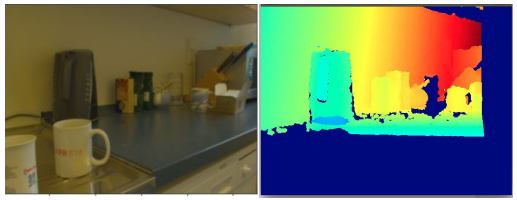


Carnegie Mellon University

Datasets

DDFF[7]

- 720 images (12 scenes)
- Resolution 383 × 552
- Missing area in depth
- Very low baseline
- Lenselet size = 9
- Available online



Flowers Dataset[8]

- 3343 images
- Resolution 376 x 541
- No depth data
- Limited Domain(1)
- Synthetic data
- Lenslet size = 14



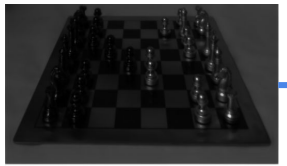
Learning to Autofocus dataset[1]

- 387,000 patches of 128x128
- Resolution 1512 × 2016
- 50 Varied Scenes (indoor and outdoor)
- Not released yet



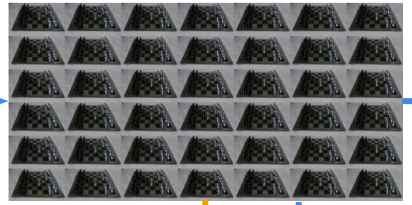
Carnegie Mellon University

Proposed Method



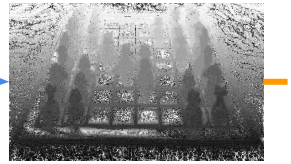
All in focus

CNN
(Dark to light)



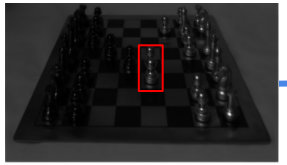
AFI

CNN



Depth

GT generation



Selected Patch

CNN

F, A

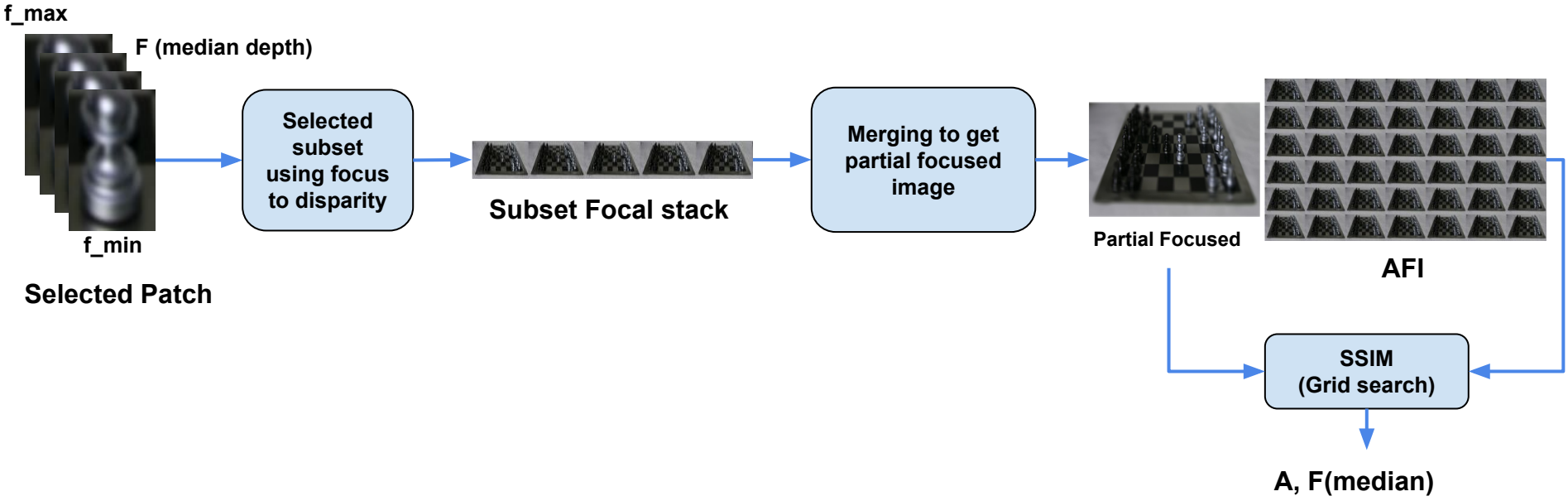


Refocusing

Project scope

Carnegie Mellon University

Ground Truth generation for Autofocus



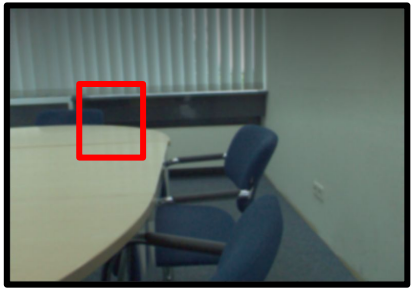
Carnegie Mellon University

Results

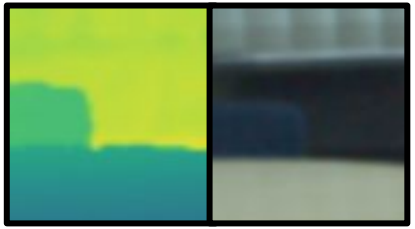
GT generation

Path	X_begin	Y_begin	Aperture size	Focus Distance
cafeteria/LF_0001.npy	32	96	7	3
cafeteria/LF_0001.npy	32	128	7	3
cafeteria/LF_0001.npy	32	160	7	3
cafeteria/LF_0001.npy	32	192	7	3
cafeteria/LF_0001.npy	64	32	7	3
cafeteria/LF_0001.npy	64	64	7	3
cafeteria/LF_0001.npy	64	96	7	3
cafeteria/LF_0001.npy	64	128	7	3
cafeteria/LF_0001.npy	64	160	7	2
cafeteria/LF_0001.npy	64	192	7	2
cafeteria/LF_0001.npy	96	32	7	2
cafeteria/LF_0001.npy	96	64	7	2
cafeteria/LF_0001.npy	96	96	7	2
cafeteria/LF_0001.npy	96	128	7	2

of Patches: 19,311
Patches Size : 64 x 64
Stride : 100



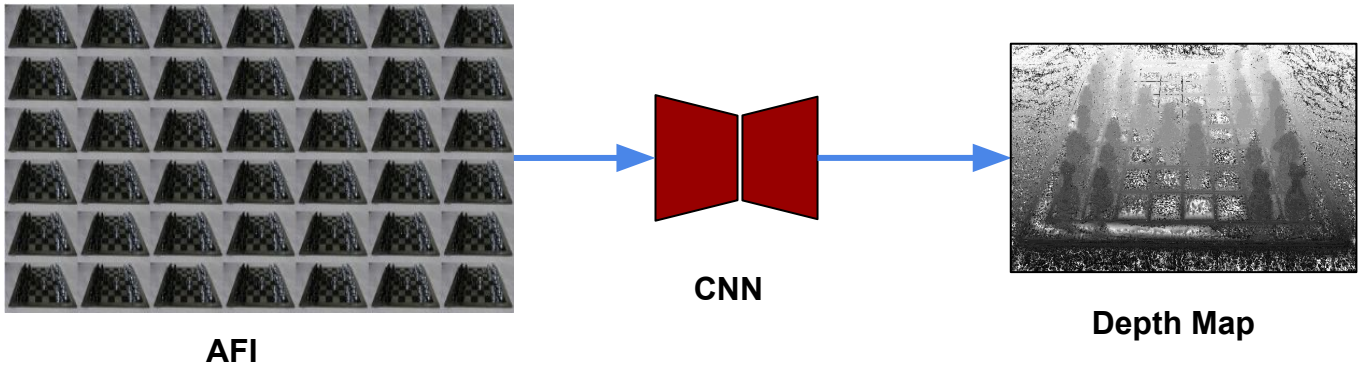
Selected Patch in Full image



Depth Patch and cropped out patch

Carnegie Mellon University

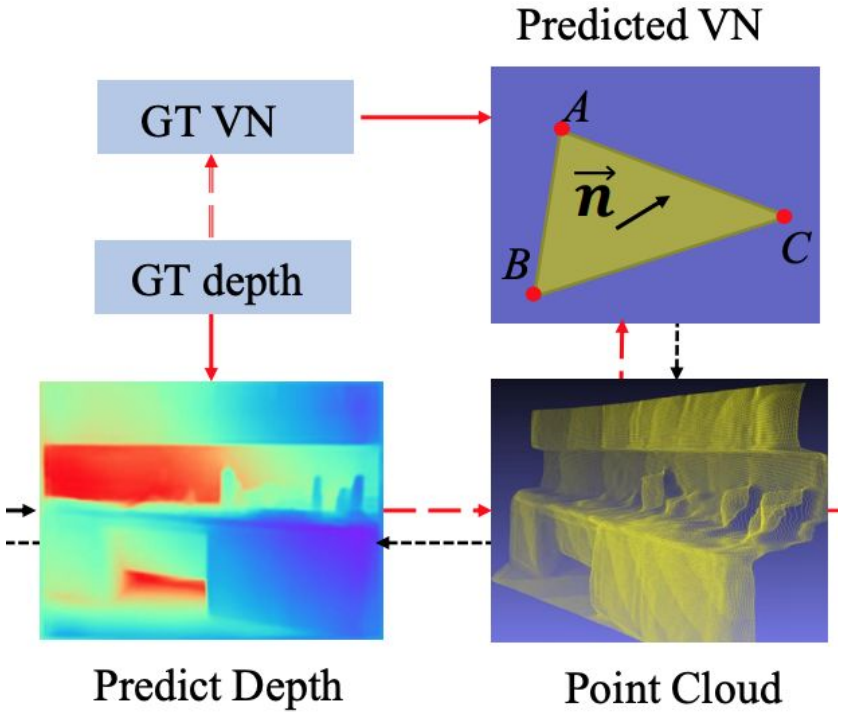
Depth Map Generation



- AFI is generated and given to Encoder-Decoder style architecture, which has 80 channels (10 focal and 8 aperture)
- The Encoder is MobileNetV2 and Decoder is FCN with Upsampling, the loss is Virtual Normal Loss

Carnegie Mellon University

Virtual Normal Loss[2]

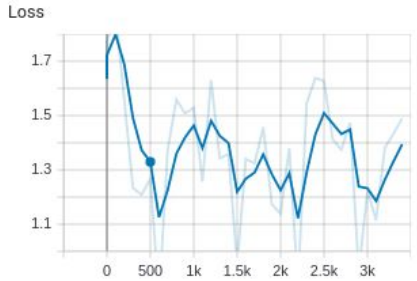


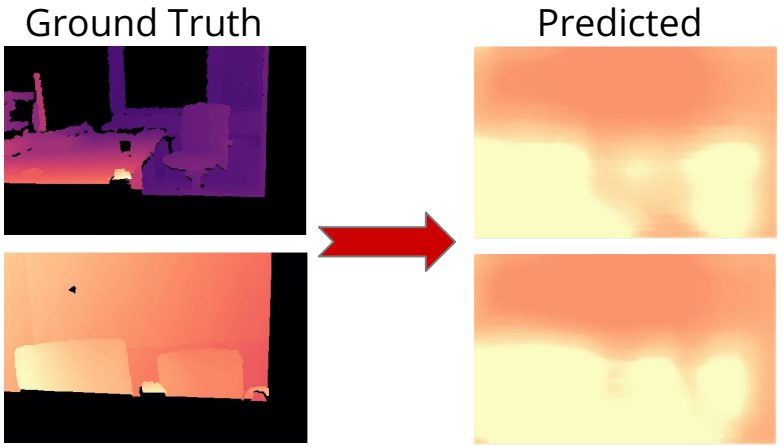
- Enforces a high order geometric constraint in the 3D space for the depth prediction task.
- Makes N groups of 3 points in Point cloud and fits a plane
- DDFF is an indoor dataset, hence VNL was chosen

Carnegie Mellon University

Results

Depth from confocal stack

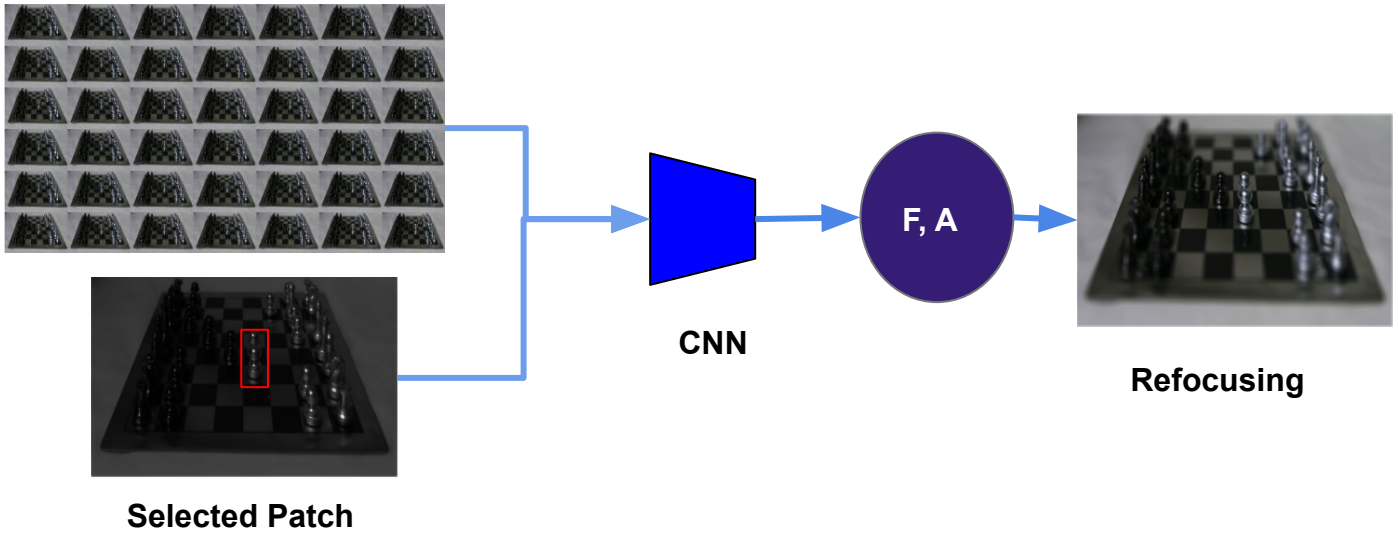




- Current encoder-decoder network with MobileNet-V2 backbone and VNL loss
- So far we trained ~4000 iterations
- Reasons for slow training
  - Small dataset
  - Limited compute resources
  - Large size of confocal stack (8 times the light field image)
  - Pre-trained model not used to initialise

Carnegie Mellon University

Auto-Focus

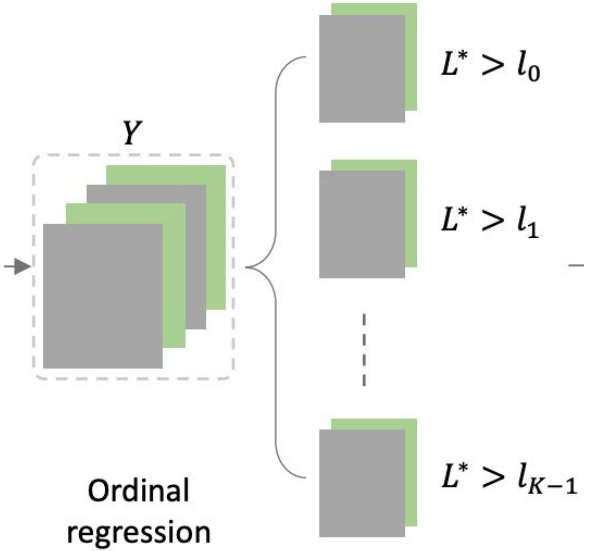


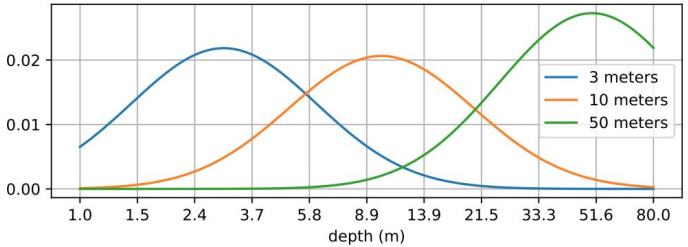
- AFI is generated and selected patch is given to given to CNN
- The selected patch is given as masked and as extra 2 channels in CoordConv[5] method, loss is using Soft Ordinal Loss[4]

Carnegie Mellon University

Ordinal Regression Loss[3,4]

- Ordinal Regression solves tasks with relative ordering by penalising more for further away predictions
- Soft Ordinal Regression makes the ground truth vector have a Softmax like weighting scheme rather than hard one hot vector





Carnegie Mellon University

Future Work

- DDFF dataset has many limitations, we would shift to Flowers dataset as it is bigger and more blurring is there due to bigger aperture size
- Flowers dataset does not have depth, but we can generate the same Aperture and Focus dataset using Focus measure to get the same mapping between focus and disparity
- Work on removing the bug, which gives maximum aperture size (could be a DDFF related problem)
- Complete the training and ablation experiments
- Work on Low light aspect of it by using RetinaNet to generate low light images from light field and use that for training

References

1. Herrmann, Charles, et al. "Learning to Autofocus." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.
2. Yin, Wei, et al. "Enforcing geometric constraints of virtual normal for depth prediction." Proceedings of the IEEE International Conference on Computer Vision. 2019.
3. Diaz, Raul, and Amit Marathe. "Soft labels for ordinal regression." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019.
4. Fu, Huan, et al. "Deep ordinal regression network for monocular depth estimation." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
5. Liu, Rosanne, et al. "An intriguing failing of convolutional neural networks and the coordconv solution." Advances in neural information processing systems 31 (2018): 9605-9616.
6. Tsai, Yu-Ju, et al. "Attention-based view selection networks for light-field disparity estimation." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 07. 2020.
7. Hazirbas, Caner, et al. "Deep depth from focus." Asian Conference on Computer Vision. Springer, Cham, 2018.
8. Srinivasan, Pratul P., et al. "Learning to synthesize a 4D RGBD light field from a single image." Proceedings of the IEEE International Conference on Computer Vision. 2017.

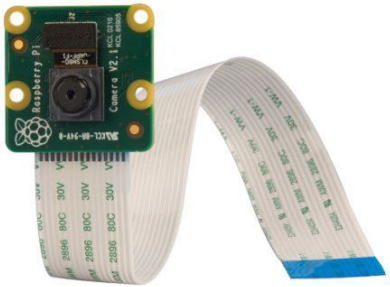
Carnegie Mellon University

Fast Separation of Direct/Global Images In A Pocket

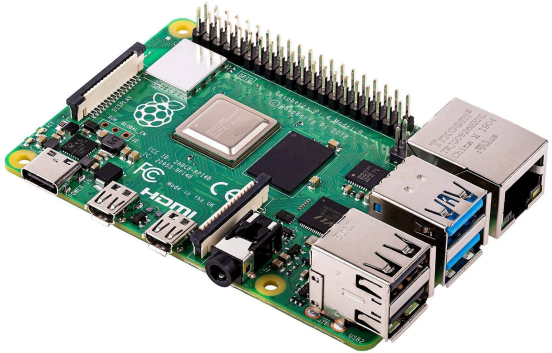
Presenter: Zili Chai (zilic)



Hardware Setup



Sony IMX219



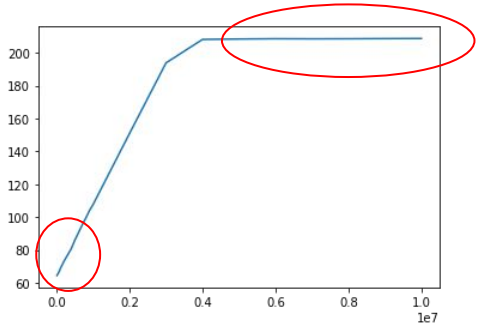
RPi 4B



DLPDLCR2000EVM

Camera Calibration

daylight result

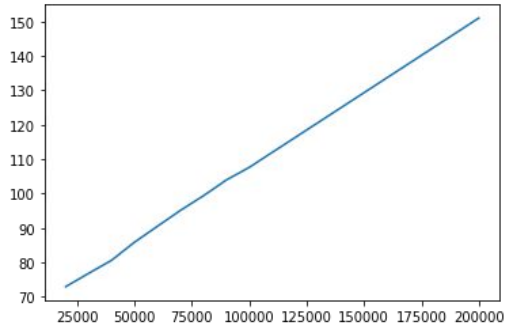


10 bit bayer data, BGGR ordering

AAAAAAAA BBBBBBBB CCCCCCCC DDDDDDDD AABBCCDD

Linearity ∝ exposure

ISO = 100





1ms



100ms

Dark Frame

- Mean = 63.90 (on 100 dark images)

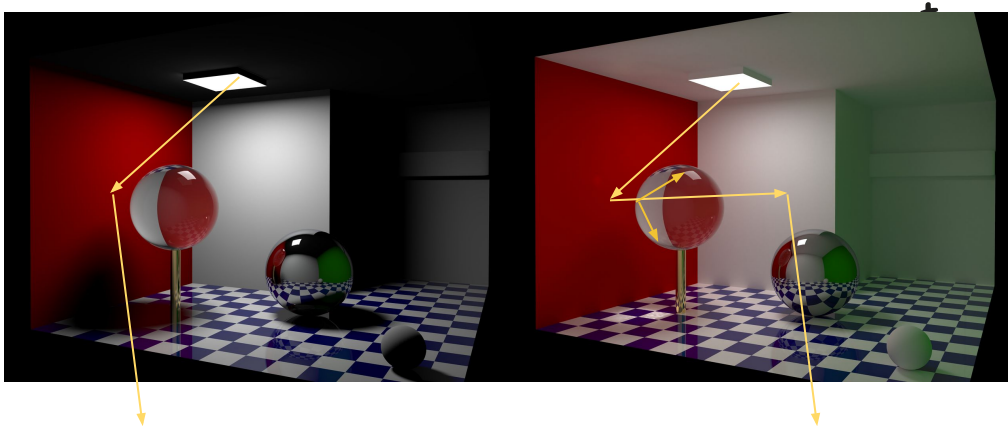
Projector also needs time!

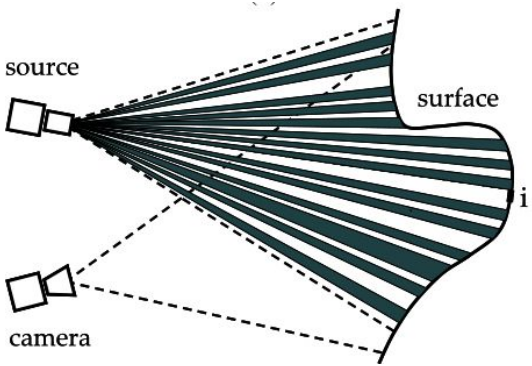
Global/Direct Images

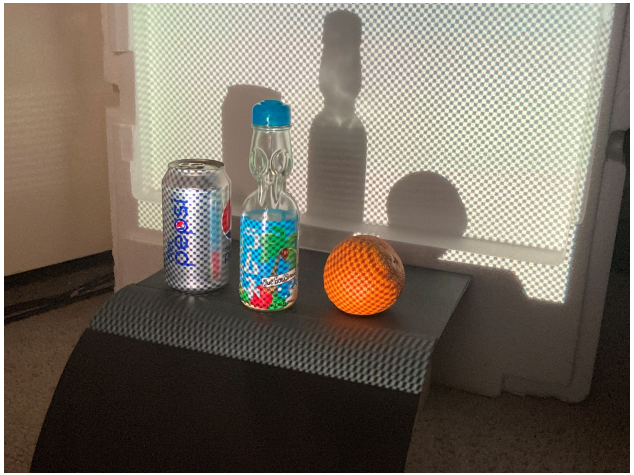
Global Illumination

scattering, interreflection, shadows…

Separation In Real World





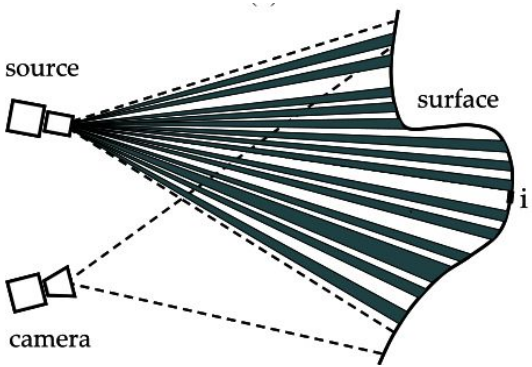


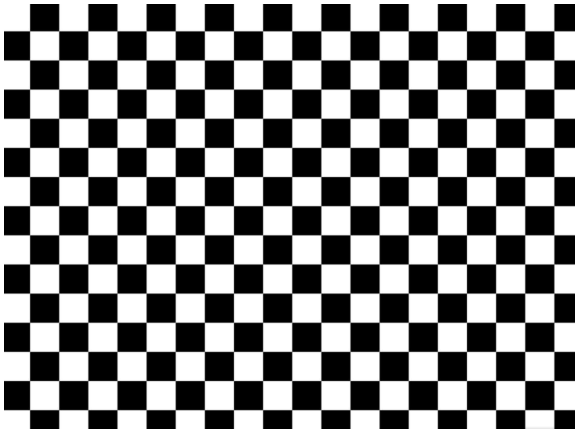
Global/Direct Images

Separation in Real World

$$
\begin{aligned}
L^{+}[c,i] &= L_d[c,i] + \alpha L_g[c,i] + b(1-\alpha)L_g[c,i]\,, \\
L^{-}[c,i] &= bL_d[c,i] + (1-\alpha)L_g[c,i] + \alpha bL_g[c,i]\,.
\end{aligned}
$$

$$
\begin{aligned}
L^{+}[c,i] &= L_d[c,i] + (1+b)\frac{L_g[c,i]}{2}\,, \\
L^{-}[c,i] &= bL_d[c,i] + (1+b)\frac{L_g[c,i]}{2}\,.
\end{aligned}
$$





Results

Checkerboard shift



max



direct



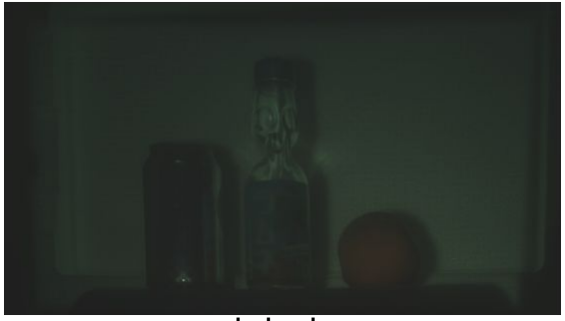
direct



min



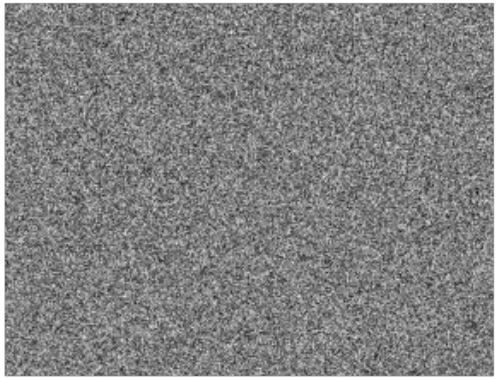
global



global

Results

More Patterns



random pattern with phase shift

0.5+0.5·sinφ

+2pi/7, 4pi/7, 6pi/7, 8pi/7, 10pi/7,12pi/7



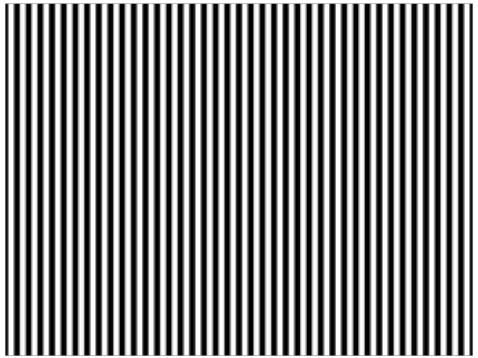
direct



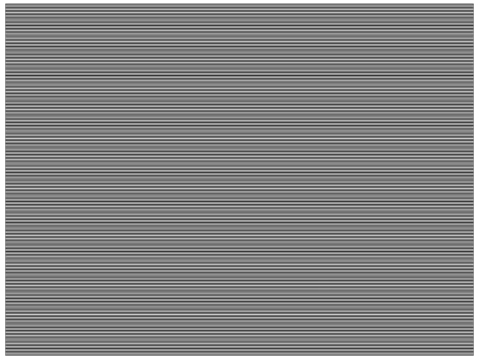
global

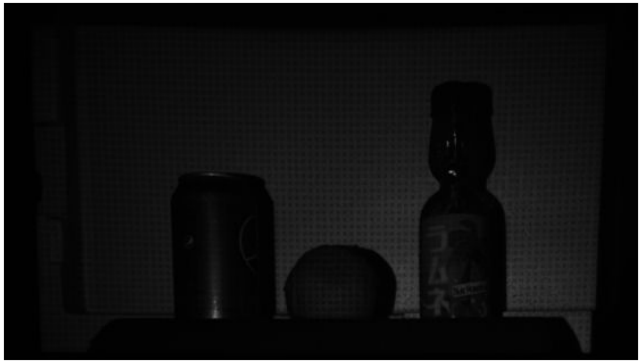
Results

More Patterns



gray code







References

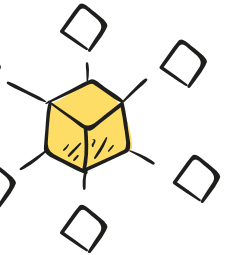
Sony, IMX219 product brief version 1.0, (Jan 2017).

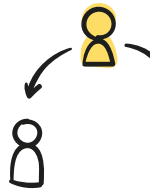
Texas Instruments, TI DLP® LightCrafter™ Display 2000 EVM User's Guide, (Oct 2017).

Pagnutti, Mary A., et al. "Laying the foundation to use Raspberry Pi 3 V2 camera module imagery for scientific and engineering purposes." *Journal of Electronic Imaging* 26.1 (2017): 013014.

Nayar, Shree K., et al. "Fast separation of direct and global components of a scene using high frequency illumination." *ACM SIGGRAPH 2006 Papers*. 2006. 935-944.







Q&A



Kirchhoff Migration for NLOS Imaging

Dorian Chan

NLOS Imaging

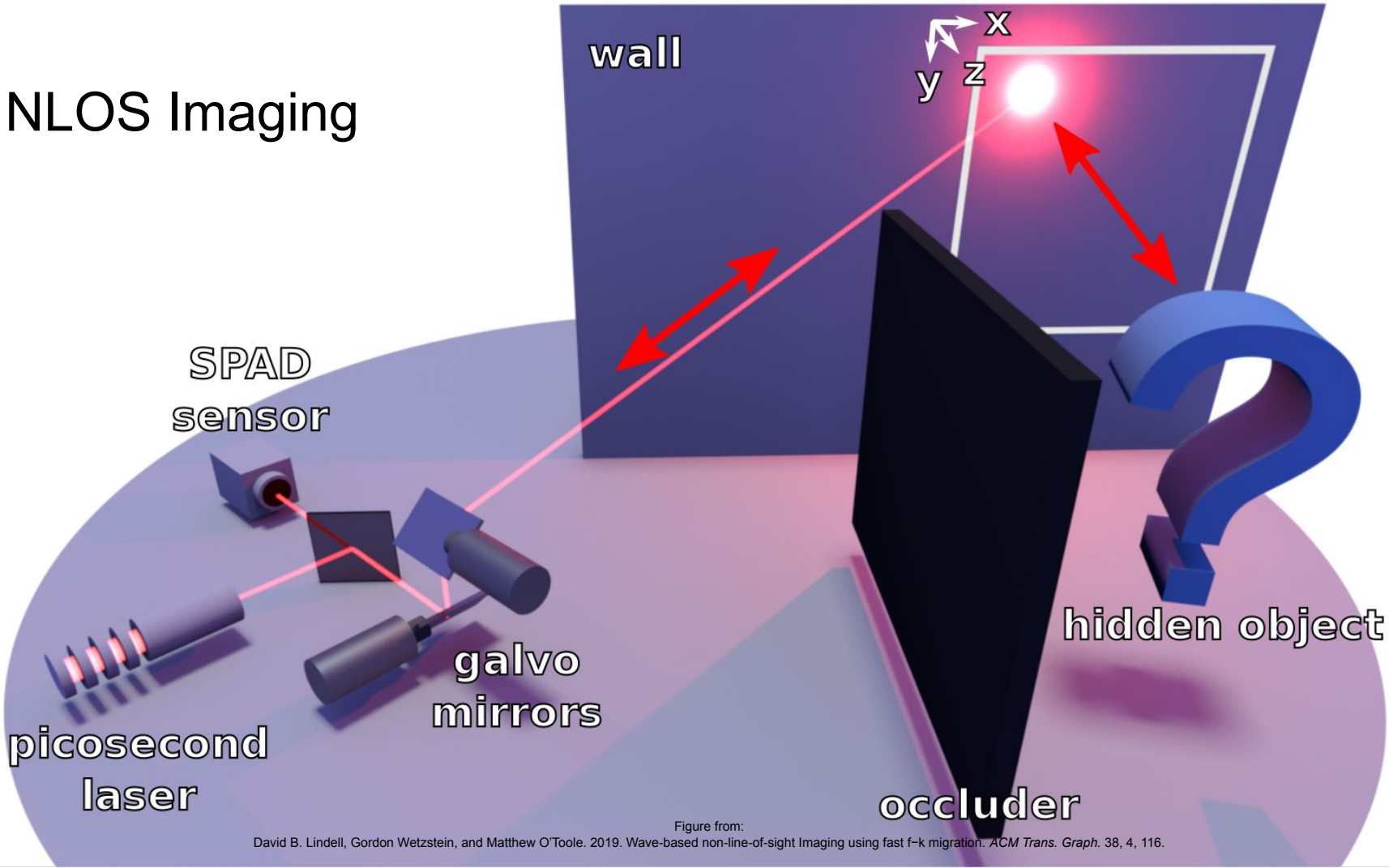
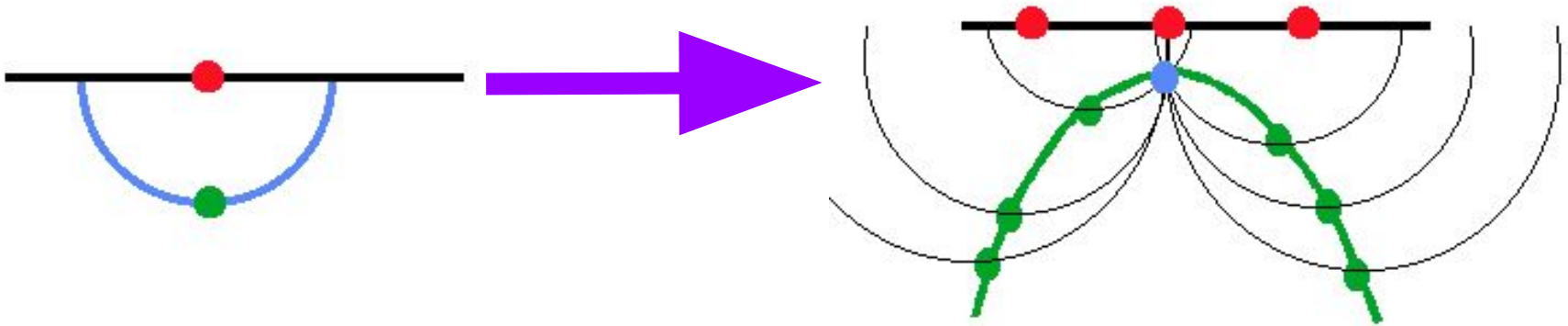


Figure from:
David B. Lindell, Gordon Wetzstein, and Matthew O'Toole. 2019. Wave-based non-line-of-sight Imaging using fast f−k migration. *ACM Trans. Graph.* 38, 4, 116.

Traditional Approach: Backprojection



From http://www.xsgeo.com/course/mig.htm

F-K Migration: a Wave-Based Theory



$$\Phi(k_x, k_y, k_z) = \frac{ck_z}{\sqrt{k_x^2 + k_y^2 + k_z^2}} \bar{\Phi}(k_x, k_y, c\sqrt{k_x^2 + k_y^2 + k_z^2})$$

Kirchhoff Migration: another idea from seismology

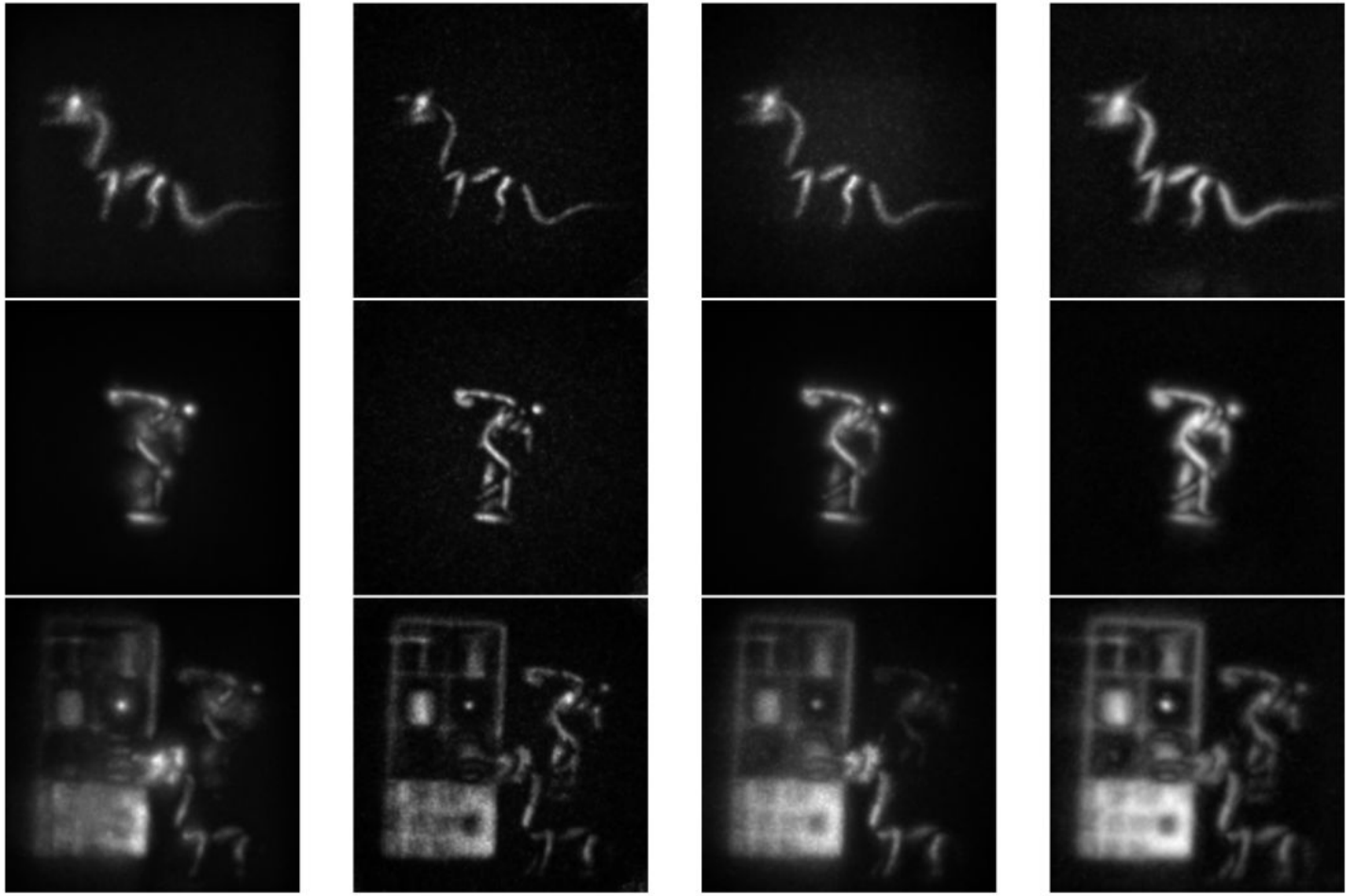
- Can be implemented as a backprojection
- Theoretically equivalent to F-K migration - derived from scalar theory

$$\rho(\mathbf{x}) = \oint_{S_0} \left(\frac{4\cos\theta}{cr} \frac{\partial \psi}{\partial t}(\mathbf{x}_s, 2r/c) - \frac{2\cos\theta}{r^2} \psi(\mathbf{x}_s, 2r/c) \right) \mathrm{d}s$$

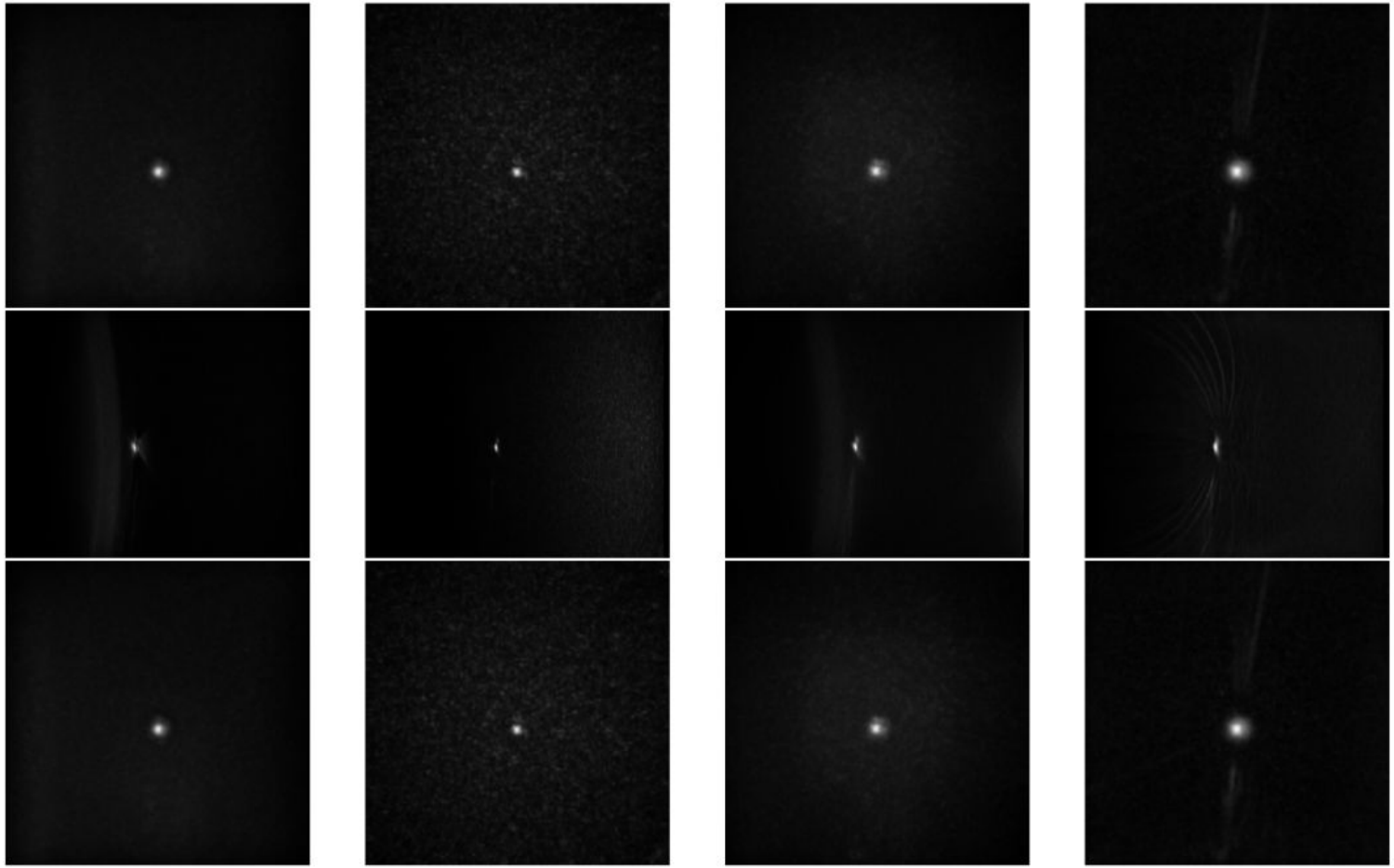
Second Order Kirchhoff Migration

- Attempt to model propagation from light source to hidden scene
- Looks a lot like the LCT

$$f(\mathbf{x}) = 4 \oint_{S_0} \frac{\cos^2 \theta}{r^2} \left(\frac{1}{c^2} \frac{\partial^2 \psi}{\partial t^2}(\mathbf{x}_s, 2r/c) - \frac{2}{cr} \frac{\partial \psi}{\partial t}(\mathbf{x}_s, 2r/c) + \frac{1}{r^2} \psi(\mathbf{x}_s, 2r/c) \right) \mathrm{d}s$$



(a) Kirchhoff migration (b) 2nd-order migration (c) F-K migration [2] (d) LCT [5]



(a) Kirchhoff migration (b) 2nd-order migration (c) F-K migration [16] (d) LCT [20]

Relating F-K migration to volumetric albedo models

- Derive F-K from a volumetric albedo model (inspired by radar literature)
- Derive a volumetric albedo model using a slightly modified F-K derivation
- Main difference is **a ramp filter**

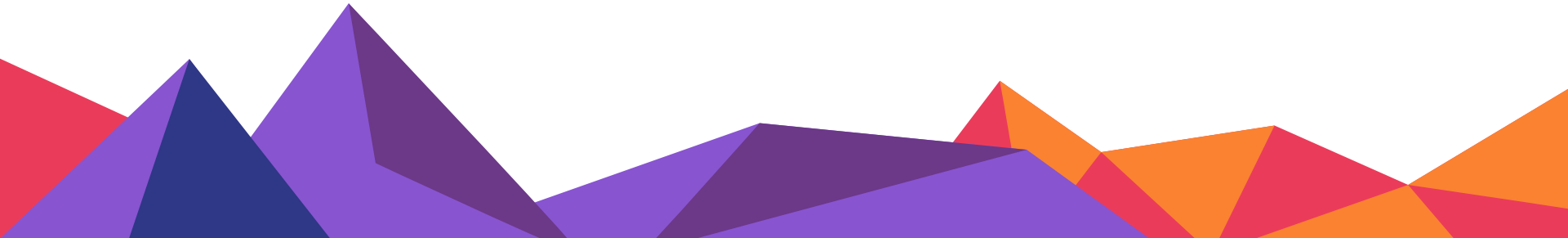
Questions?

Colorization through Optimization

Shruti Chidambaram | 15-463, Fall 2020

Colorization: Background

- Colorization typically requires:
  - Image segmentation + tracking segments over image sequences
  - Lots of user input/intervention
- Generally an ill-formed problem
- Contemporary techniques: deep learning to predict likely colors of pixels



Acknowledgement

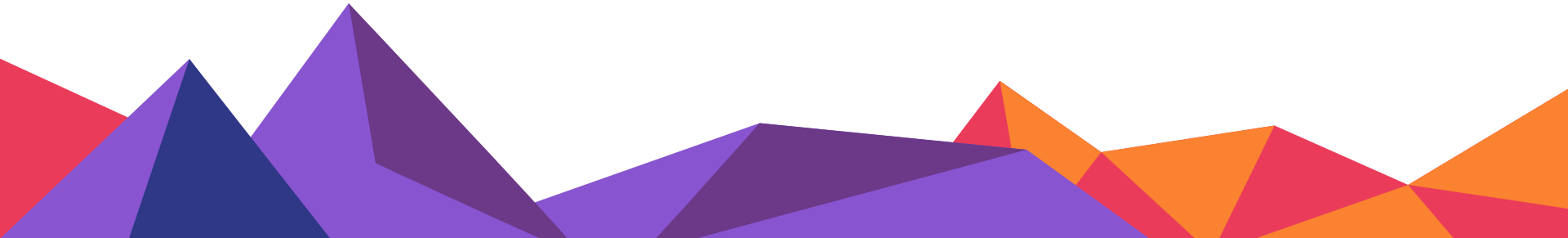
My project reimplements the algorithm presented in

Colorization using Optimization

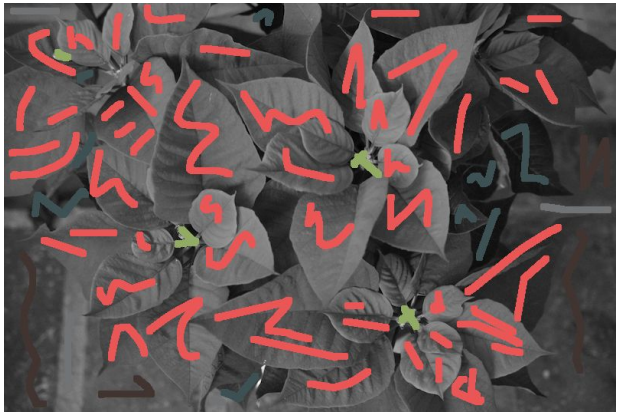
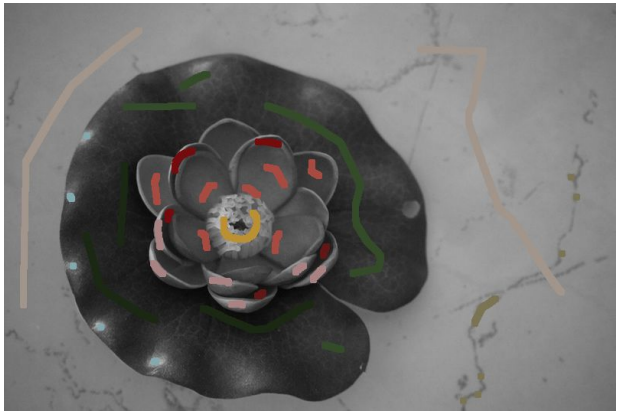
(Anat Levin, Dani Lischinski, and Yair Weiss, 2004)

Algorithm

- Neighboring pixels should have similar colors if their intensities are similar
- Working in YUV color space
  - Y → intensity
  - U, V → chrominance (encode color)
- Minimize difference between color at a pixel and the weighted average of colors at neighboring pixels, subject to constraints of the user-specified colors
- Optimization problem → solve large, sparse system of linear equations



Results

MARKED-UP GRAYSCALE	COLORIZED	GROUND TRUTH
		
		

Results: Experimenting with Varied User Input

MARKED-UP GRAYSCALE







COLORIZED







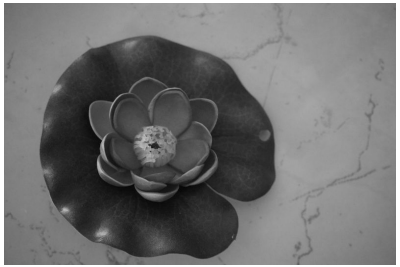
GROUND TRUTH



Experimenting with a Deep Learning Approach

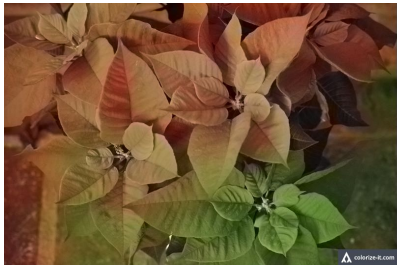
INPUT

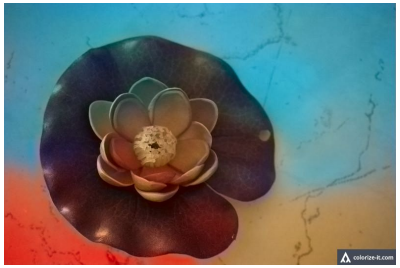






OUTPUT







GROUND TRUTH







Autocolorization results from demos.algorithmia.com/colorize-photos
paper: *Colorful Image Colorization (*Zhang, Isola, Efros; 2016)

Non-Realistic Rendering App

Han Deng (handeng)
Heshan Liu (heshanl)

Motivation

Nowadays there are a lot of different types of filters you can choose when you are using camera app. We would like to make something different and slightly more complicated rather than just adjust the color of the image.



Original



Sketch



Blending



Watercolor

Goal

- Implement multiple image stylization effects on **mobile device**
- Effects
  - Pencil Sketch
  - Watercolor Painting
  - Image Blending
- Android App
  - Take photos or select photo from the album.
  - Then do the target filter on the taken/selected photo.

Environment Setup

- Mobile Device: Android

- Package
  - Original Bitmap Factory provided by Android
  - RenderScripts: Framework for running computationally intensive tasks

Algorithms: Pencil Sketch

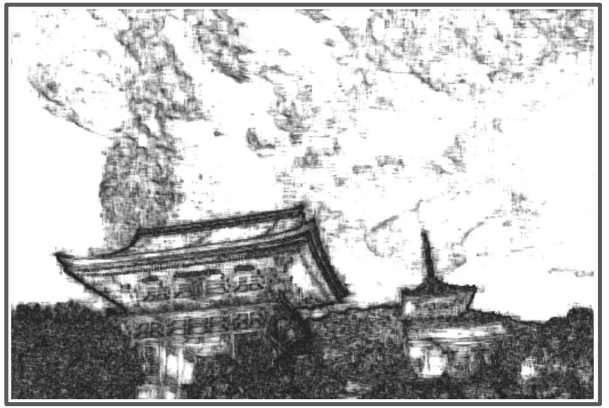
- Two types of implementation
    - Grayscale Image divided by Gaussian Blurred result
    - Bilateral Filtering with a shade factor -> into Grayscale Output.



Input Image

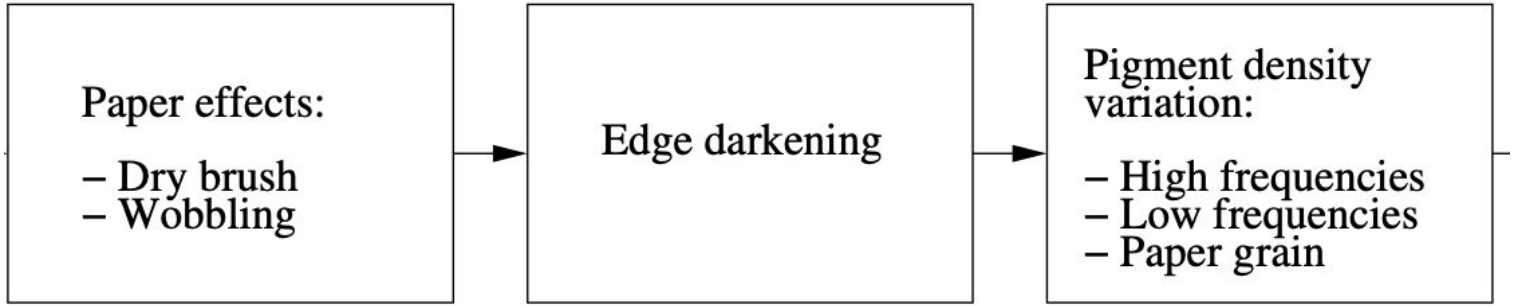


Output 1



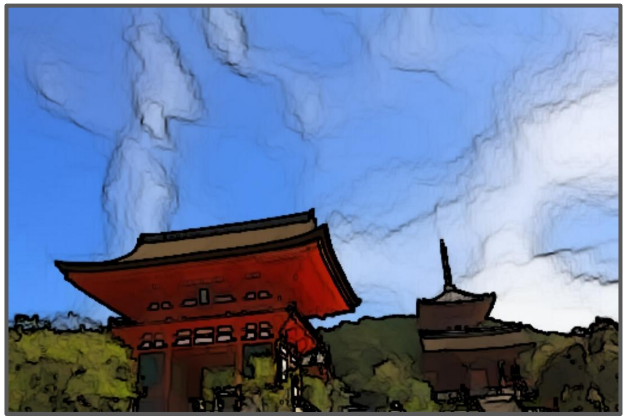
Output 2

Algorithms: WaterColor





Input Image



Output Image

Algorithms: Image Blending - Poisson Blending

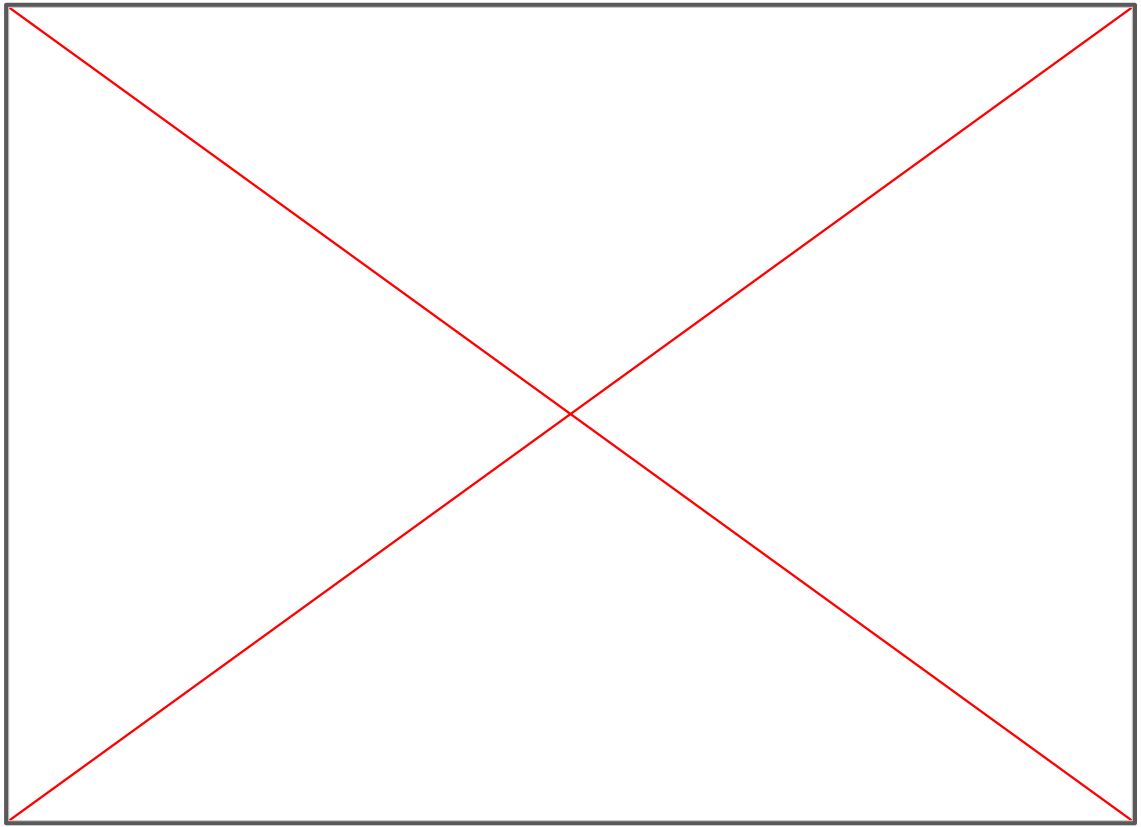
- Goal: Seamless clone the source image into the target image

- Mix gradients of source and target image in target region:

$$\text{for all } \mathbf{x} \in \Omega,\ \mathbf{v}(\mathbf{x}) = \begin{cases} \nabla f^*(\mathbf{x}) & \text{if } |\nabla f^*(\mathbf{x})| > |\nabla g(\mathbf{x})| \\ \nabla g(\mathbf{x}) & \text{otherwise.} \end{cases}$$

Pérez, P., Gangnet, M., & Blake, A. (2003). Poisson image editing. In *ACM SIGGRAPH 2003 Papers* (pp. 313-318).

Algorithm and app Demo: Image blending



Possible Improvement

1. More complicated filters could be added like oil painting style

2. The camera UI could be more elegant

Thank you!

Uncertainty in Radiometric Calibration

15663 Course Project

Advait Gadhikar



Goal

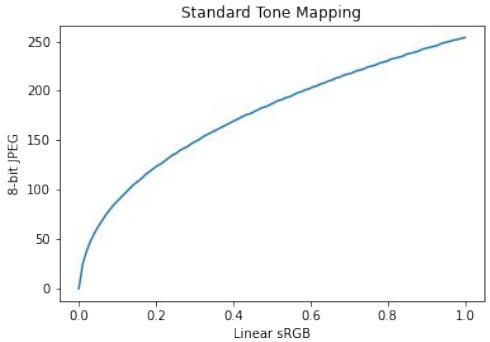
- Quantifying the uncertainty in Inverse Tone Mapping function
- Estimate a distribution to quantify the uncertainty in linearizing tonemapped images
- Using this uncertainty model to combine a HDR stack

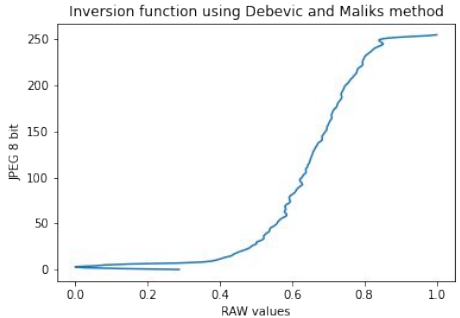
Motivation

- Linearizing images is an important preprocessing step in many Computer Vision Applications
- Tone mapping for narrow gamut displays is a lossy compression
- Standard methods to linearize JPEG images use a deterministic mapping
- Such a mapping doesn’t consider information lost while converting to JPEG
- A Probabilistic Model for inversion (JPEG to RAW) is more informative
- Gives better performance on downstream tasks like HDR imaging

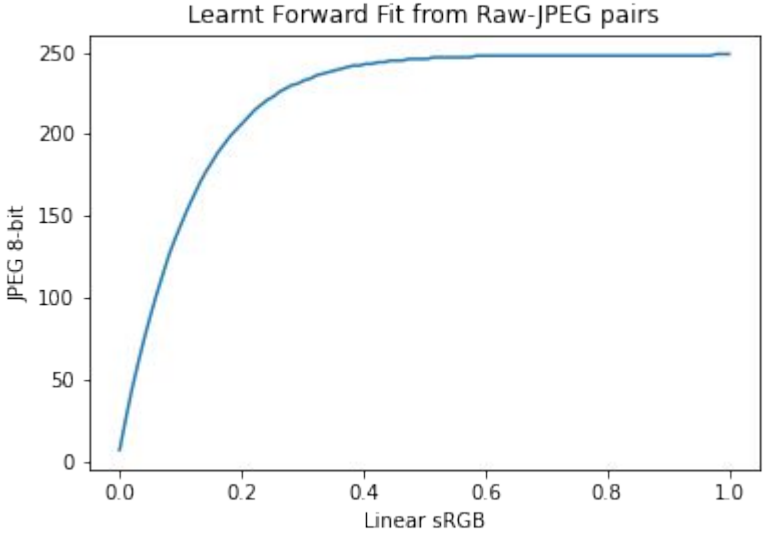
Standard Tone Map vs Learnt Tone Map

$$C_{\text{non-linear}} = \begin{cases} 12.92 \cdot C_{\text{linear}}, & C_{\text{linear}} \leq 0.0031308 \\ (1 + 0.055) \cdot C_{\text{linear}}^{\frac{1}{2.4}} - 0.055, & C_{\text{linear}} \geq 0.0031308 \end{cases}$$





If tone mapping is deterministic, Debevic and Malik's method assumes this mapping



$$\tilde{\mathbf{y}} = f(\mathbf{v}_c^T \mathbf{x})$$
$$\mathbf{y} = Q(B(\tilde{\mathbf{y}}) + g(\tilde{\mathbf{y}}))$$

f is a polynomial and v is a linear transform, with g being a correction factor

RMSE for Rendering Function (Raw to JPEG)

- RMSE on a stack of images collected under multiple exposure
  - 16 images, with 1 stop exposure step
- RMSE: 16.995 (For 8-bit JPEG images)





Sample Images in stack

Inverse Distribution for Linearization

- The inverse distribution is assumed to be a Multi Variate Gaussian

$$\begin{aligned}\tilde{p}(x|y) &= \mathcal{N}(x|\mu(y), \Sigma(y)), \\ \mu(y) &= \int x p(x|y)\, dx, \\ \Sigma(y) &= \int (x-\mu(y))\,(x-\mu(y))^T\, p(x|y)\, dx.\end{aligned} \qquad p(x|y) = \frac{1}{Z}\, p(x) \exp\left(-\frac{\|y - \mathbb{J}(x)\|^2}{2\sigma_f^2}\right)$$

- Here J is the forward rendering model
- The integral is approximated over a grid of pixel values
    - A grid of 10 x 10 x 10 points was used for rgb values in $[0,1]^3$

HDR with Uncertainty

- Combining exposure stack to get HDR Image with Uncertainty

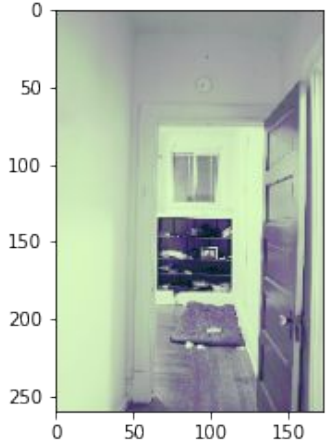
$$x = \arg\min_x \sum_i \|\mu_i - \alpha_i x\|^2 = \frac{\sum_i \alpha_i \mu_i}{\sum_i \alpha_i^2}$$

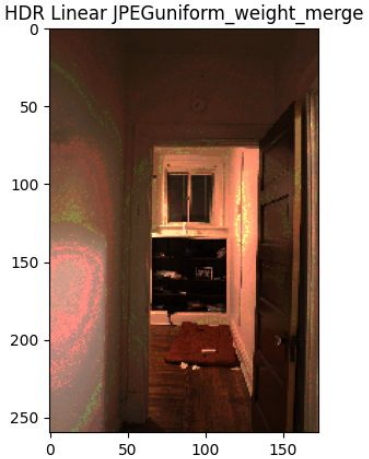
$$\Sigma_i = \frac{\Sigma(y_i) + \sigma_z^2 I_{3\times 3}}{\alpha_i^2}$$

$$x = \arg\max_x \prod_i \mathcal{N}(x|\mu_i, \Sigma_i)$$

$$= \left(\sum_i \Sigma_i^{-1}\right)^{-1} \left(\sum_i \Sigma_i^{-1}\mu_i\right)$$

Estimated HDR image with uncertainty





Thank You

Questions!?

References

- Chakrabarti, Ayan, et al. "Modeling radiometric uncertainty for vision with tone-mapped color images." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.11 (2014): 2185-2198.
- Paul E. Debevec and Jitendra Malik. 1997. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (*SIGGRAPH '97*). ACM Press/Addison-Wesley Publishing Co., USA, 369–378. DOI:https://doi.org/10.1145/258734.258884

Carnegie Mellon University

One-Shot to *Vertigo*: Novel View Synthesis using Light Field Cameras

Rohan Rao
(rgrao@andrew.cmu.edu)

Introduction

- Extension of The “Vertigo Effect” on Your Smartphone: Dolly Zoom via Single Shot View Synthesis [1], a CVPR2020 paper on single-shot view synthesis. Also implements US Patent by Lytro/Google, “*Generating dolly zoom effect using light field image data*”. [2]
- Creating a “dolly-zoom” or “vertigo” effect, popularized by Alfred Hitchcock’s 1958 movie, **Vertigo**.
- Keep the subject in focus while moving the camera and zooming simultaneously, creating a tunnelling/falling effect. Also widely used for aerial cinematography shots.





Carnegie Mellon University

Single-camera Pipeline: Pre-processing

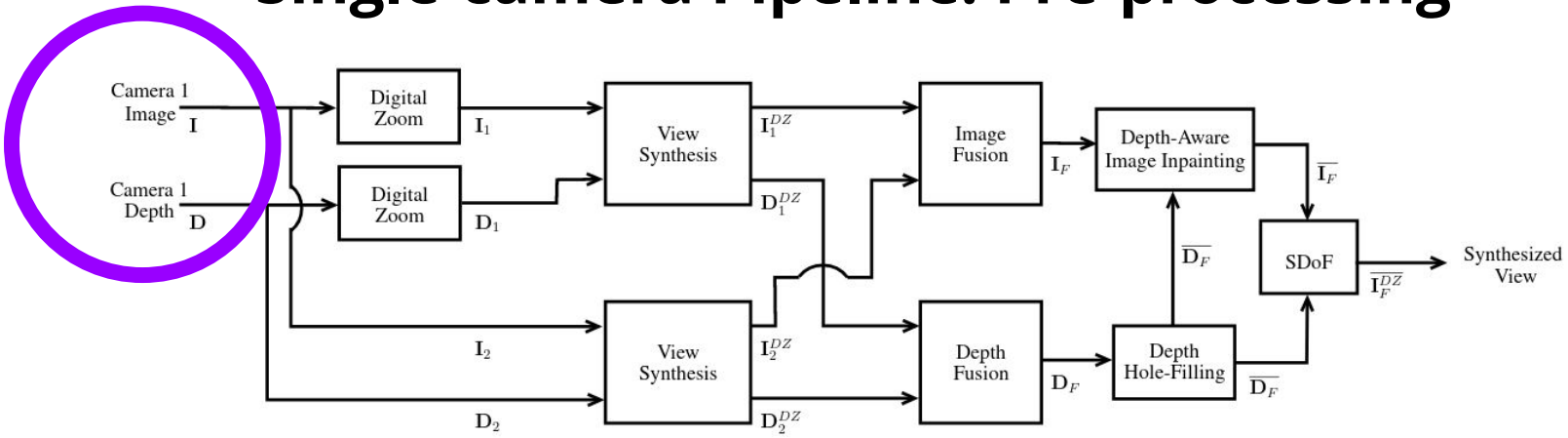


Figure 8: Single camera single shot synthesis pipeline
(from [1])

- Process RAW lightfield images (calibration, demosaicking, etc) using the PlenPy library [3].
- Obtain depth image from lightfield using Structure Tensors [5] to estimate disparity [3].
- Used the INRIA Lytro dataset [4] for testing calibration and depthmap results.

Carnegie Mellon University

Single-camera Pipeline: Digital Zoom

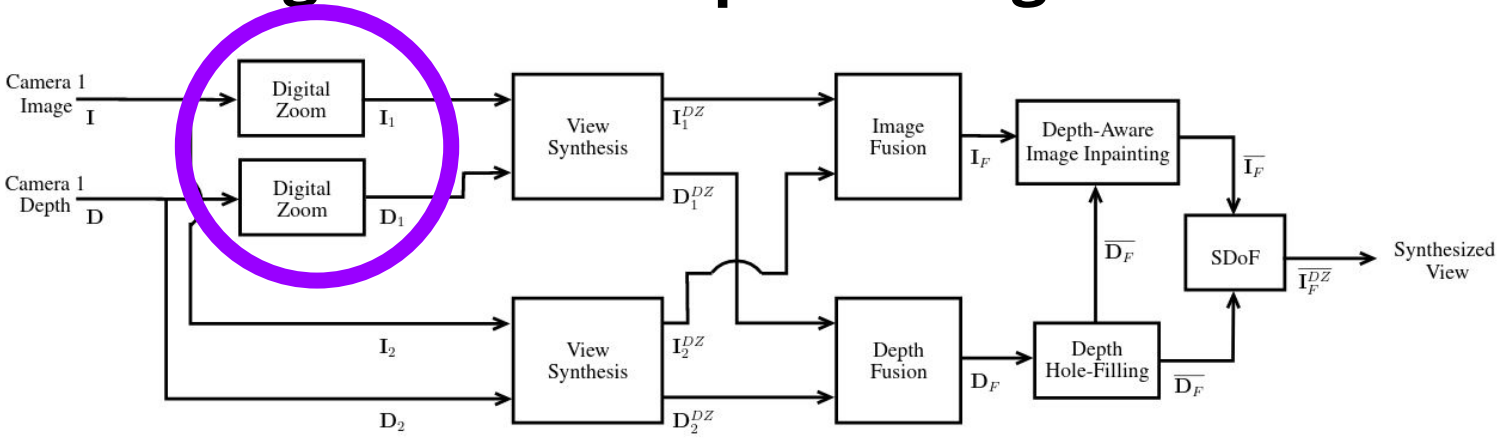


Figure 8: Single camera single shot synthesis pipeline
(from [1])

- Can obtain closed-form expression for digital zoom using focal length and camera intrinsics.
- Implemented clipped zooming using OpenCV and efficient resizing.

Carnegie Mellon University

Single-camera Pipeline: View Synthesis

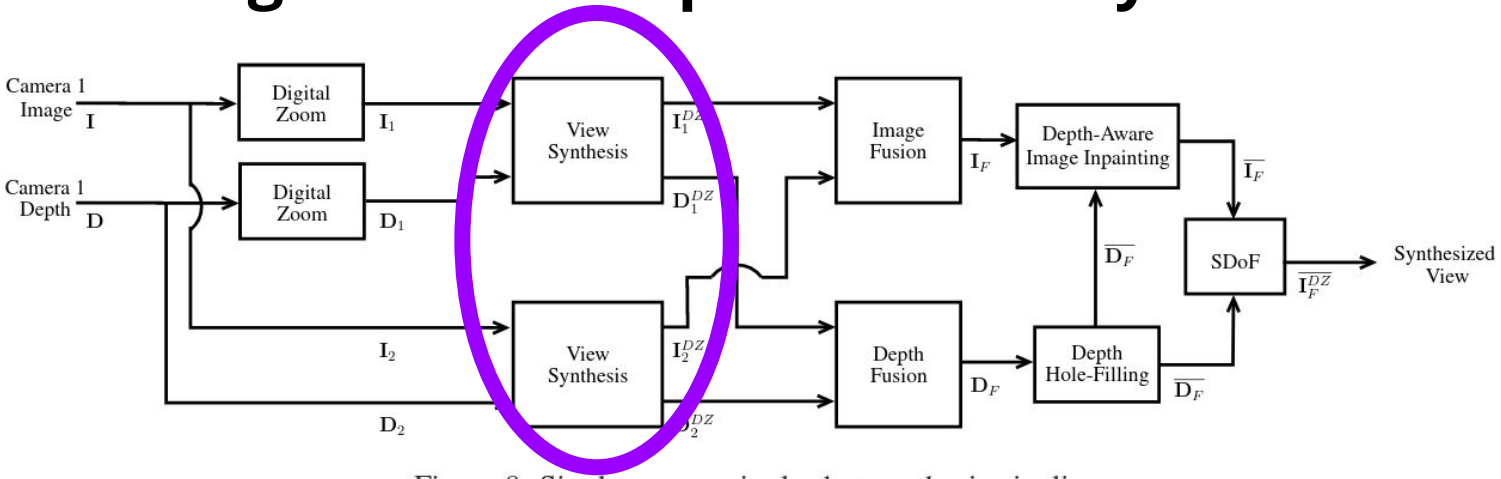


Figure 8: Single camera single shot synthesis pipeline

(from [1])

- Can obtain closed-form expression for single-shot view synthesis and warp image and depth map using z-buffering and forward warping.
- Implemented with a z-buffer and looping over all pixels (more efficient in C++ though)

Carnegie Mellon University

Single-camera Pipeline: Image/Depth fusion

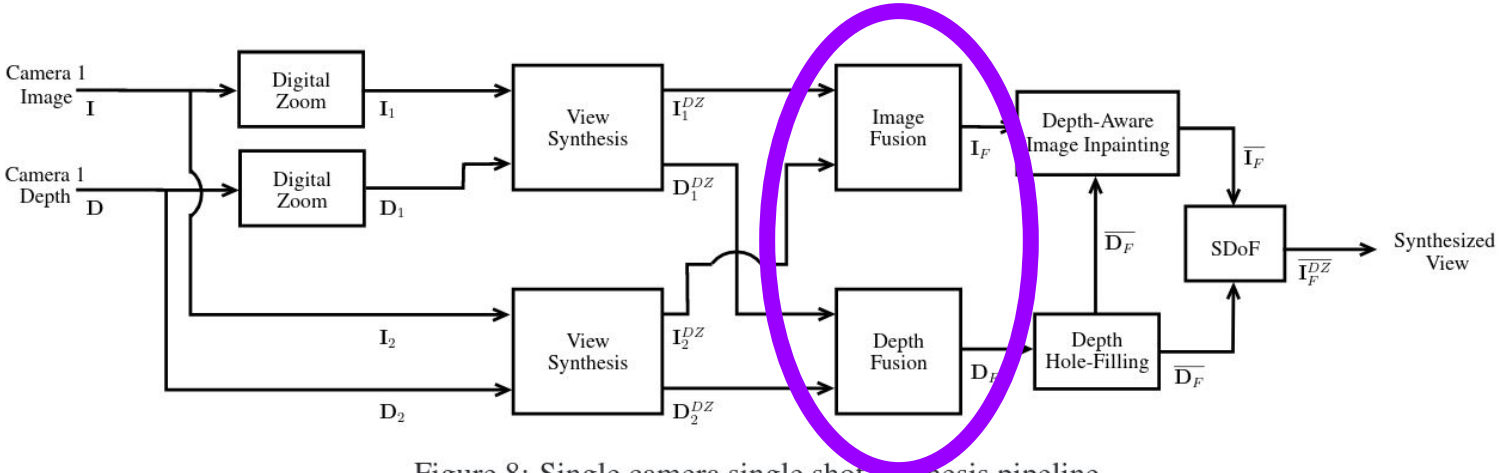


Figure 8: Single camera single shot synthesis pipeline

(from [1])

- Can fuse the images by creating image/depth occlusion masks from the previous step (regions where warping did not update the value)

Carnegie Mellon University

Single-camera Pipeline: Depth Hole Filling

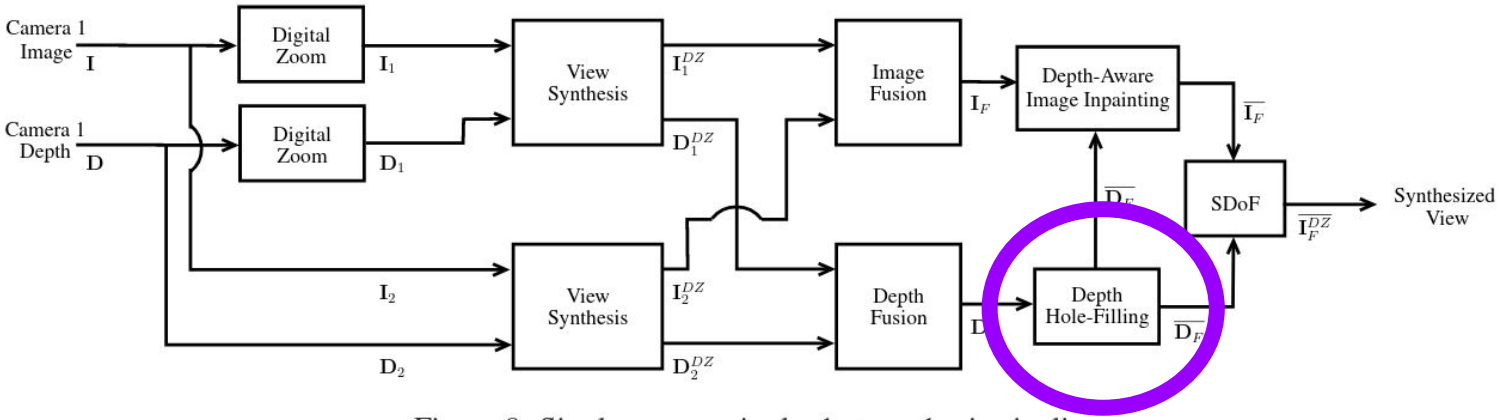


Figure 8: Single camera single shot synthesis pipeline
(from [1])

- Simplistic method - just takes max of neighboring values to replace missing

Carnegie Mellon University

Single-camera Pipeline: Image Inpainting

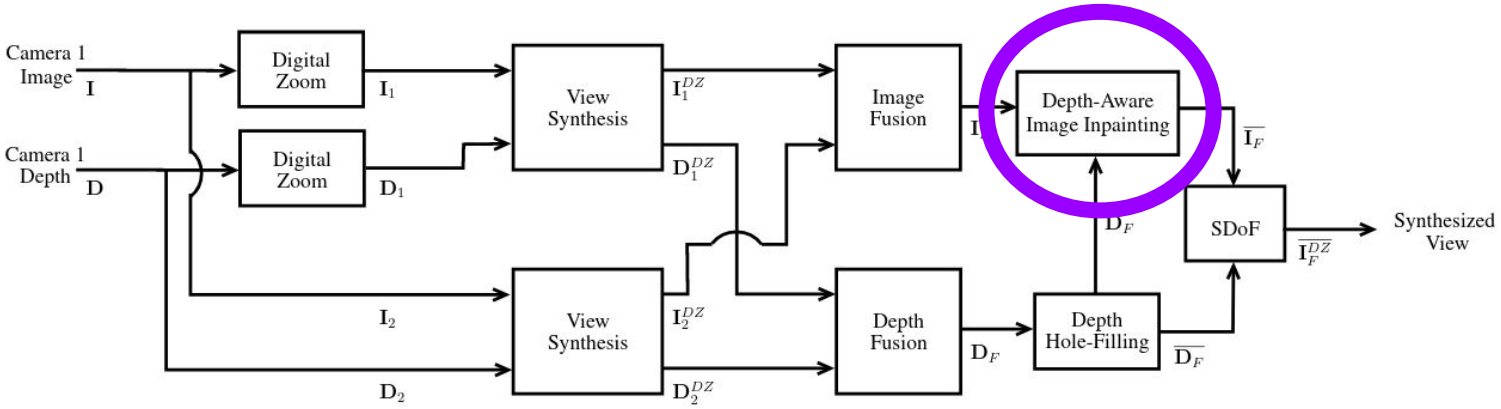


Figure 8: Single camera single shot synthesis pipeline
(from [1])

- Extension: convert continuous depth values to discrete ones using K-Means clustering
- Inpainting using masks and nearest valid pixel in the row

Carnegie Mellon University

Single-camera Pipeline: Shallow Depth of Field (SDoF)

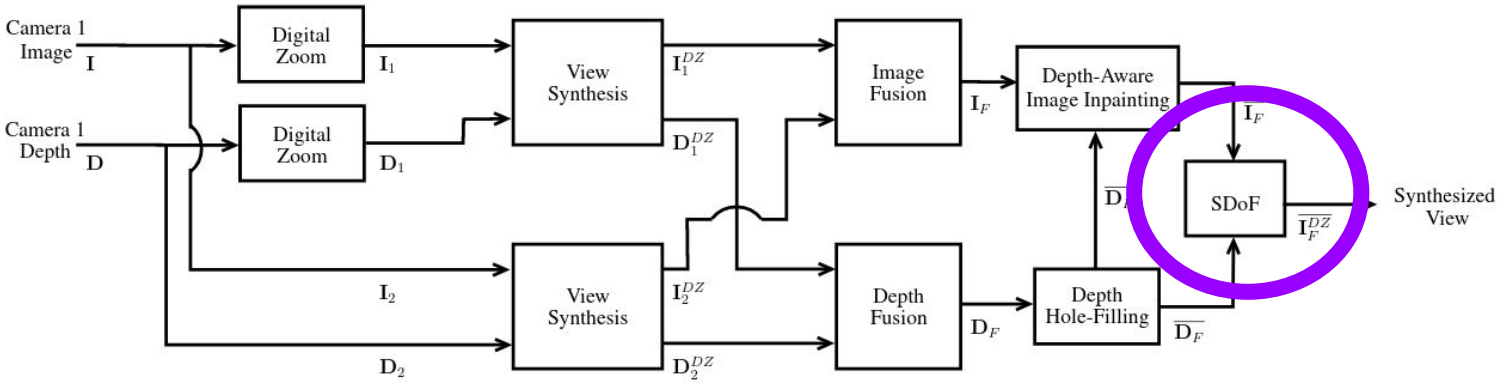
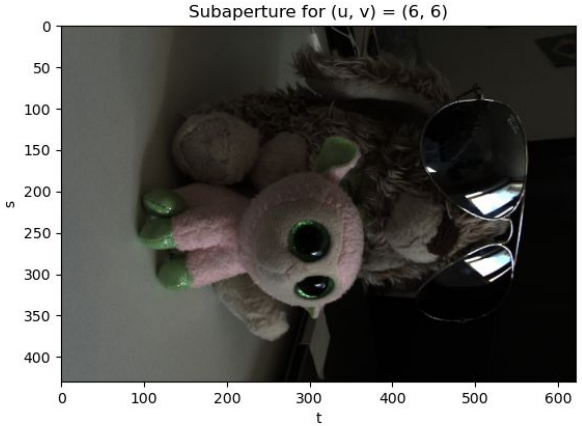


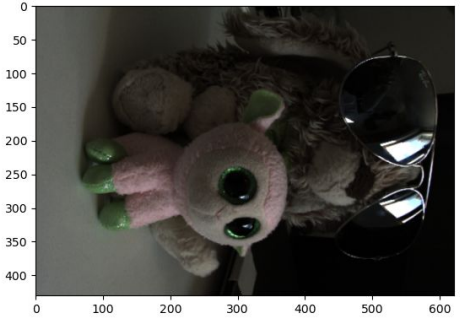
Figure 8: Single camera single shot synthesis pipeline
(from [1])

- Extension: smoother blending of depth segments using finite bounding values, then scaling
- Blur depth segments with different sigma values to keep subject focused

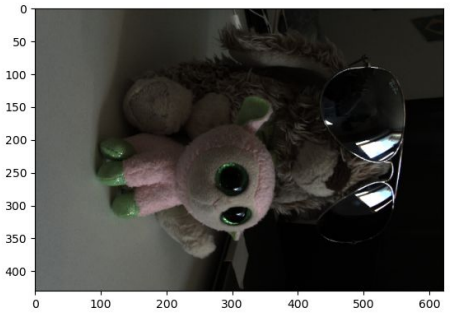
Carnegie Mellon University

Intermediate steps towards final results

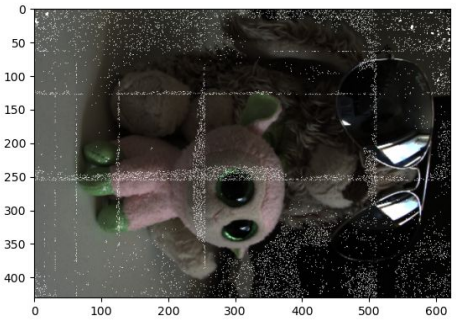




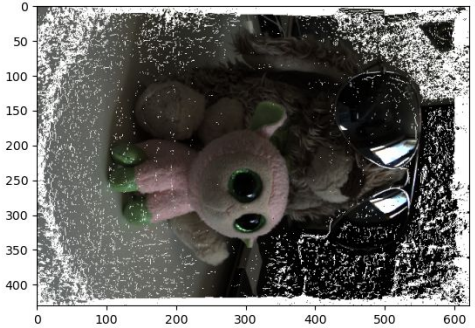
Post digital zoom (I1)



Without digital zoom (I2)



$I_{z_1}^{D_1}$



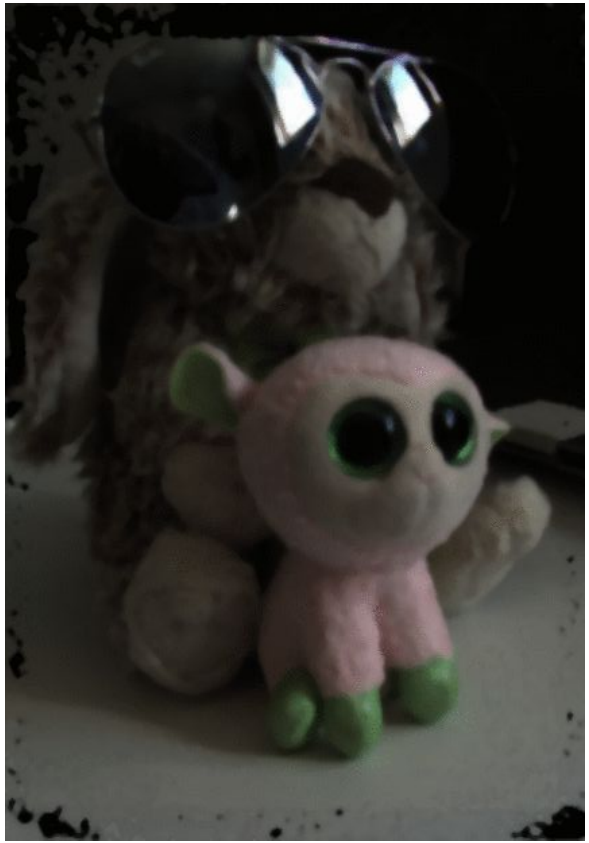
$I_{z_2}^{D_2}$



$I_{z_1}^{D_1}$

Carnegie Mellon University

Final result



Carnegie Mellon University

Extensions of prior work

- Improved depth map estimate obtained from a light field instead of monocular depth estimation from a single RGB image.
- Algorithm for using continuous-valued depth maps through K-Means based discretization.
- Algorithm for Shallow Depth of Field (SDoF) and smoother merging of multiple image segments to prevent artifacts at the edges.

Ideas for future work

- Improve the inpainting approaches using more sophisticated search for valid pixels (currently searches same row) or using deep-learning for inpainting (experimented but didn't show results here)
- Obtain more realistic depth values with calibration, since this will give better results
- Improve the view synthesis from light fields using Multiplane Images (learning view synthesis using MPIs), this can be used to create more novel views from a single image/video sequence

Carnegie Mellon University

References

1. Liang et al., *The "Vertigo Effect" on Your Smartphone: Dolly Zoom via Single Shot View Synthesis*, *CVPRW 2020.*
2. US Patent US8971625B2, *Generating dolly zoom effect using light field image data*, *Lytro Inc.*
3. Schambach, Maximilian and Puente León, Fernando, *Microlens array grid estimation, light field decoding, and calibration*, *IEEE Transactions on Computational Imaging, 2020.*
4. M. Le Pendu, X. jiang, C. Guillemot, *Light Field inpainting propagation via low rank matrix completion*, *IEEE Trans. on Image Processing, vol. 27, No. 4, pp. 1981-1993, Jan. 2018.*
5. Sven Wanner and Bastian Goldlueck, *Globally Consistent Depth Labeling of 4D Light Fields*

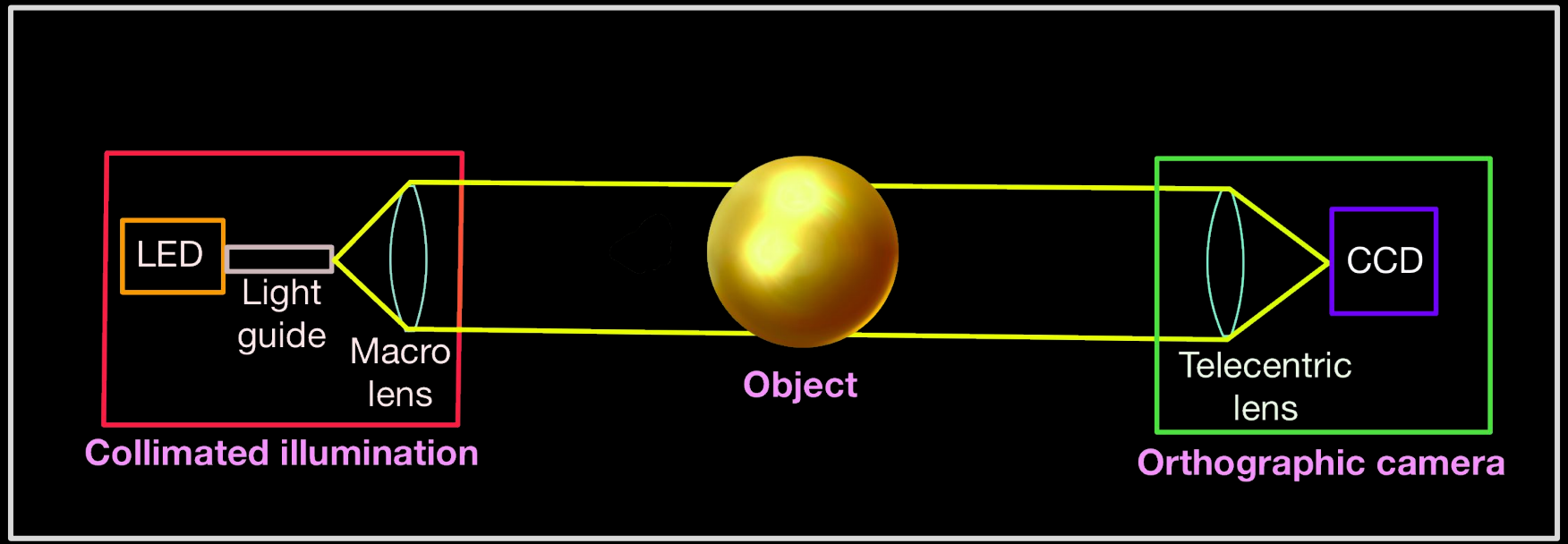
Thanks for listening! Questions?

Carnegie Mellon University

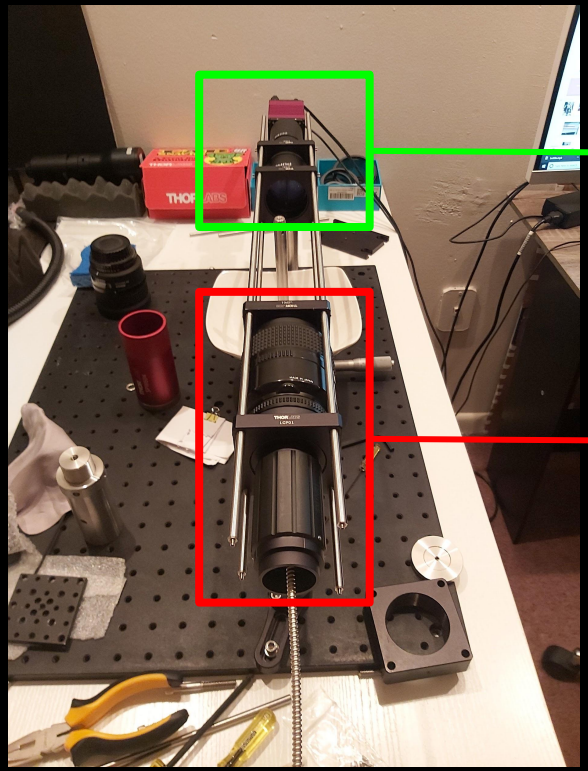
Epipolar scanning for shape-from-silhouette in scattering medium

Shirsendu S Halder
shirsenh

Shape from silhouette



Shape from silhouette

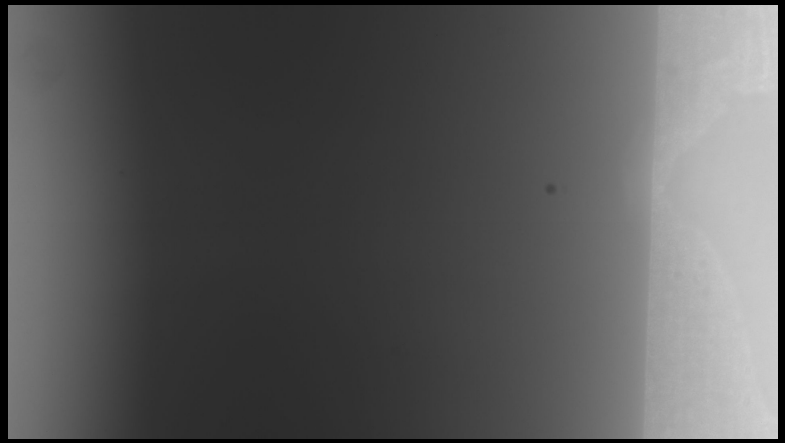


Orthographic camera

Collimated illumination

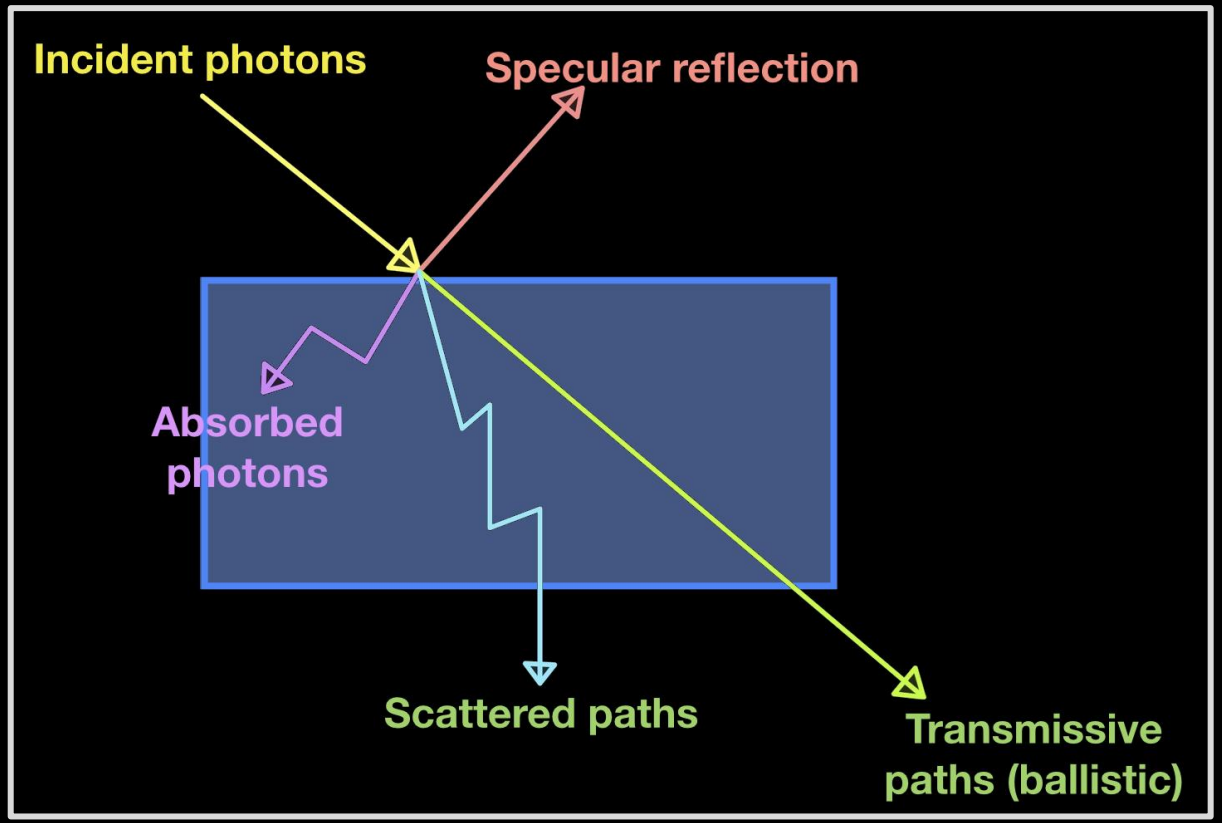


Object





Imaging through scattering medium

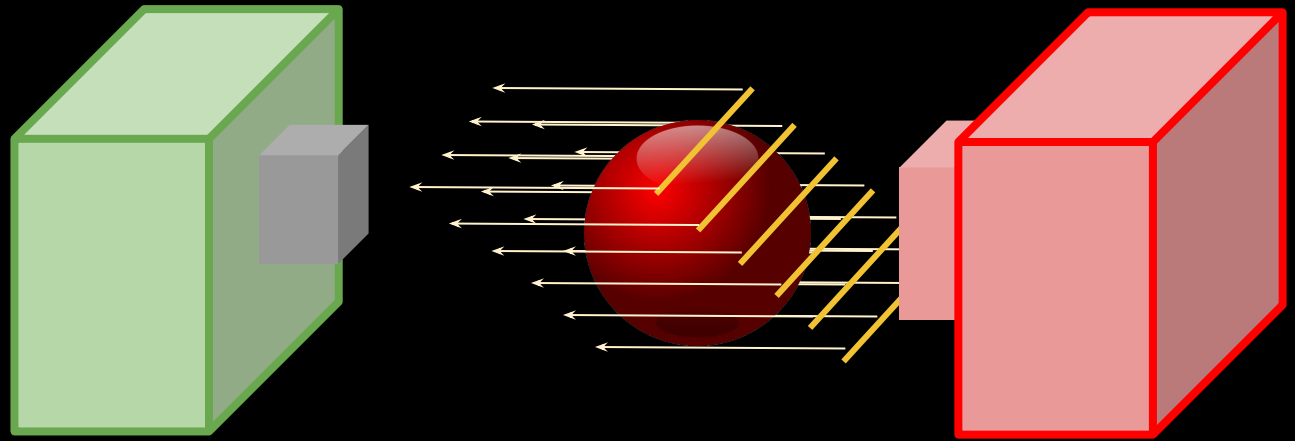


We want the transmissive paths (ballistic photons) as they sharpen the object image and also enhances contrast.

Imaging through scattering medium

Camera captures ballistic paths from each row.

Rows of an orthographic projector lit.

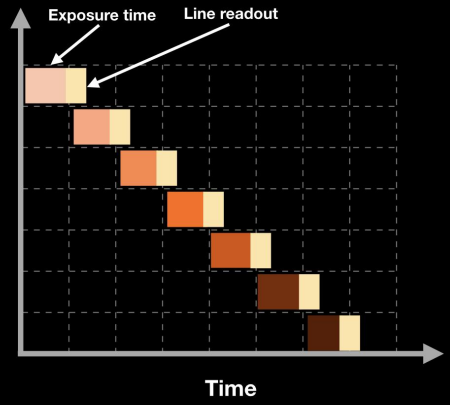


Telecentric camera

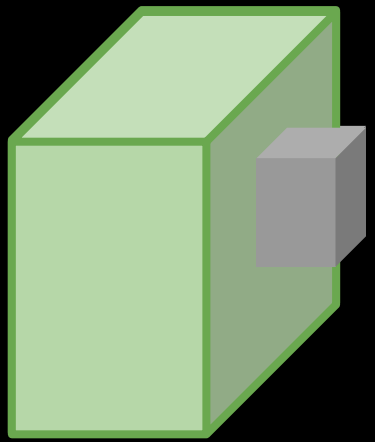
Orthographic projector

Combination of position cues, angle cues, and probing

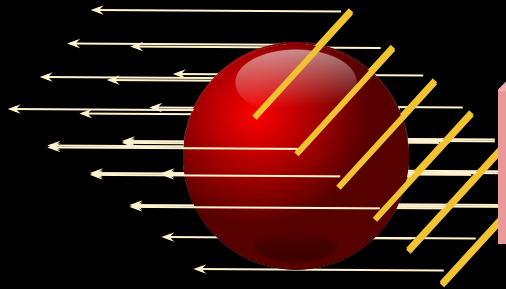
Imaging through scattering medium



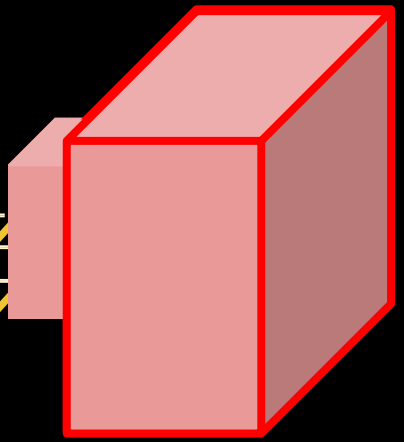
Camera captures ballistic paths from each row.



Telecentric camera



Rows of an orthographic projector lit sequentially.



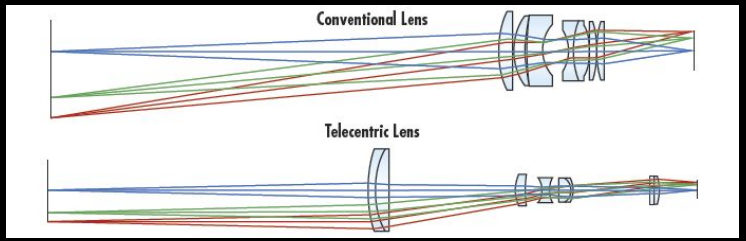
Orthographic projector

Why telecentricity?

- Low perspective distortion



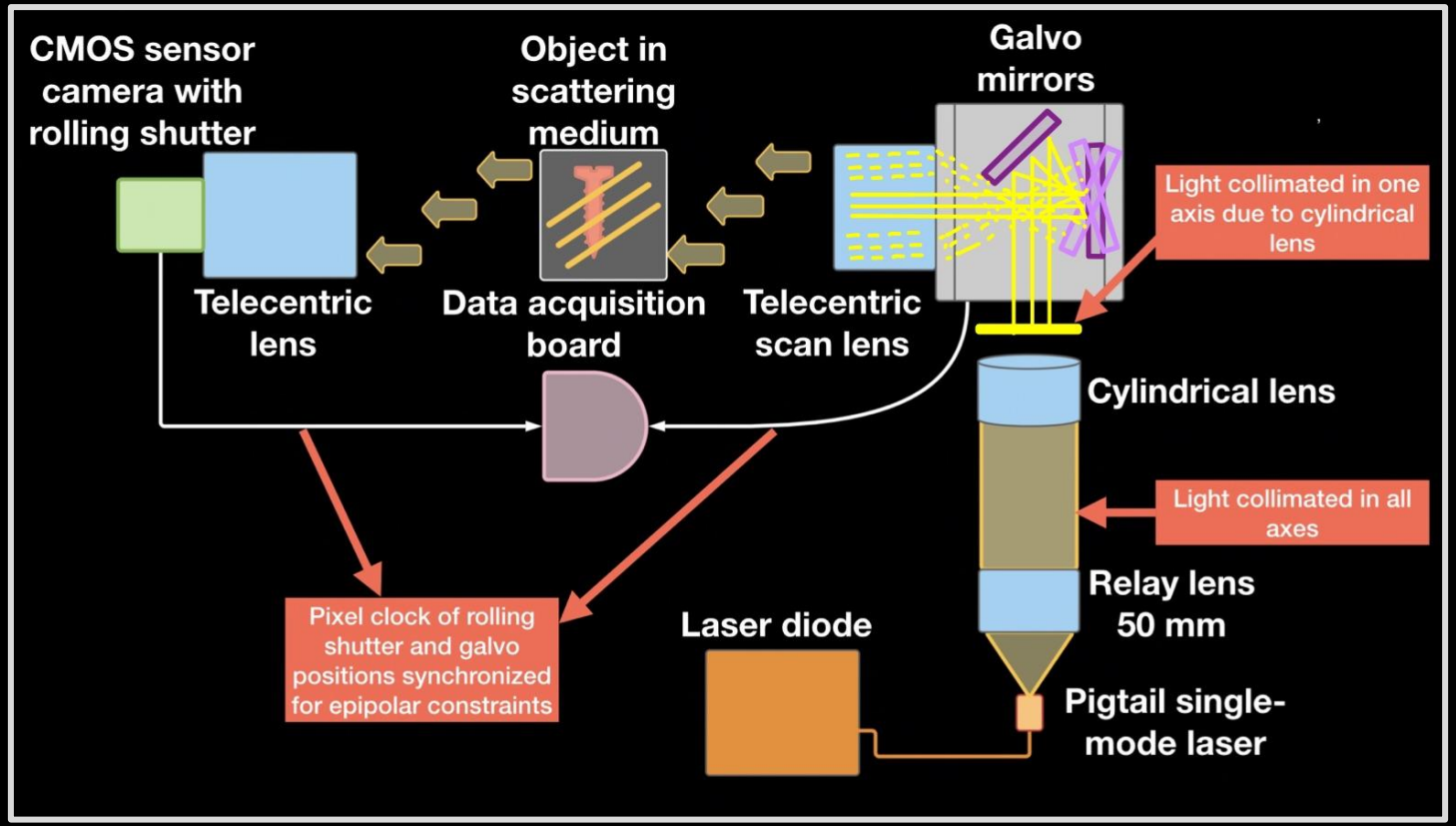
- Zero angular field-of-view: minimum parallax error



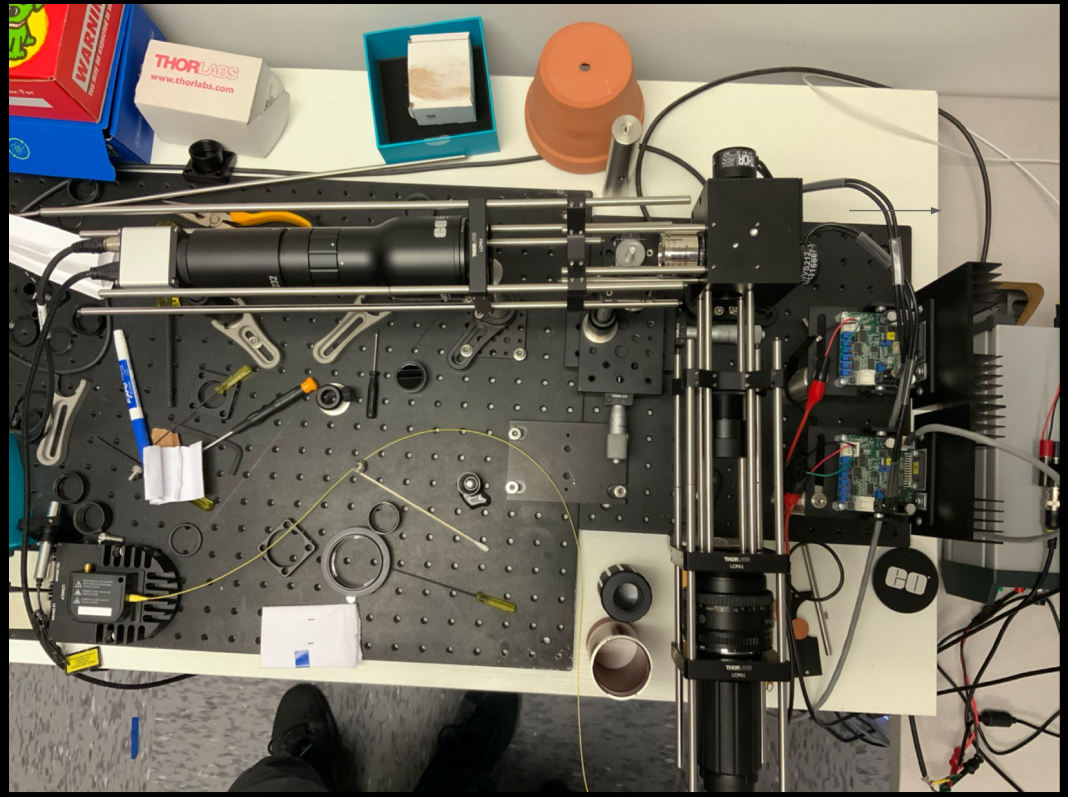
- Larger usable depth of field than conventional lenses due to symmetrical blurring

Image credit: https://www.edmundoptics.com/knowledge-center/application-notes/imaging/advantages-of-telecentricity/

Setup


CMOS sensor camera with rolling shutter
Object in scattering medium
Galvo mirrors
Light collimated in one axis due to cylindrical lens
Telecentric lens
Data acquisition board
Telecentric scan lens
Cylindrical lens
Light collimated in all axes
Relay lens 50 mm
Pixel clock of rolling shutter and galvo positions synchronized for epipolar constraints
Laser diode
Pigtail single-mode laser

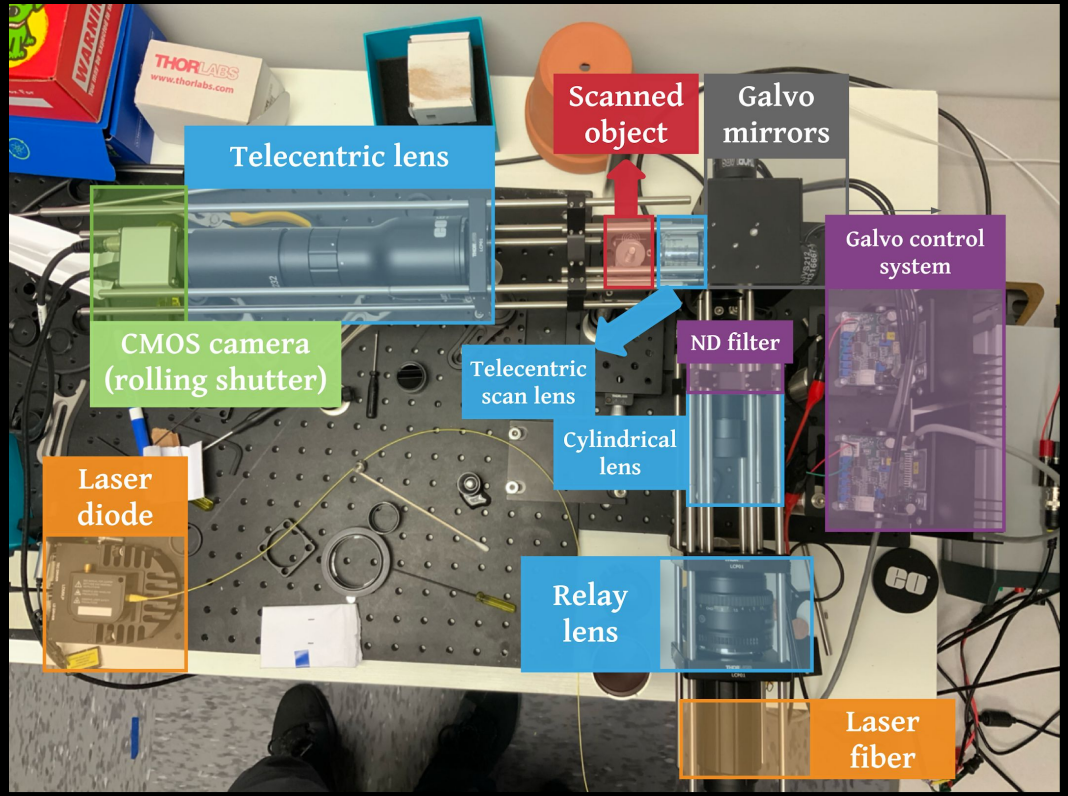
Setup



Syncing electronics



Setup


Scanned object
Galvo mirrors
Telecentric lens
Galvo control system
CMOS camera (rolling shutter)
ND filter
Telecentric scan lens
Cylindrical lens
Laser diode
Relay lens
Laser fiber

Laser line

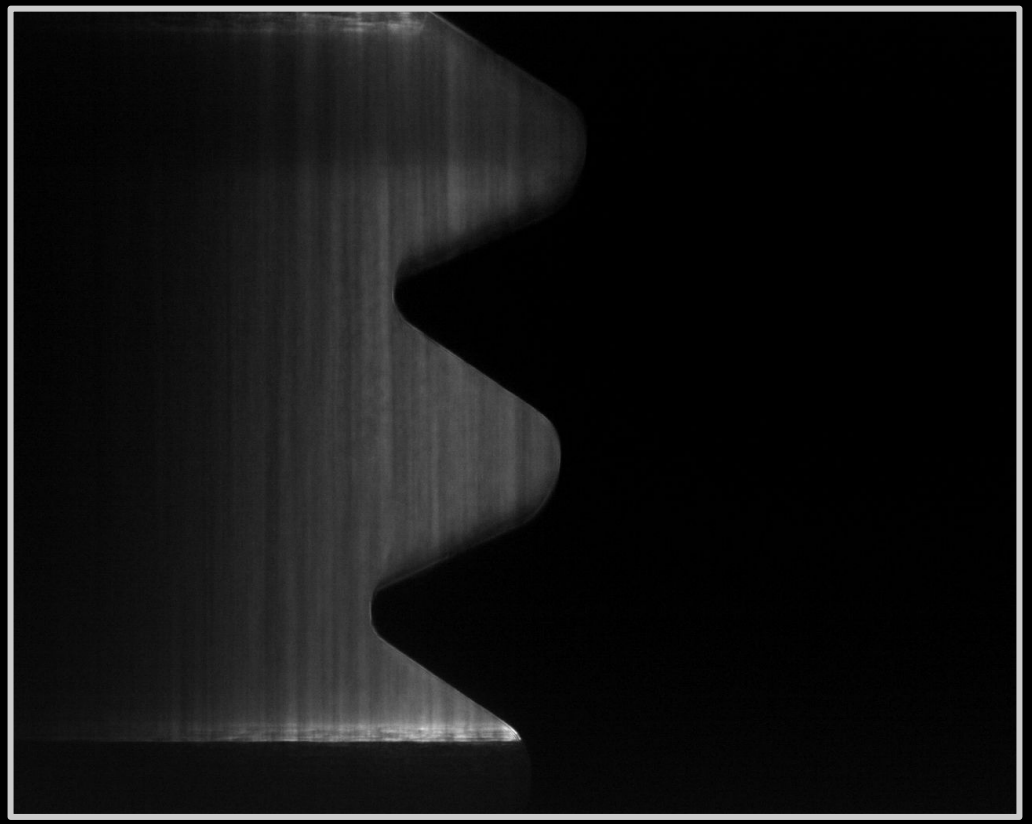


Object used for scanning



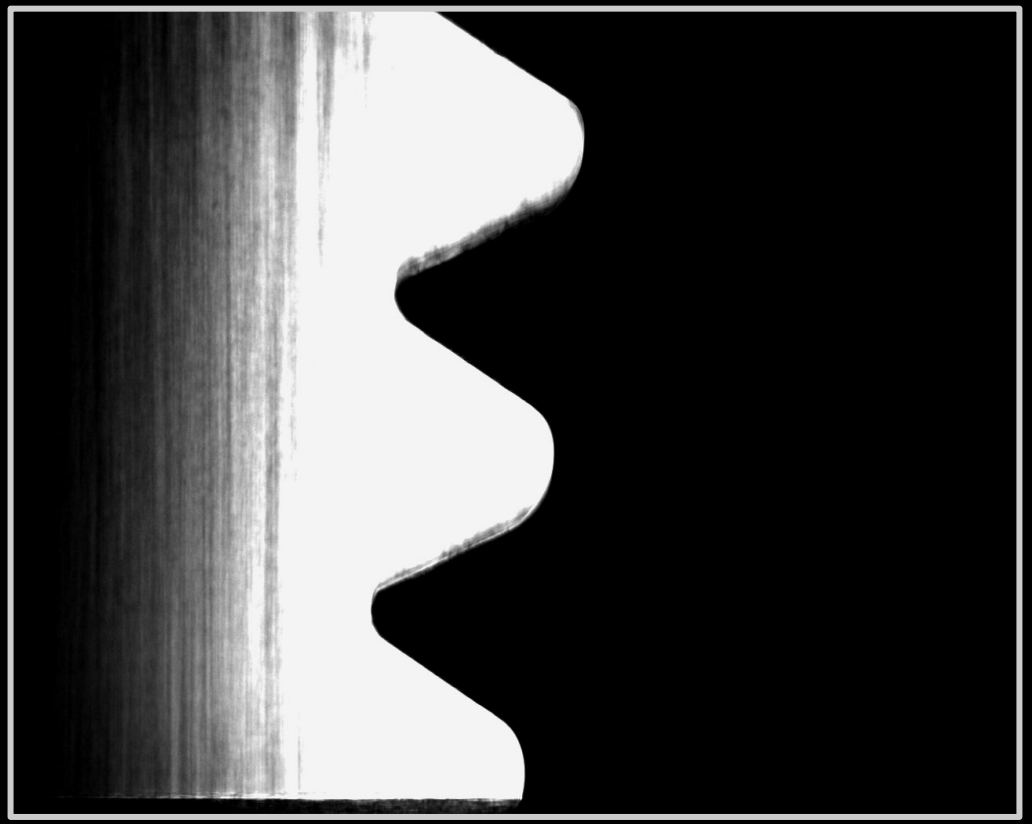
Scanning (in air)

Global



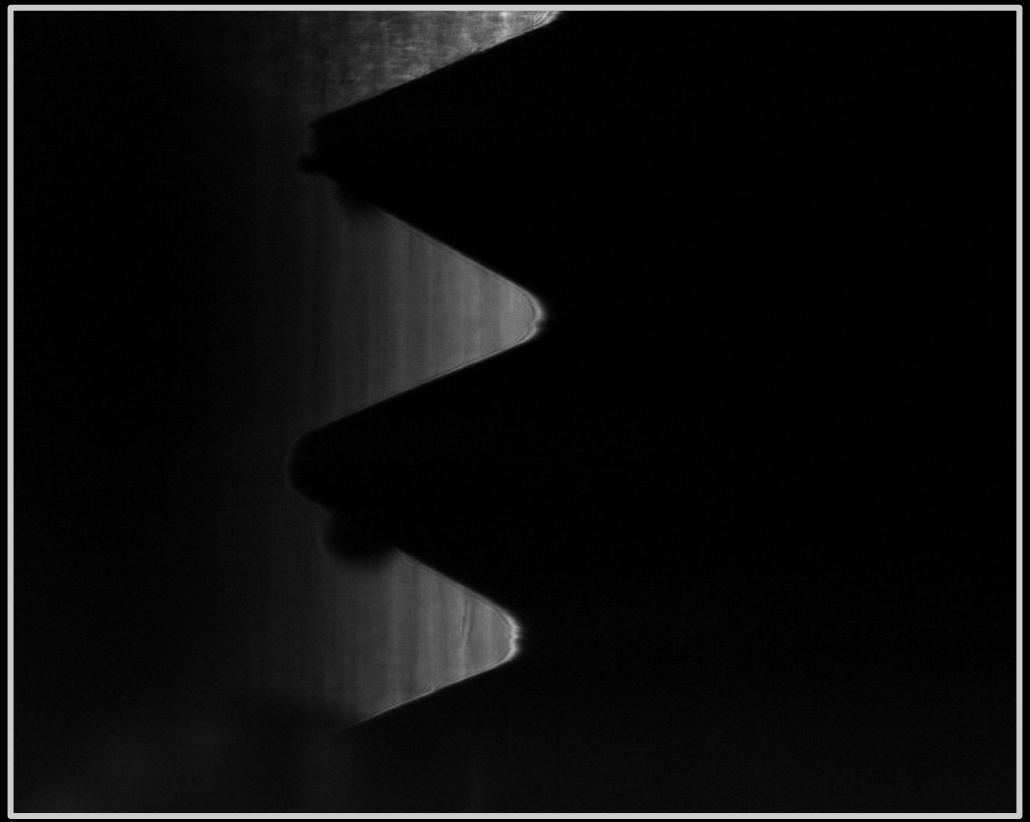
Scanning (in air)

Epipolar



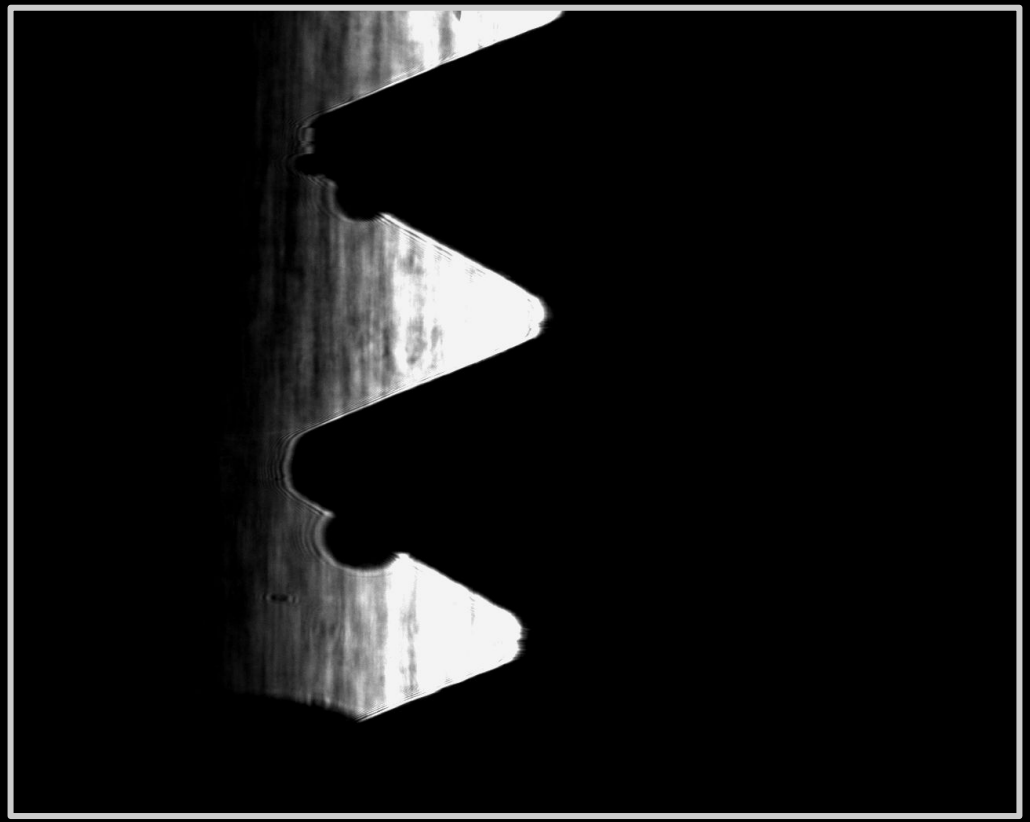
Scanning (in water)

Global



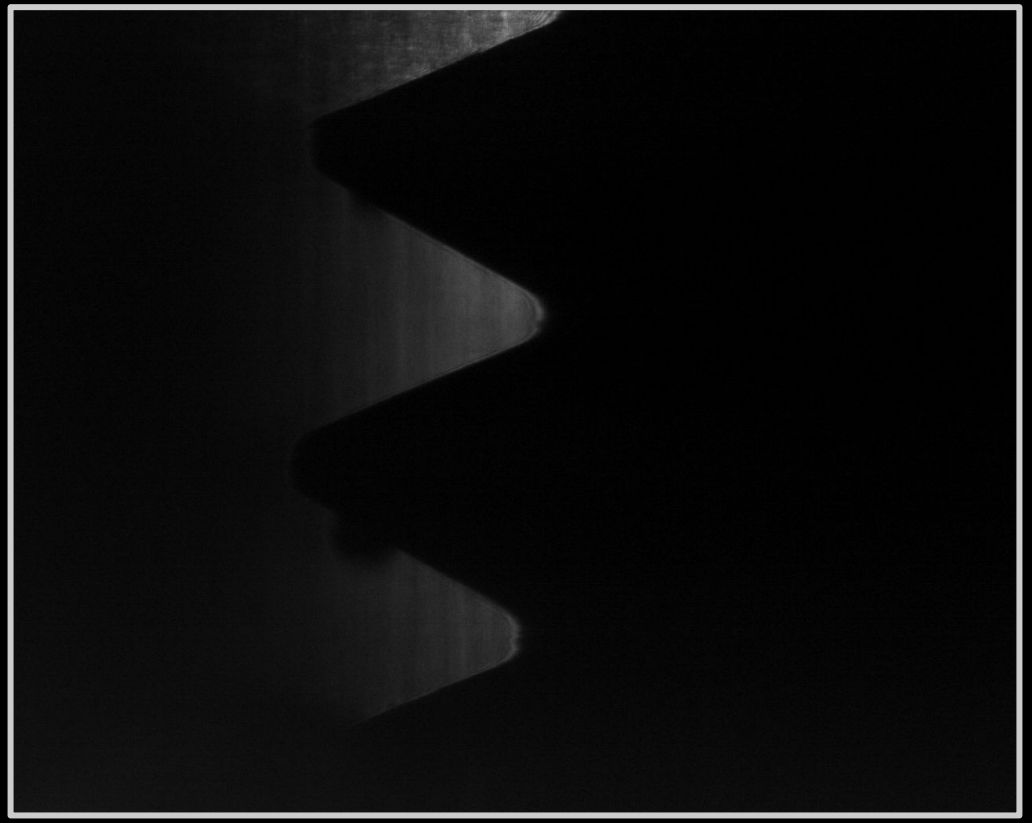
Scanning (in water)

Epipolar



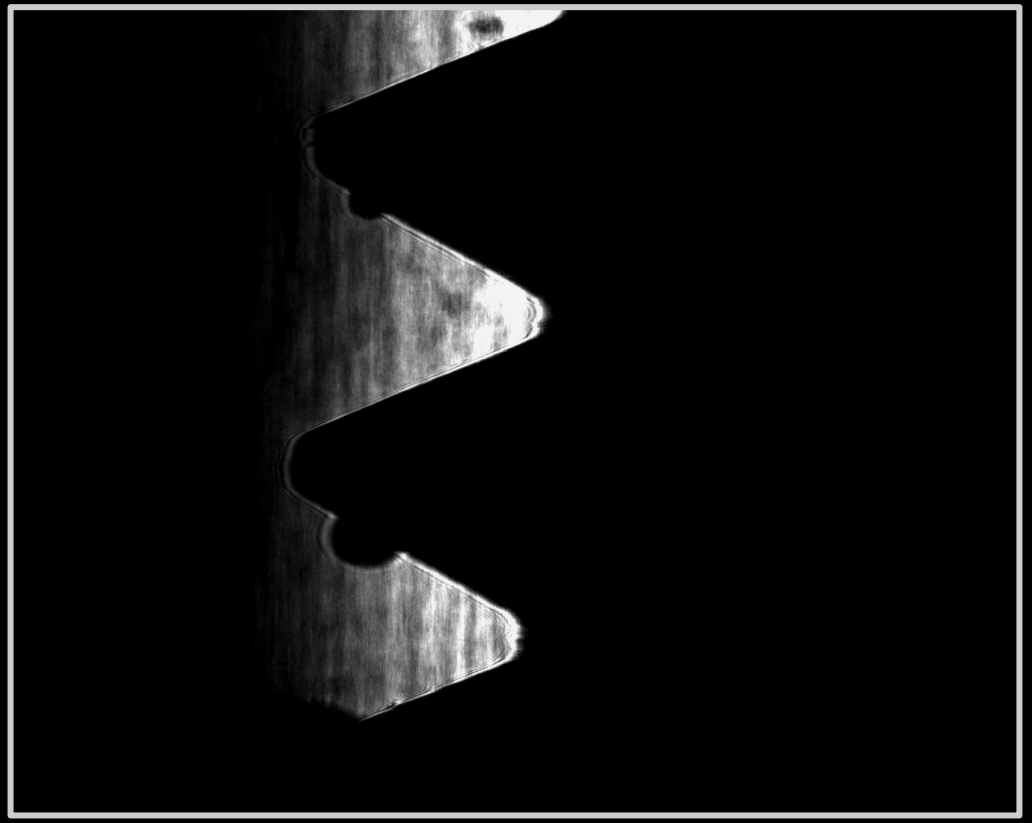
Scanning (milk + water - low concentration)

Global



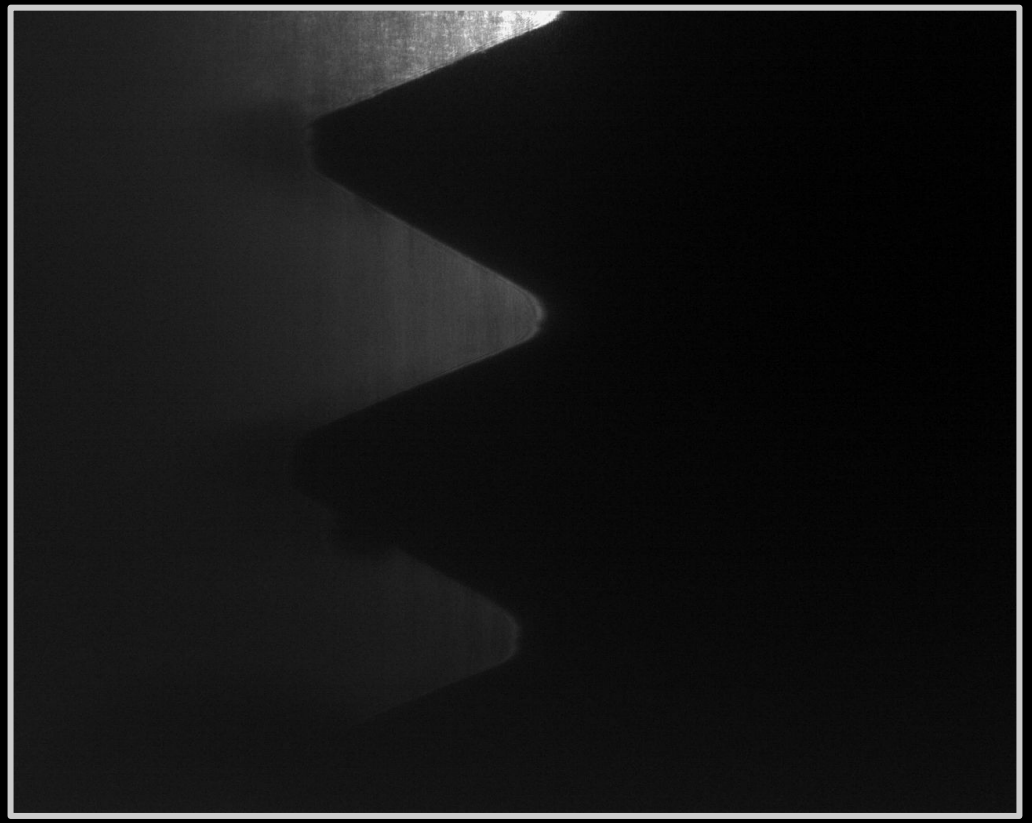
Scanning (milk + water - low concentration)

Epipolar



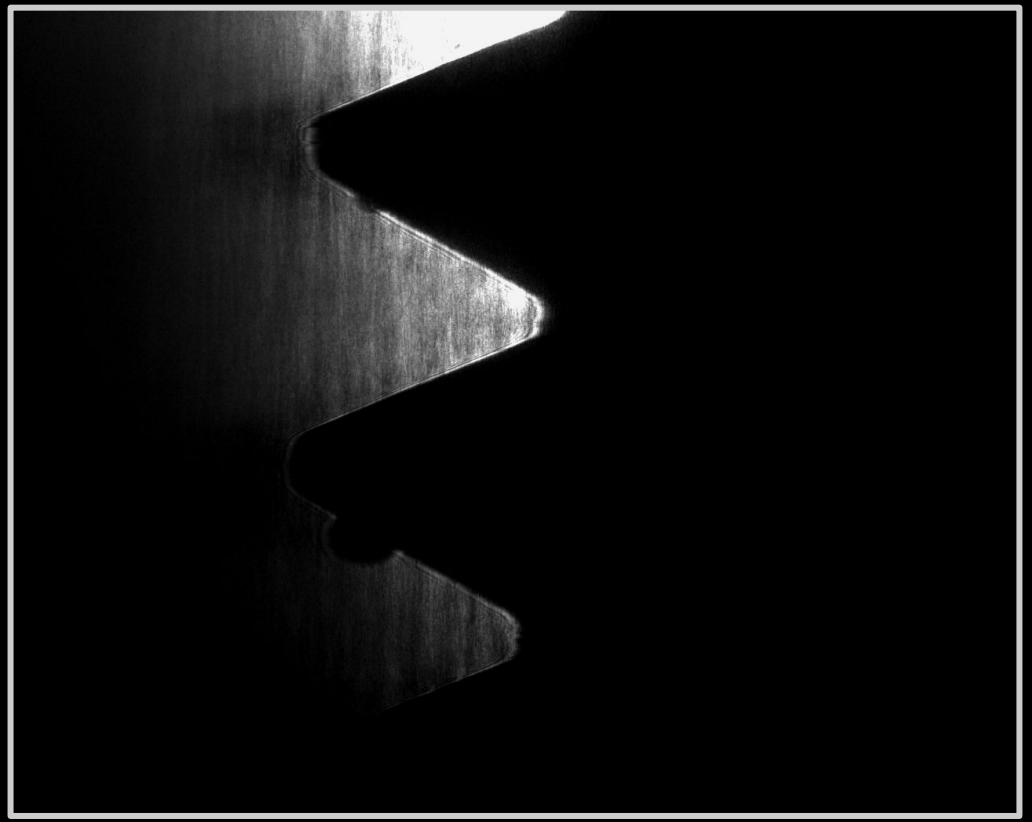
Scanning (milk + water - medium concentration)

Global



Scanning (milk + water - medium concentration)

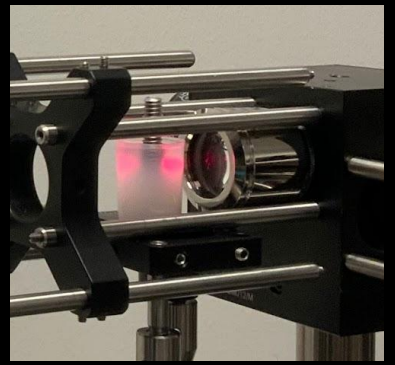
Epipolar



Scanning (milk + water – high concentration)

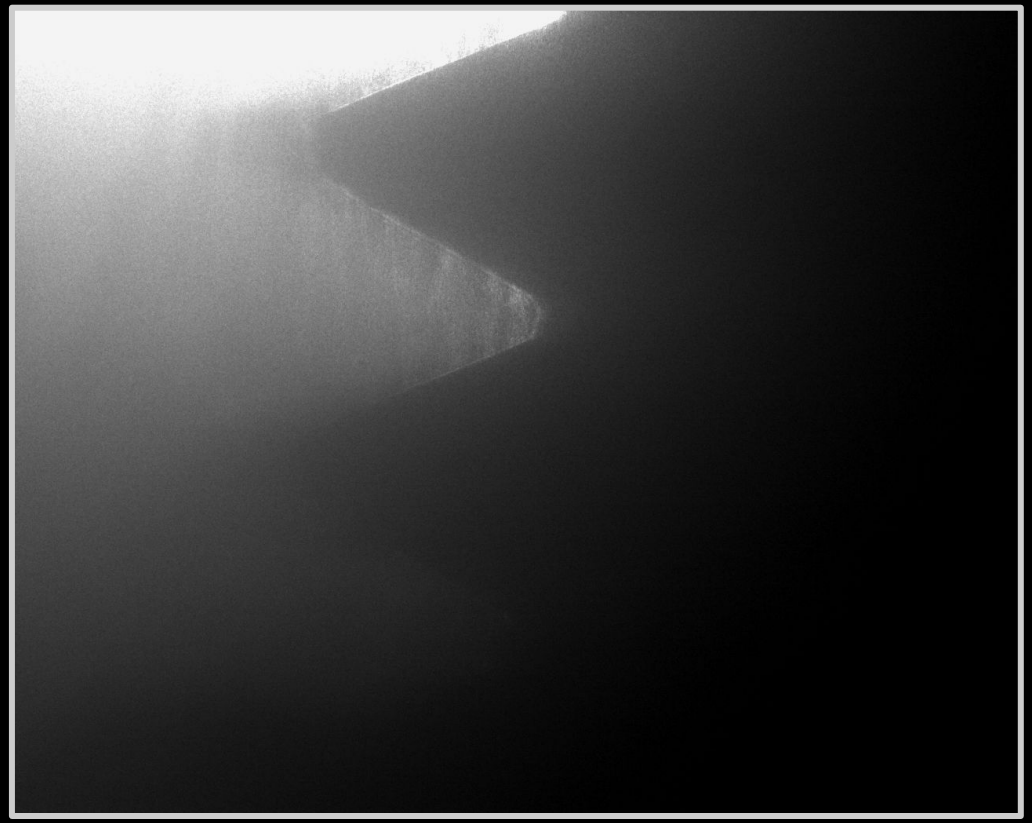
Global





Scanning (milk + water – high concentration)

Epipolar



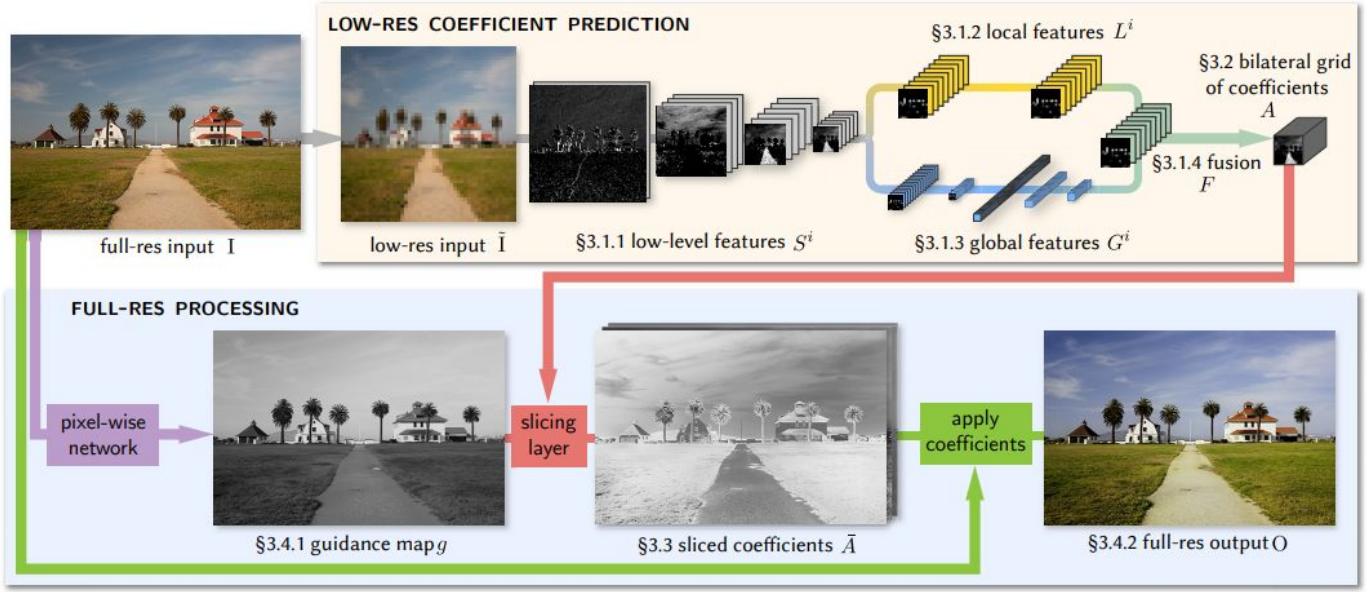
Thank you

Questions?

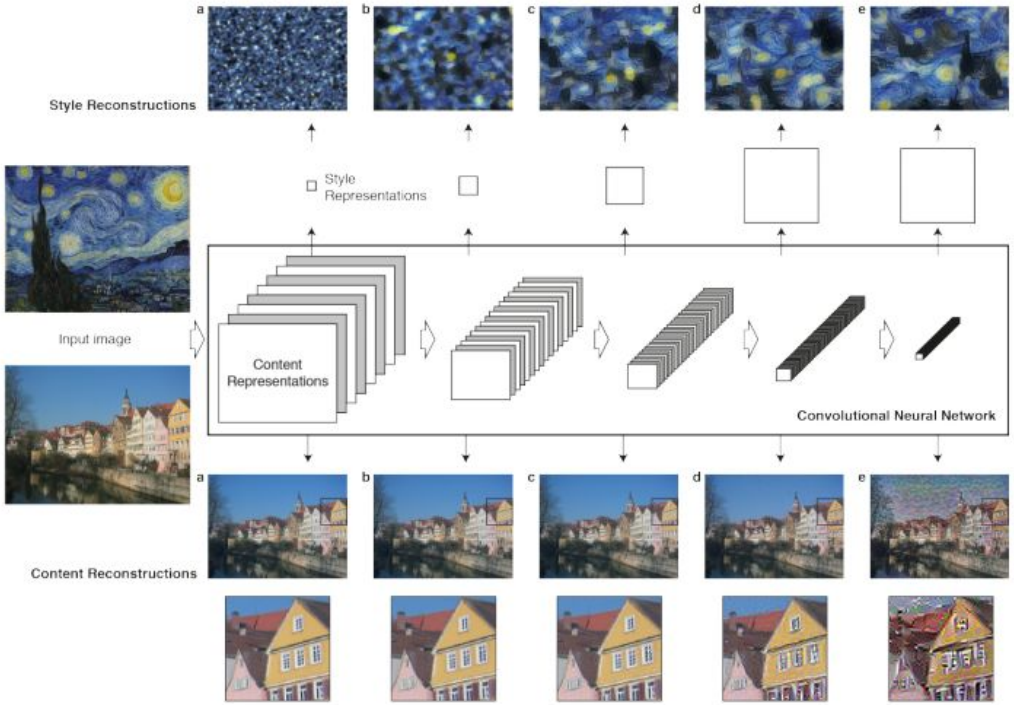
HDRnet and Artistic Style Transfer for HDR replication

Brandon Hung

HDRNet


LOW-RES COEFFICIENT PREDICTION
§3.1.2 local features L^i
§3.2 bilateral grid of coefficients A
§3.1.4 fusion F
full-res input I
low-res input $\tilde{I}$
§3.1.1 low-level features S^i
§3.1.3 global features G^i
FULL-RES PROCESSING
pixel-wise network
slicing layer
apply coefficients
§3.4.1 guidance map g
§3.3 sliced coefficients $\bar{A}$
§3.4.2 full-res output O

Artistic Style Transfer

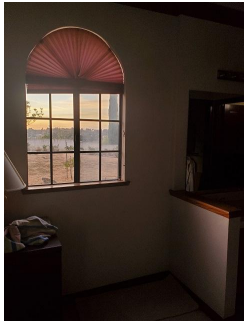


HDRnet for Tonemapping

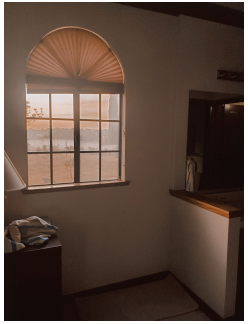
Input

Output

Learned







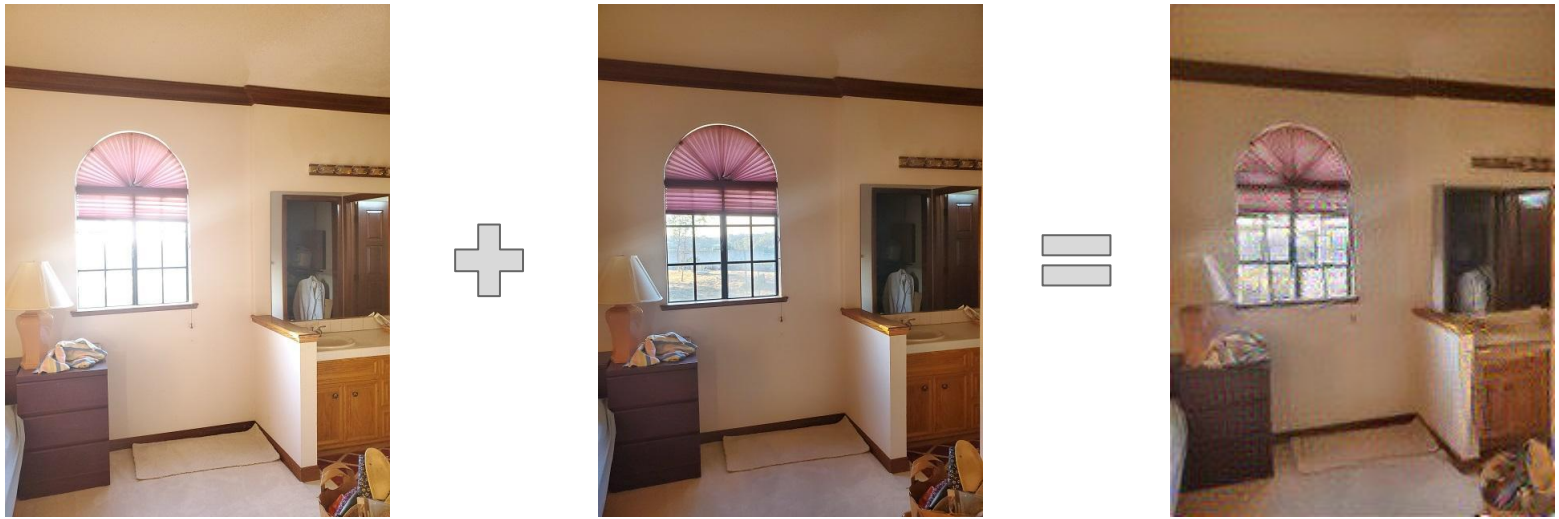






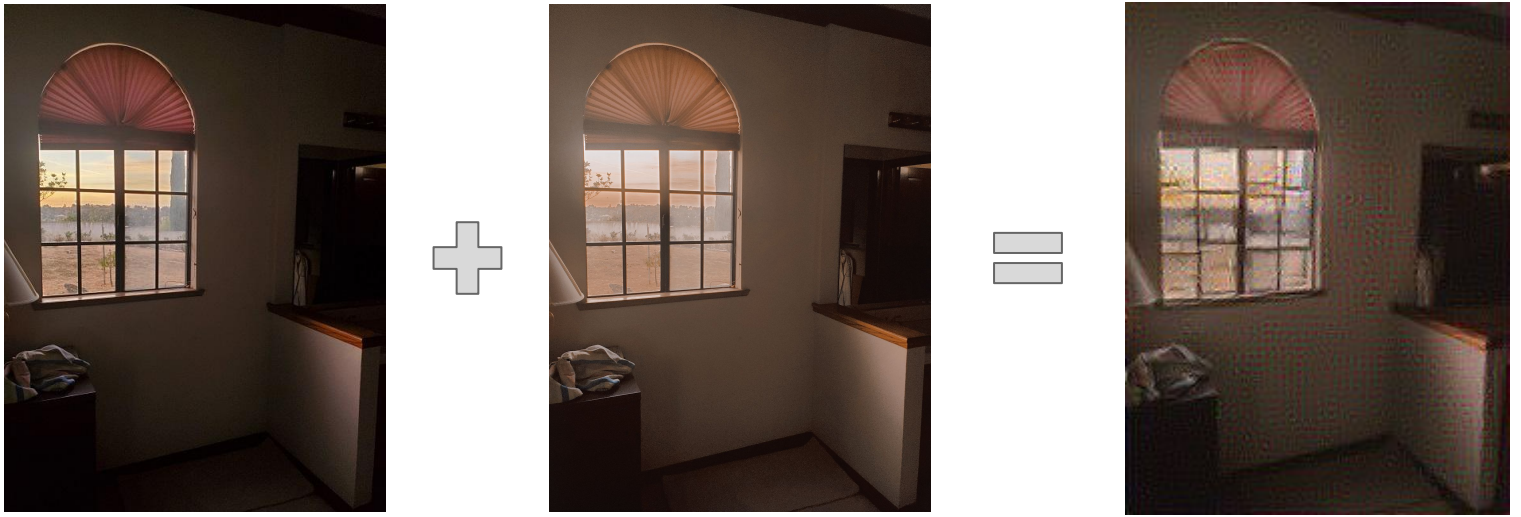
Style Transfer to Preserve Content

- Idea: map bright exposure image on to low-exposure image
- Bright exposure = “style”, low-exposure = “content”



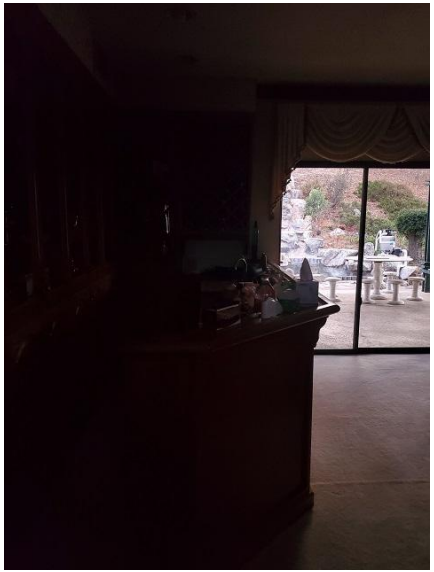
Combining the two?

- Style transfer maps colors to image
- HDRnet tonemaps color
- Idea: tonemap image using HDRnet ⟹ transfer tonemap on original image
  - Hopefully preserves tonemap color + original image content



Combining the two?







Akshath Jain and Deepayan Patra

Unstructured Lightfields

Inspiration

Assignment 4 - Planar Unstructured Lightfields

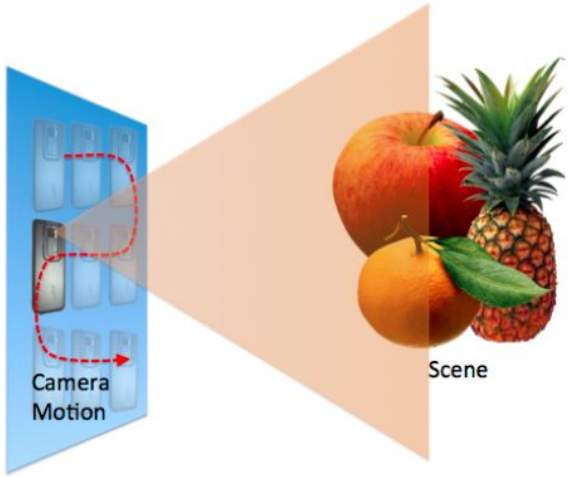


Figure 5: Capturing an unstructured lightfield.

Technical Background

Goal: Generating arbitrary viewpoint rendering given unstructured, non-planar, input

Proposed Solution: Interpolate on a triangular mesh of input viewpoints with appropriate depth information to generate new image

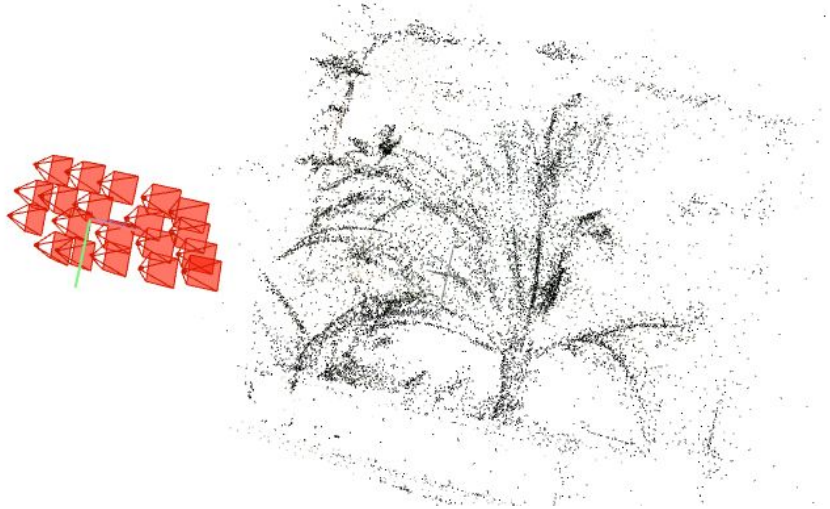
Steps

1. Generate Input Poses
2. Triangulate Viewpoints
3. Rendering

Generating Poses

COLMAP estimates camera poses and a 3D reconstruction of the scene using structure from motion. This is broken down into:

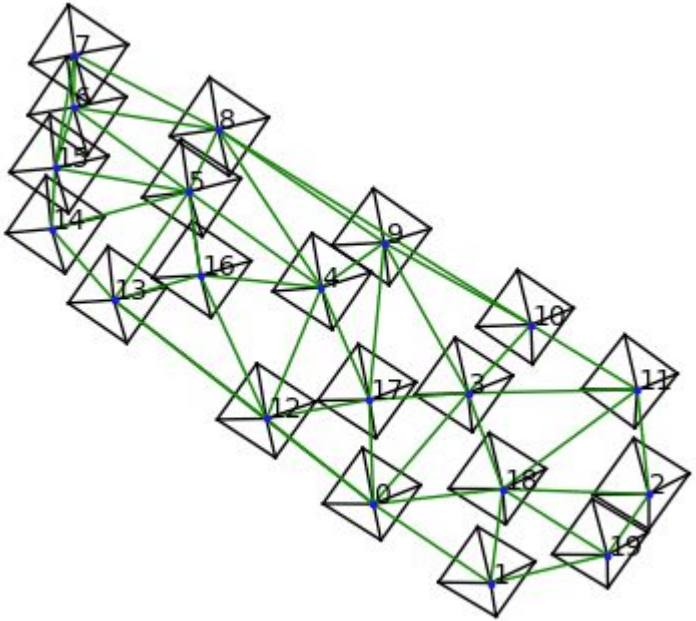
- Feature detection and extraction
- Feature matching and geometric verification
- Structure and motion reconstruction

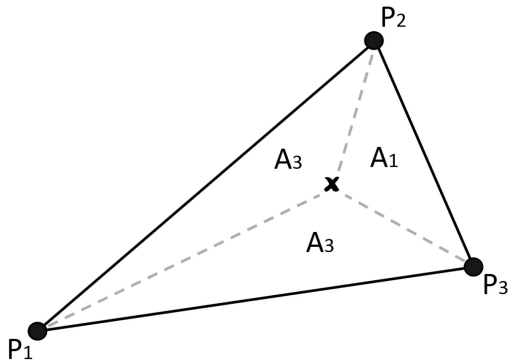


Triangulating Viewpoints

From the new projected camera origins, we develop a Delaunay Triangulation of the viewpoints.

Each point within the input space has exactly three neighboring viewpoints to interpolate.

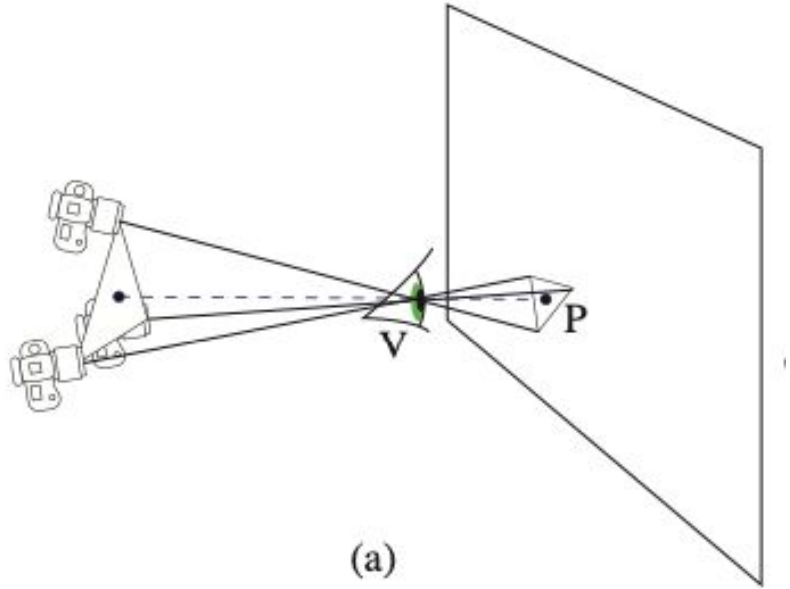




Rendering

Generate each novel viewpoint's image as:

1. Backproject novel pixel onto mesh to determine intersecting face
2. Project novel pixel onto scene
3. Backproject scene pixel onto mesh face vertices
4. Assign novel pixel color using Barycentric interpolation



Results

Unstructured Images



Output





Credits

Our implementation was inspired by the Unstructured Light Fields paper by Davis et. al. We also relied on work done by Mildenhall et. al in LLFF to get camera poses from COLMAP and to identify ideal new sample viewpoints. Our full list of sources is as follows:

Davis et al. “Unstructured Light Fields.” Eurographics 2012.

Mildenhall et. al. “Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines.” SIGGRAPH 2019.

Levoy, Marc. “Light Fields and Computational Imaging.” IEEE 2006.

Buehler et. al. “Unstructured Lumigraph Rendering.” SIGGRAPH 2001.

Isaksen et. al. “Dynamically Reparameterized Light Fields.” SIGGRAPH 2000.

Schönberger, Johannes Lutz and Frahm, Jan-Michael. “Structure-from-Motion Revisited.” CVPR 2016.

Pollefeys, Marc. “Visual 3D Modeling from Images” 2002.

Carnegie Mellon University

15663: Computational Photography
Fall 2020

Using Spatio-Temporal Radiance Variations to Look Around Corners

Varun Jain
varunjai@andrew.cmu.edu

Carnegie Mellon University

Idea[1]

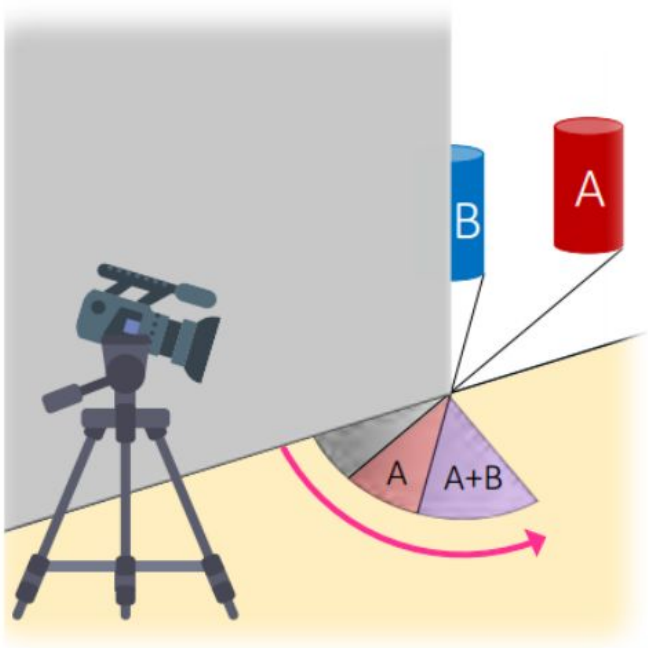


Figure 1a: A and B represent two people hidden from the camera's view by a wall.

Carnegie Mellon University

Idea[1]

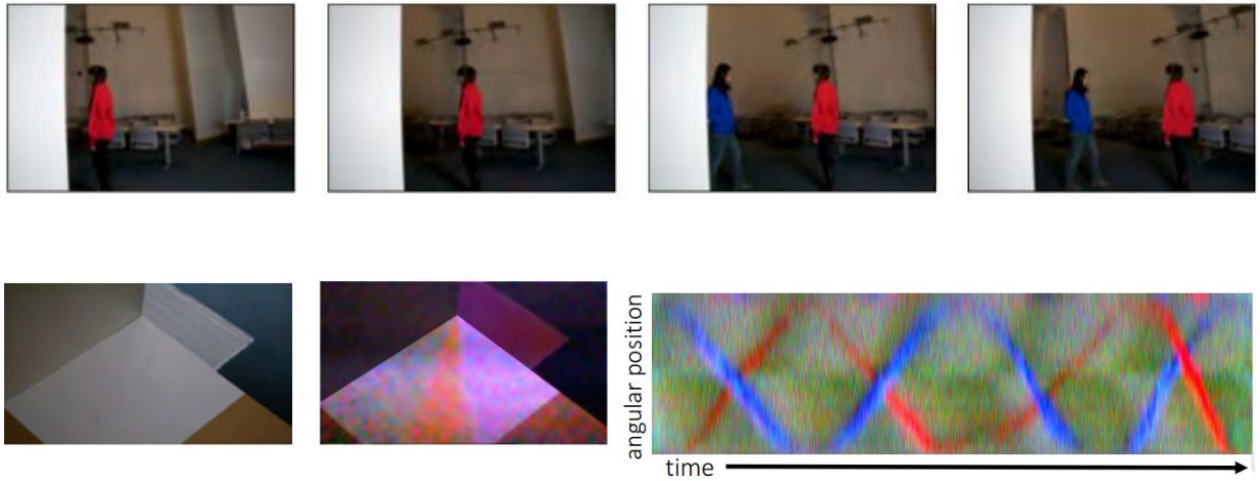
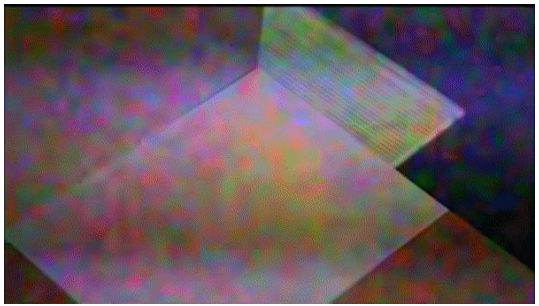


Figure 2: (top) the hidden scene with 2 actors, (bottom) penumbra as seen by the naked eye, as seen by the camera and the groundtruth trajectories over time.





Carnegie Mellon University

Contributions

1. **Re-implementation of the existing code-base in Python**

    a. **Benchmark the implementation by comparing results on the given video sequences**

2. Explore the efficacy of features extracted by deep methods

Carnegie Mellon University

Results [1 person]

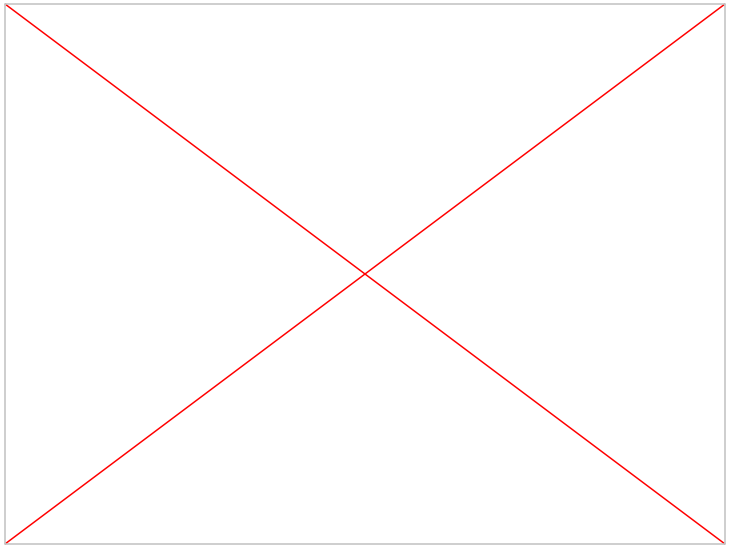


Fig: Observed video

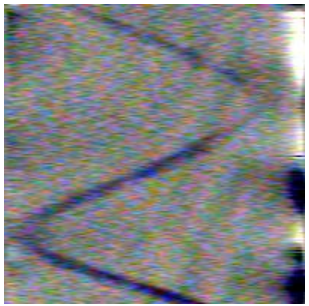


Fig: 1-D angular projections of the hidden scene

Carnegie Mellon University

Results [2 people]

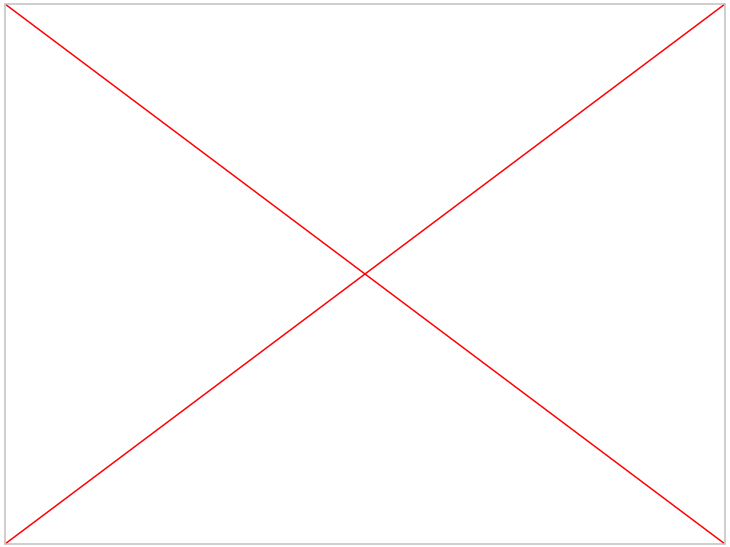


Fig: Observed video

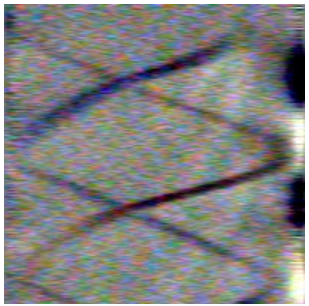


Fig: 1-D angular projections of the hidden scene

Carnegie Mellon University

Comparative Results

Method for Background Subtraction	Error wrt Official Code (MSE)
First Frame	7.41
Mean Frame	**2.94**
Windowed Average Frame	20.16

Carnegie Mellon University

Limitations

1. 1D reconstruction
2. Temporal motion required
3. Assumes uniform floor albedo

Carnegie Mellon University

Contributions

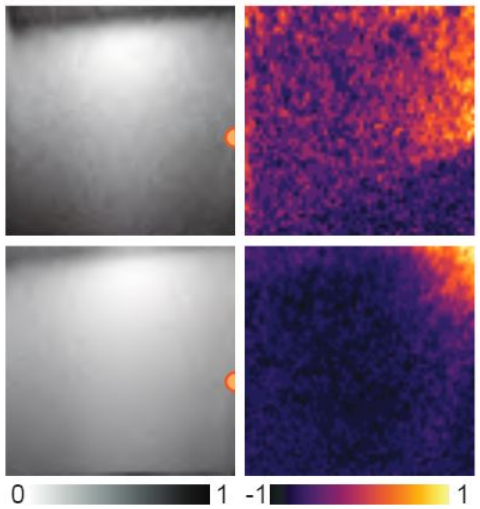
1. Re-implementation of the existing code-base in Python

    a. Benchmark the implementation by comparing results on the given video sequences

2. **Explore the efficacy of features extracted by deep methods**

Carnegie Mellon University

Dataset and Methodology



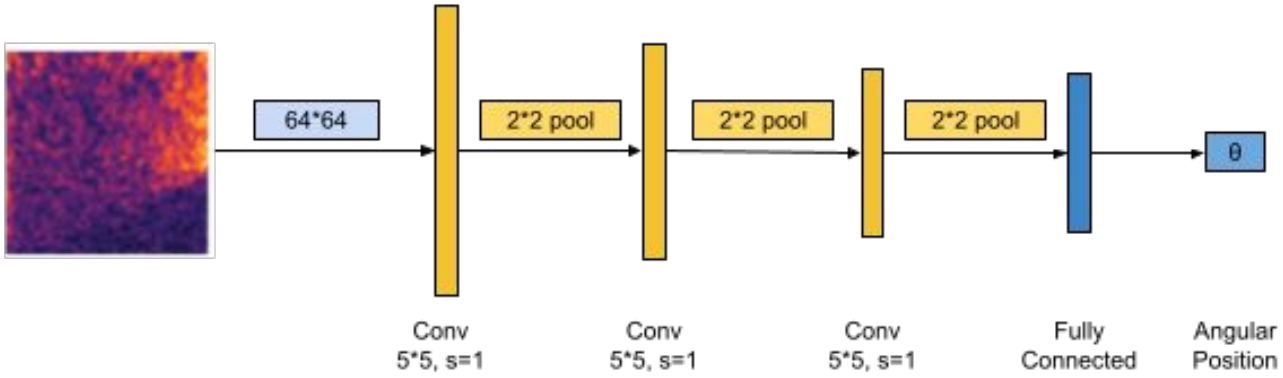


Figure 3: (left) generated dataset, and, (right) model architecture.

Carnegie Mellon University

References

[1] Bouman, Katherine L., Vickie Ye, Adam B. Yedidia, Frédo Durand, Gregory W. Wornell, Antonio Torralba, and William T. Freeman. "Turning corners into cameras: Principles and methods." In Proceedings of the IEEE International Conference on Computer Vision, pp. 2270-2278. 2017.

[2] Seidel, Sheila W., Yanting Ma, John Murray-Bruce, Charles Saunders, William T. Freeman, C. Yu Christopher, and Vivek K. Goyal. "Corner occluder computational periscopy: Estimating a hidden scene from a single photograph." In 2019 IEEE International Conference on Computational Photography (ICCP), pp. 1-9. IEEE, 2019.

[3] Tancik, Matthew, Guy Satat, and Ramesh Raskar. "Flash photography for data-driven hidden scene recovery." arXiv preprint arXiv:1810.11710 (2018).

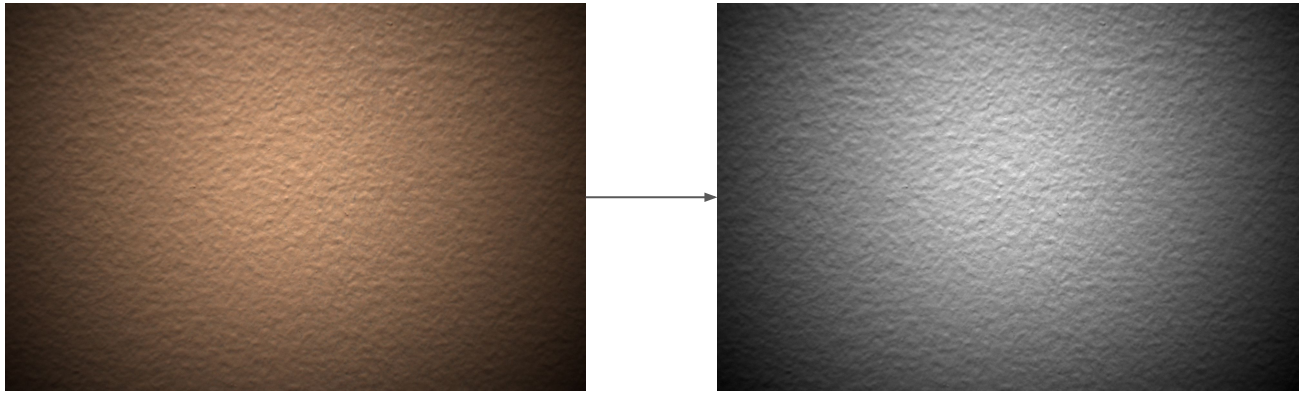
[4] Saunders, Charles, John Murray-Bruce, and Vivek K. Goyal. "Computational periscopy with an ordinary digital camera." Nature 565, no. 7740 (2019): 472-475.

Evaluating Confocal Stereo

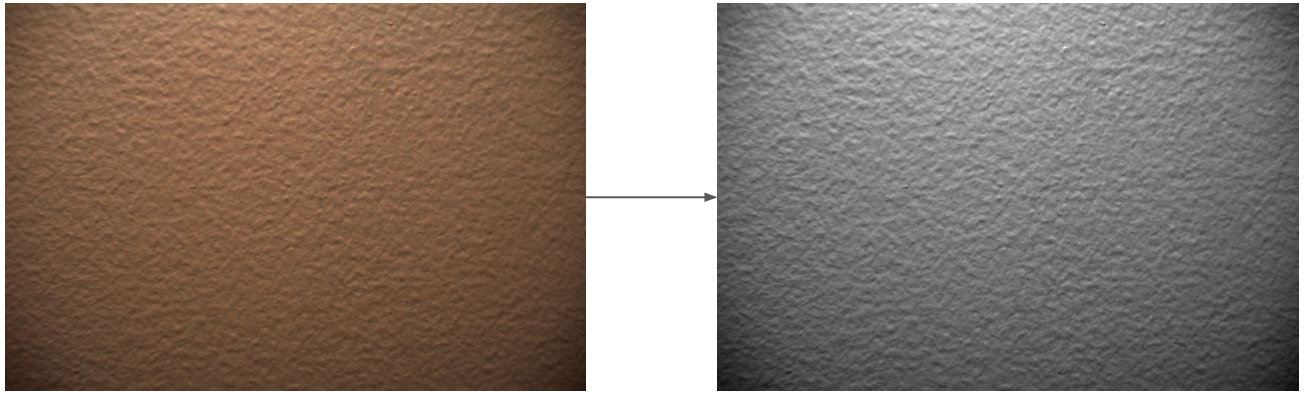
Kyle Jannak-Huang

Relative Exitance Estimation

Aperture f3.5
Focus dist ~ 2cm

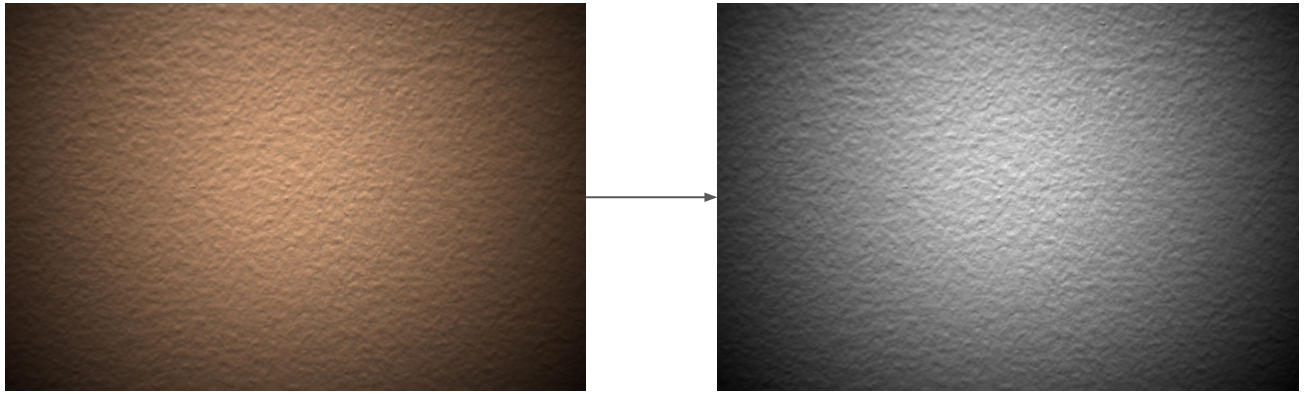


Aperture f7.1
Focus dist ~ 2cm



Relative Exitance Estimation

Aperture f3.5
Focus dist ~ 2cm



Aperture f3.5
Focus dist ~ 90cm

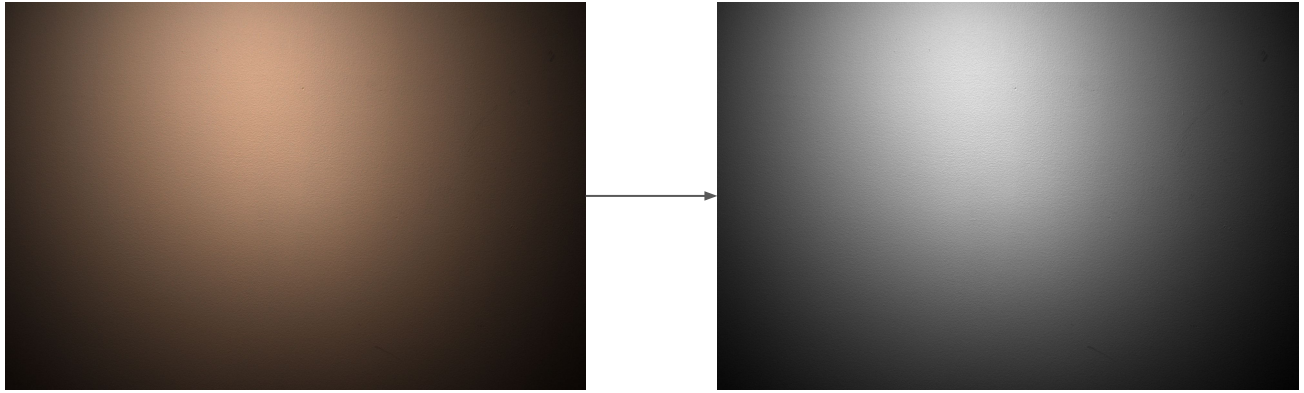
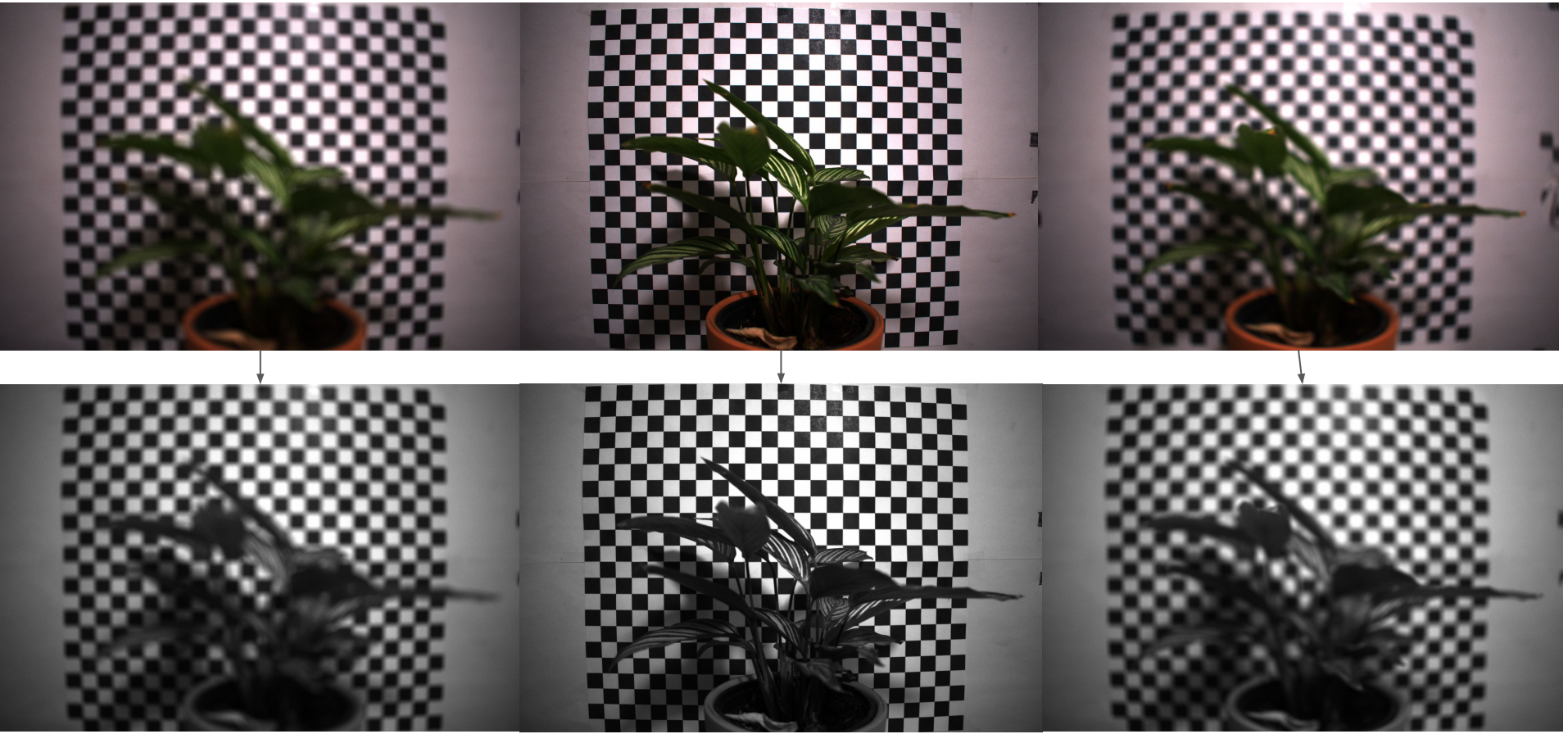
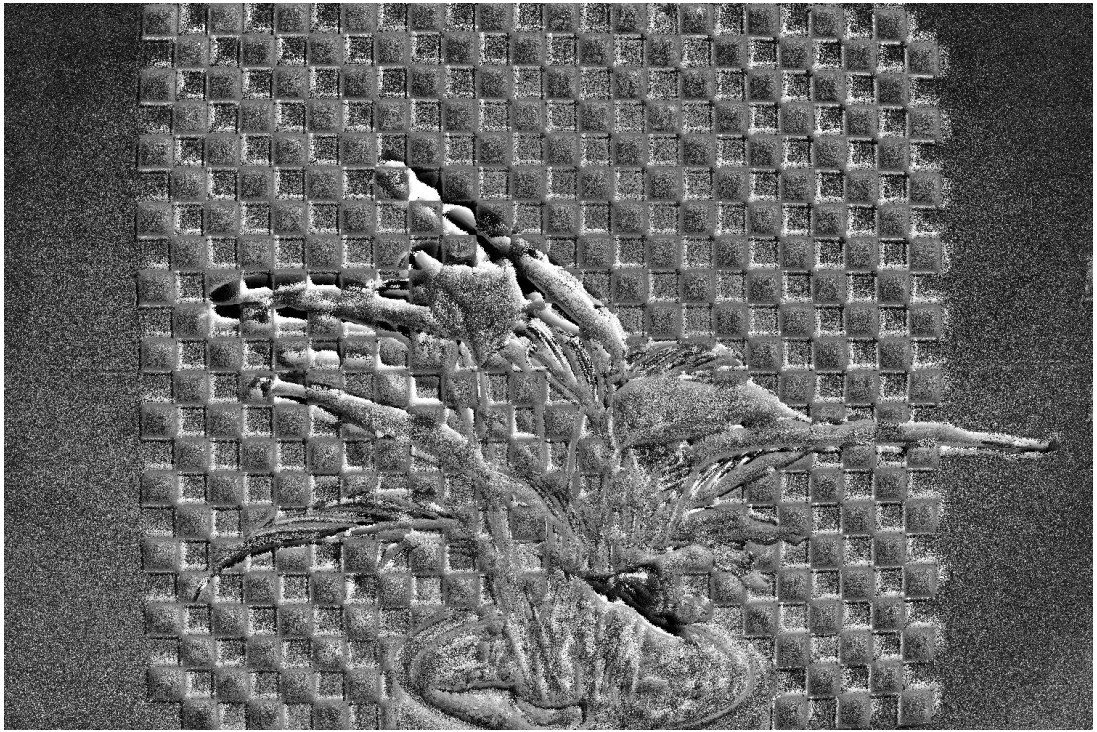


Image Alignment



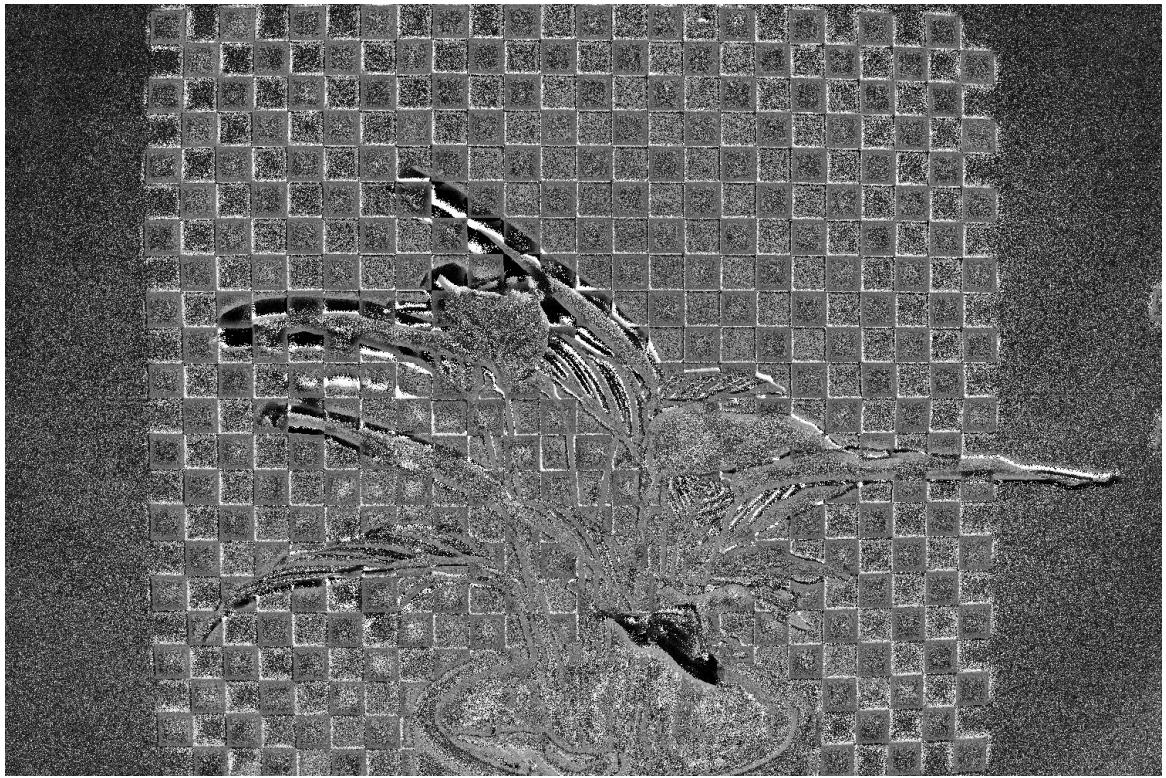
Depthmap from Aperture-Focal Images

- Diagonal line artifacts related to small differences in corner detection during image alignment
- White/black artifacts on leaves that moved due to air flow during capture
- Poor depth estimation on rest of plant due to incorrect depth estimates



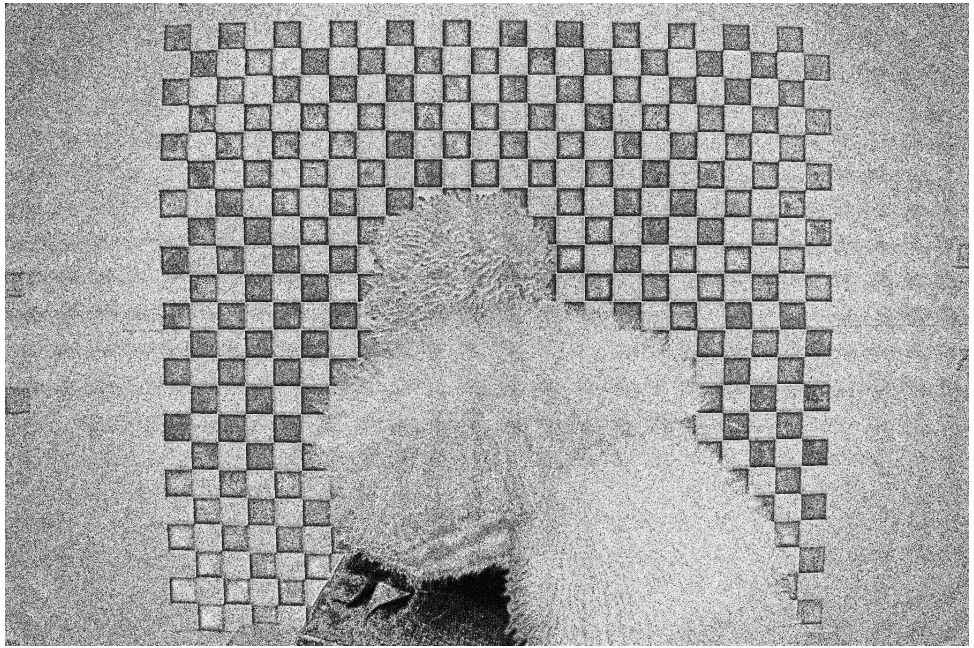
Depthmap from Aperture-Focal Images

- ~~Diagonal line artifacts related to small differences in corner detection during image alignment~~
- White/black artifacts on leaves that moved due to air flow during capture
- Poor depth estimation on rest of plant due to incorrect depth estimates

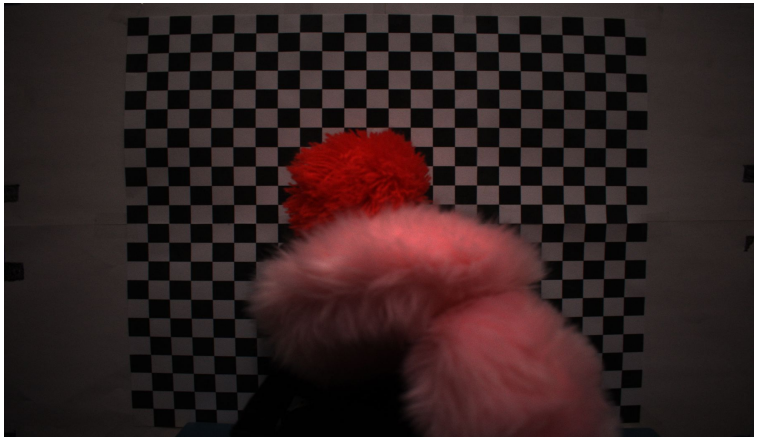


Depthmap from Aperture-Focal Images





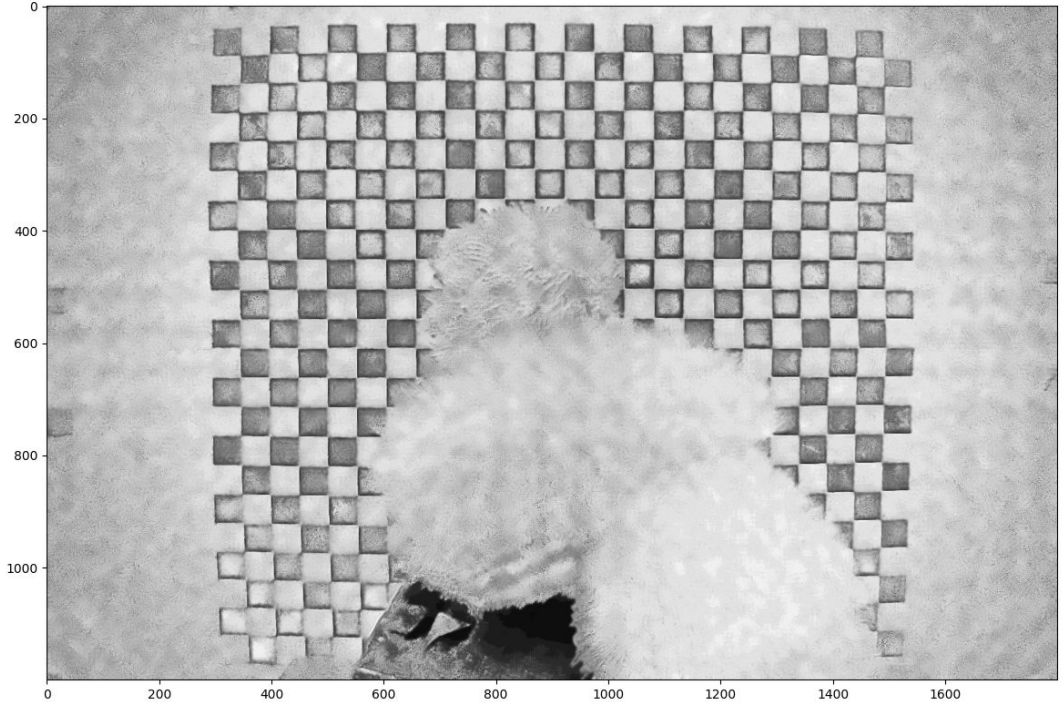
Fuzz dataset images







Denoised result



Discussion

- Benefits from using a custom feature detector that will find features in the center regardless of blur
- Improvements in capture - optical table, diffuse plane, better lighting
- Optimizer for image alignment needs tuning
- Could benefit from segmenting out the background

Gradient-Domain Path Tracing

Zhi Jing (Zoltan), Ran Zhang (Ryan)

Gradient-Domain Path Tracing

Markus Kettunen[1] Marco Manzi[2] Miika Aittala[1] Jaakko Lehtinen[1,3] Frédo Durand[4] Matthias Zwicker[2]

[1]Aalto University [2]University of Bern [3]NVIDIA [4]MIT CSAIL

ACM Transactions on Graphics 34(4) (Proc. SIGGRAPH 2015).




Primal domain
Gradient domain

Gradient-Domain Path Tracing

- Perform standard Monte Carlo rendering to obtain primal image
- Sample gradients
- Reconstruct image from primal and gradients

Gradient-Domain Path Tracing

- Perform standard Monte Carlo rendering to obtain primal image
- Sample gradients
- Reconstruct image from primal and gradients

Standard Monte Carlo rendering

$$f_j(\bar{x}) = \text{(Materials) x (Geometries)}$$
$$\text{x Emitted Lum. x Pixel filtering}$$

$$I_j \approx \frac{1}{N} \sum_{k=1}^{N} \frac{f_j(\bar{x}_k)}{p(\overline{x_k})}$$

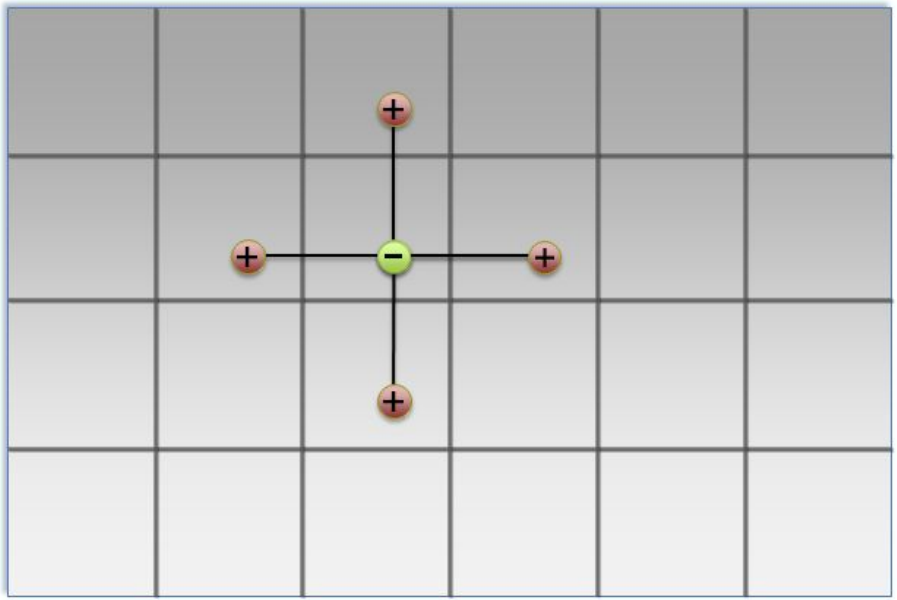
Gradient-Domain Path Tracing

- Perform standard Monte Carlo rendering to obtain primal image
- Sample gradients
- Reconstruct image from primal and gradients

Render result - primal image



Sample gradients



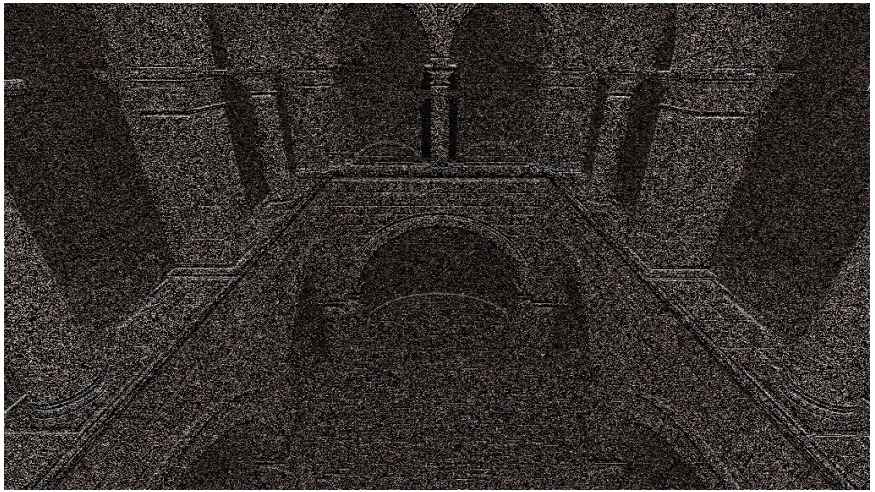
The most naive approach

```
1  function simpleGradientPathTracing()
2      (P, G_x, G_y) = initialize images with zeros
3      for k := 1 to N do
4          foreach pixel sample (x, y) do
5              u = random_numbers()
6              base = radiance(u, x, y)
7              P[x, y] += base
8              G_x[x − 1, y] += 0.5 ∗ (base − radiance(u, x − 1, y))
9              G_y[x, y − 1] += 0.5 ∗ (base − radiance(u, x, y − 1))
10             G_x[x, y] += 0.5 ∗ (radiance(u, x + 1, y) − base)
11             G_y[x, y] += 0.5 ∗ (radiance(u, x, y + 1) − base)
12         end
13     end
14     P /= N, G_x /= N, G_y /= N
15     I = reconstruct(P, G_x, G_y)
16     return I
17 end
```

Gradient image - naive approach



dx



dy

Reconstruction result - naive approach

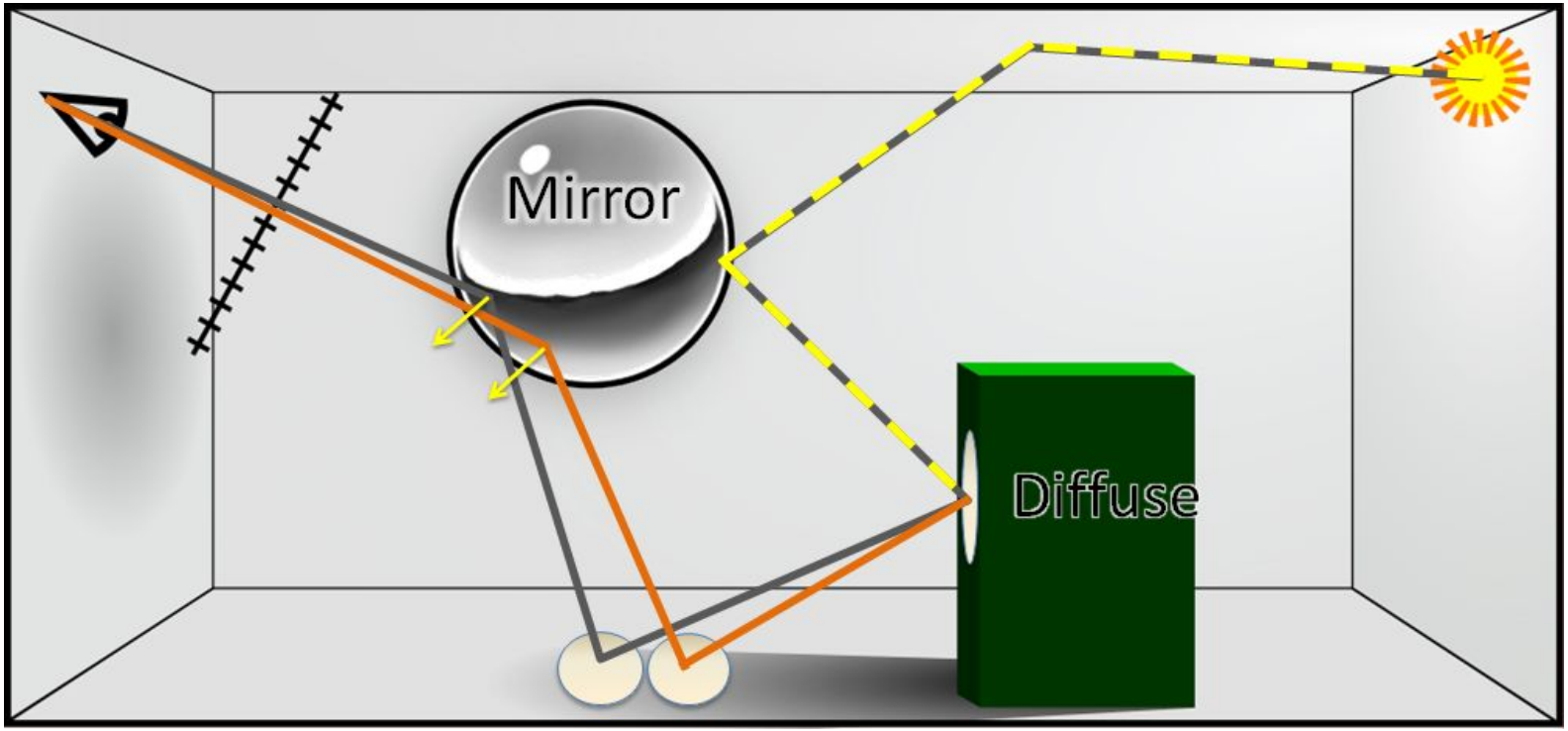


primal

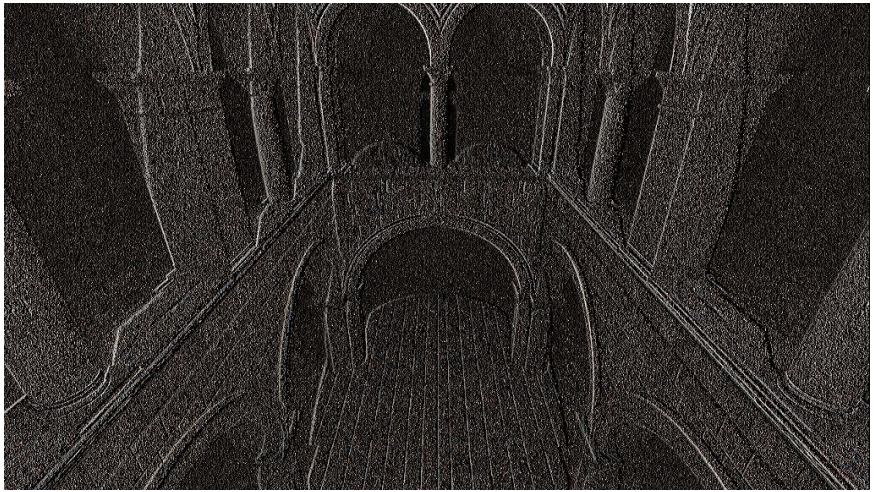


final

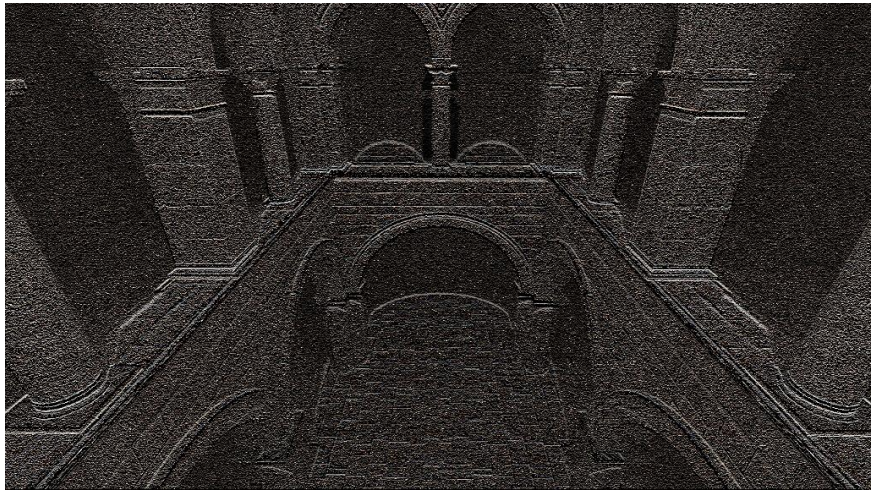
Reconnection



Gradient image - with reconnection



dx

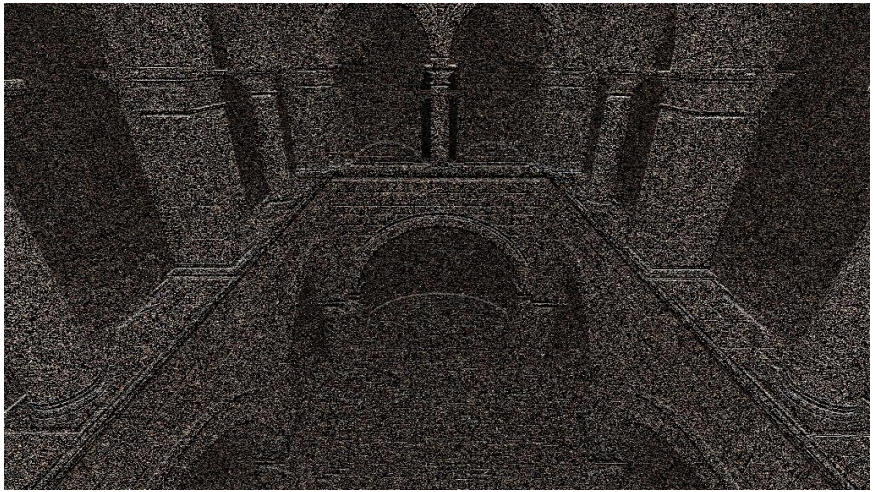


dy

Gradient image - naive approach



dx



dy

Reconstruction result - with reconnection

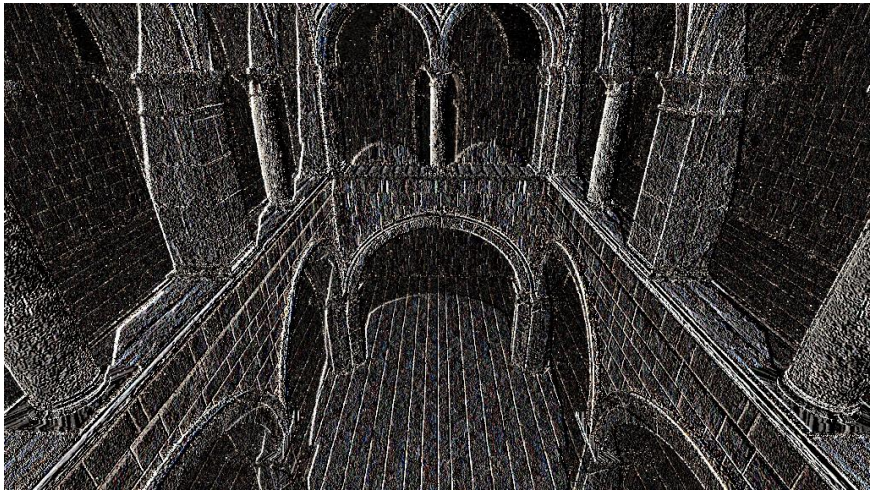


primal

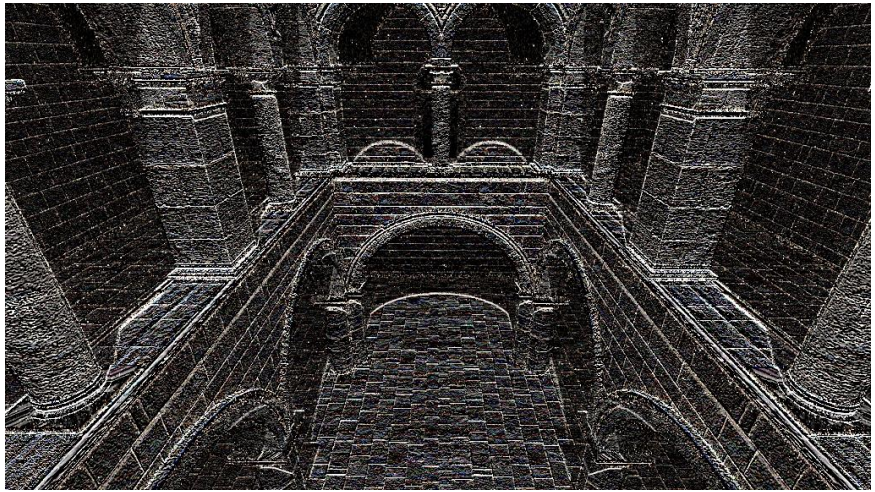


final

Paper’s approach - add MIS



dx

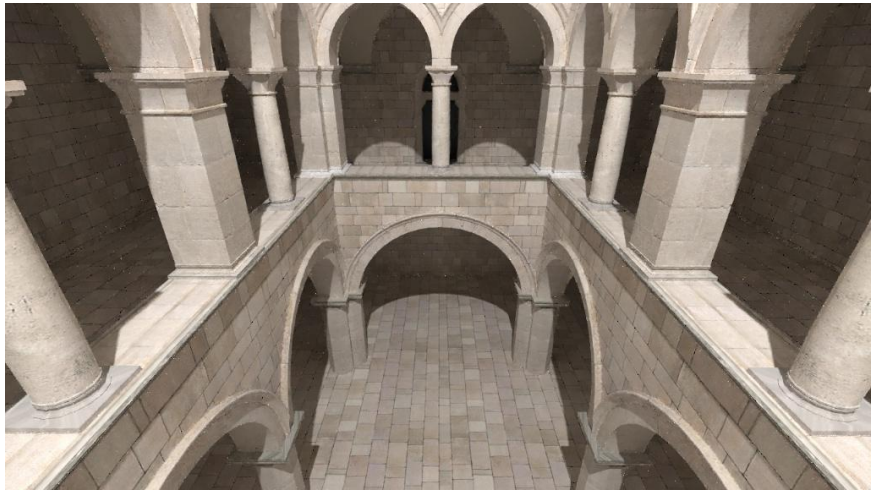


dy

Paper’s approach - add MIS



primal



final

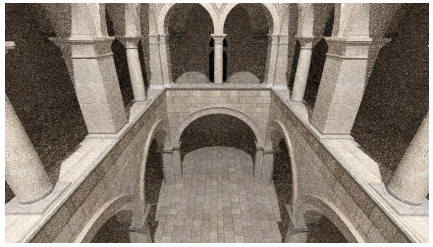
Gradient-Domain Path Tracing

- Perform standard Monte Carlo rendering to obtain primal image
- Sample gradients
- Reconstruct image from primal and gradients

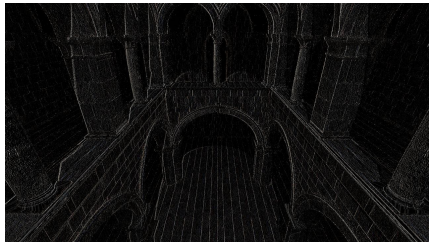
Image Reconstruction

Find an image that best fits the estimated primal and gradients

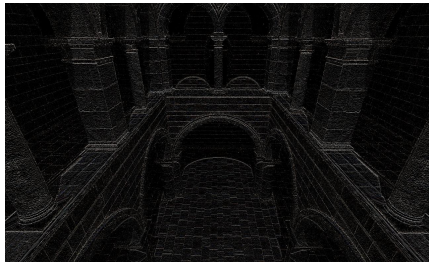
Primal

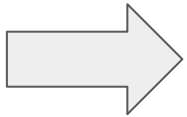


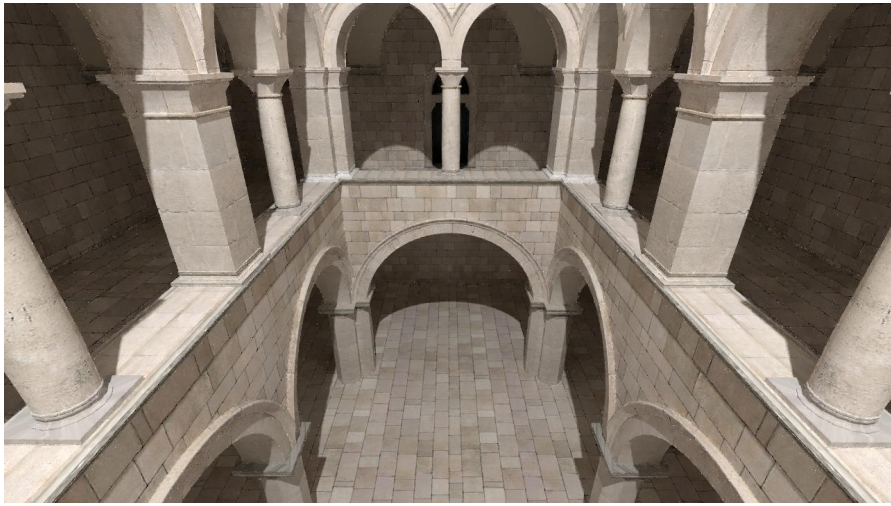
Gx



Gy







Poisson Reconstruction

L2-norm reconstruction

$$I = \operatorname{argmin}_{\mathrm{I}} \left\| \begin{pmatrix} H^{dx} I \\ H^{dy} I \end{pmatrix} - \begin{pmatrix} \boldsymbol{G_x} \\ \boldsymbol{G_y} \end{pmatrix} \right\|_2^2 + \| \alpha (I - \boldsymbol{P}) \|_2^2$$

Gradient term — Primal term

L1-norm reconstruction

$$I = \operatorname{argmin}_{\mathrm{I}} \left\| \begin{pmatrix} H^{dx} I \\ H^{dy} I \end{pmatrix} - \begin{pmatrix} \boldsymbol{G_x} \\ \boldsymbol{G_y} \end{pmatrix} \right\|_1 + \| \alpha (I - \boldsymbol{P}) \|_1$$

Poisson Reconstruction

$$I = \arg\min_{\bar{I}} \left\| \begin{pmatrix} \boldsymbol{\alpha} \\ H^{dx} \\ H^{dy} \end{pmatrix} \bar{I} - \begin{pmatrix} P \\ G_x \\ G_y \end{pmatrix} \right\|$$

A x = b

Solve the least squares minimization problem using Conjugate Gradient Descent

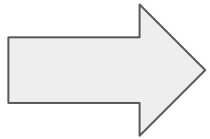
Naïve Implementation using CGD

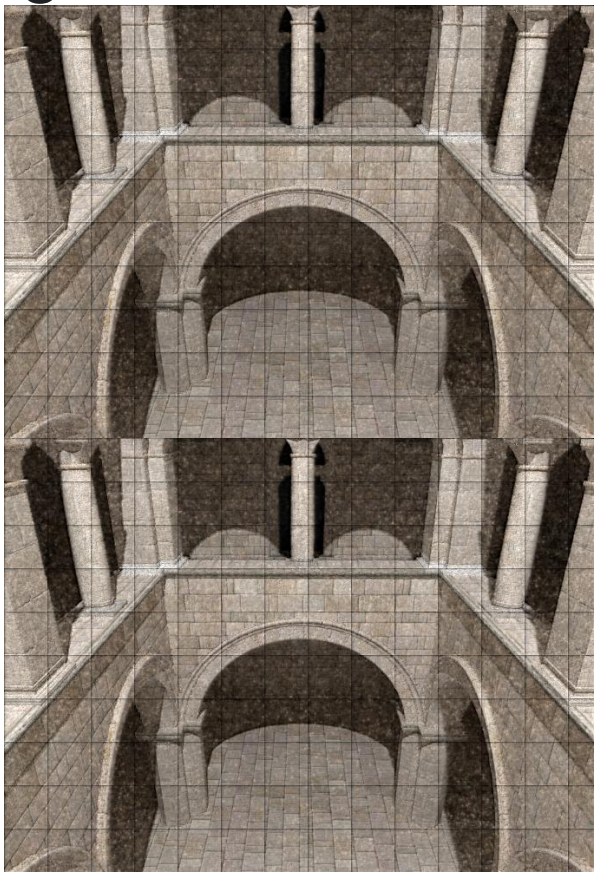
Input: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, initial guess $x_0 \in \mathbb{R}^n$, tolerance $tol \in \mathbb{R}$, and maximum number of iterations $maxit \in \mathbb{R}$.
Output: an approximate solution $x_{sol} \in \mathbb{R}^n$.

$r_0 = b - Ax_0$
$s_0 = p_0 = A^T r_0$
$\gamma_0 = \|s_0\|^2$
for $i = 0, \cdots, maxit$
    $q_i = Ap_i$
    $\alpha_i = \gamma_i / \|q_i\|^2$
    $x_{i+1} = x_i + \alpha_i p_i$
    $r_{i+1} = r_i - \alpha_i q_i$
    if converged within tol, **then** stop.
    $s_{i+1} = A^T r_{i+1}$
    $\gamma_{i+1} = \|s_{i+1}\|^2$
    $\beta_i = \gamma_{i+1} / \gamma_i$
    $p_{i+1} = s_{i+1} + \beta_i p_i$
end
$x_{sol} = x_{i+1}$

Naïve Implementation using CGD



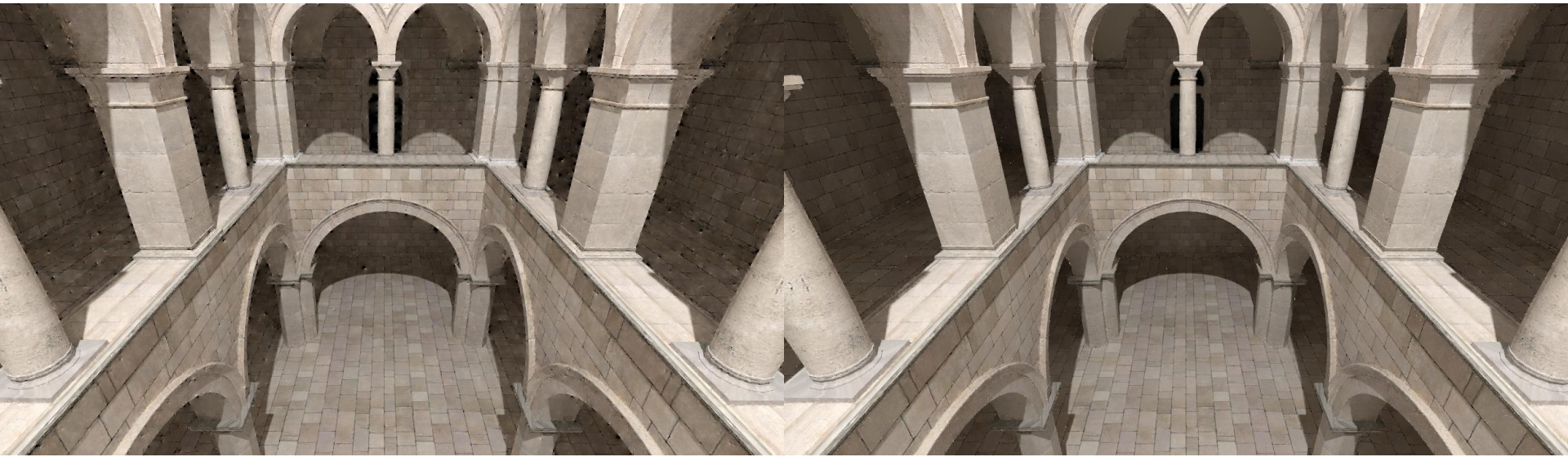




L2 norm

L1 norm

Re-implement and Experiment (L1 norm vs. L2 norm)



L2 Norm

L1 Norm

Experiment (L1 norm vs. L2 norm)



L2 Norm

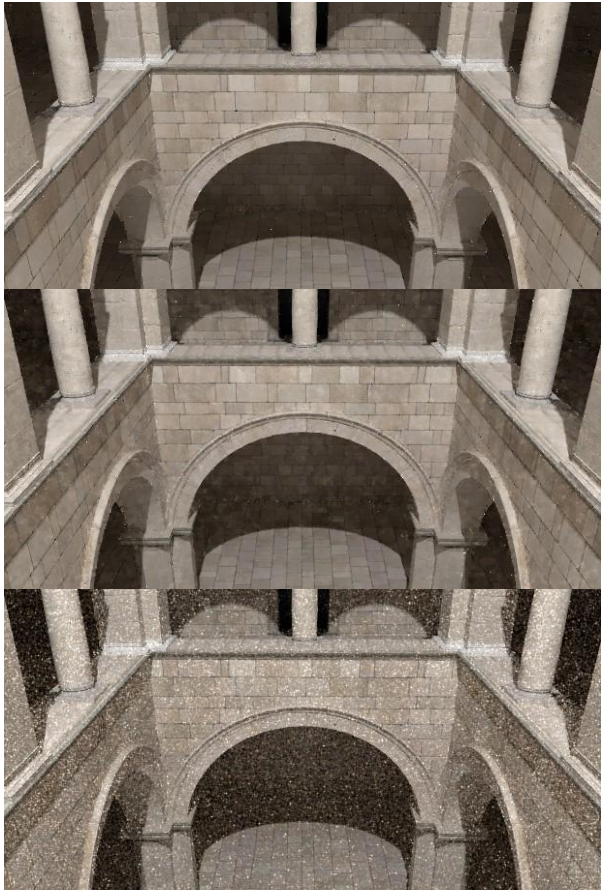
Unbiased but has some artifacts



L1 Norm

Biased but has less artifacts

Experiment (different α values)



α = 0.2

α = 1.0

α = 10



Thank you!

References

Kettunen, Markus, et al. "Gradient-domain path tracing." *ACM Transactions on Graphics (TOG)* 34.4 (2015): 1-13.

Lehtinen, Jaakko, et al. "Gradient-domain metropolis light transport." *ACM Transactions on Graphics (TOG)* 32.4 (2013): 1-12.

Manzi, Marco, et al. "Gradient-Domain Bidirectional Path Tracing." *EGSR (EI&I)*. 2015.

Pérez, Patrick, Michel Gangnet, and Andrew Blake. "Poisson image editing." *ACM SIGGRAPH 2003 Papers*. 2003. 313-318.

Manzi, Marco, Delio Vicini, and Matthias Zwicker. "Regularizing Image Reconstruction for Gradient‐Domain Rendering with Feature Patches." *Computer graphics forum*. Vol. 35. No. 2. 2016.

Ji, Hao, and Yaohang Li. "Block conjugate gradient algorithms for least squares problems." *Journal of Computational and Applied Mathematics* 317 (2017): 203-217.

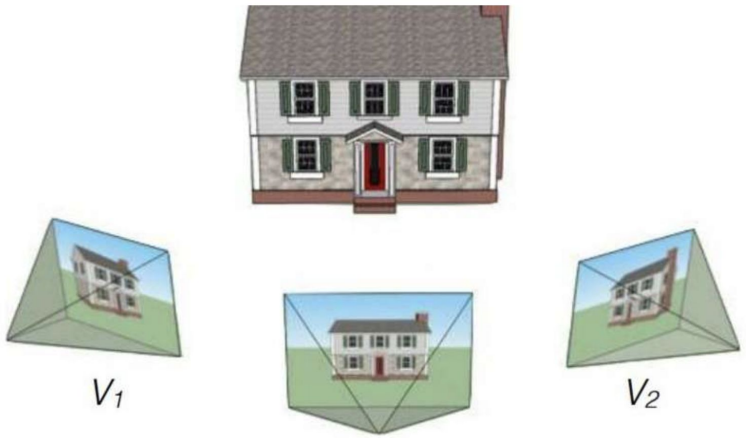
View Synthesis using Neural Radiance Fields

Rohan Joshi

Problem Definition:

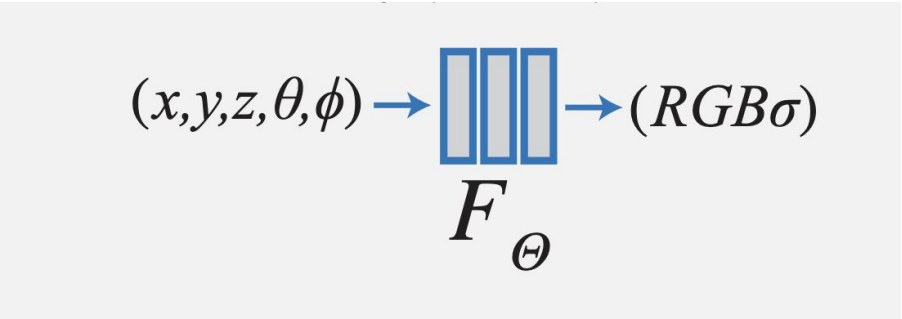
Given: Images of a scene from known camera poses,

Task: Render new views of the scene



Method: Scene Representation as Neural Radiance Fields

- Use a fully connected network instead of a voxel grid (Advantages: Simplicity and efficient storage)



- Calculate by querying points along rays

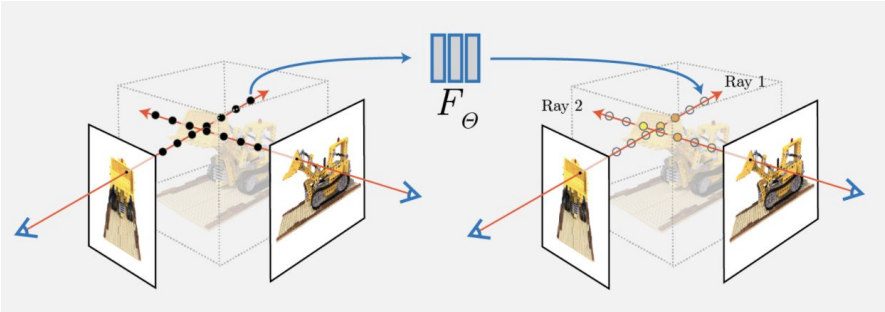
Algorithm for Volume Rendering

Forward Pass

Step 1: Generate Camera Rays

Step 2: Sample points along each ray (affects the resolution of output image)

Step 3: Get RGB and σ (Volume Density at the point) value



Algorithm for Volume Rendering

Error Calculation and Backpropagation

Step 1: Composite the pixel color along the ray to get the rgb image

Step 2: Minimise the mse for known images

Step 3: Backprop ...

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} \mathrm{W}(\sigma_i) \mathbf{c}_i$$

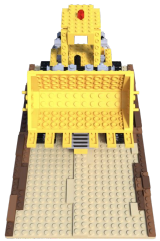
$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i (1 - \exp(-\sigma_i \delta_i)) \mathbf{c}_i$$

$$T_i = \exp\left(- \sum_{j=1}^{i-1} \sigma_j \delta_j \right)$$

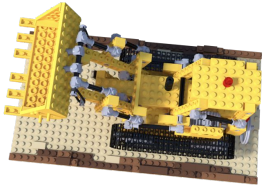
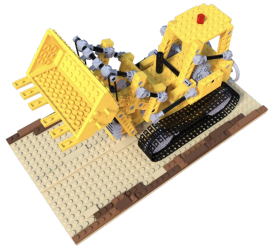
$$\delta_i = t_{i+1} - t_i$$

Results: Performed on Synthetic Data

- Training Images:
  - 100 images taken from different views
  - Camera Poses provided in the dataset

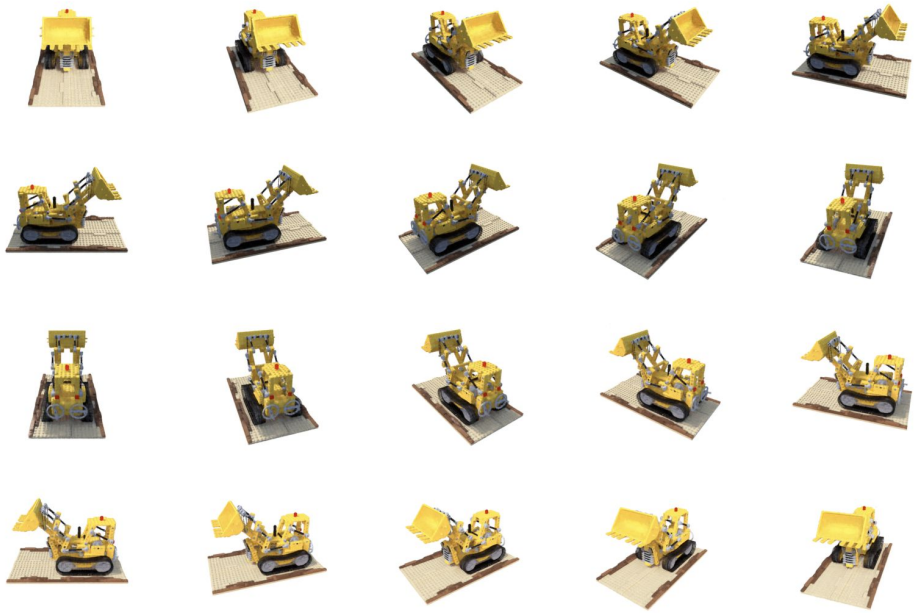




Training Set

Result: Rendered Views



References

- **DeepStereo: Google quartet has method for new-view synthesis-** Credit: John Flynn et al. arXiv:1506.06825
- Nerf: https://arxiv.org/pdf/2003.08934.pdf
- https://www.matthewtancik.com/nerf

Thank you

Color-Filtered Aperture For Image Depth & Segmentation

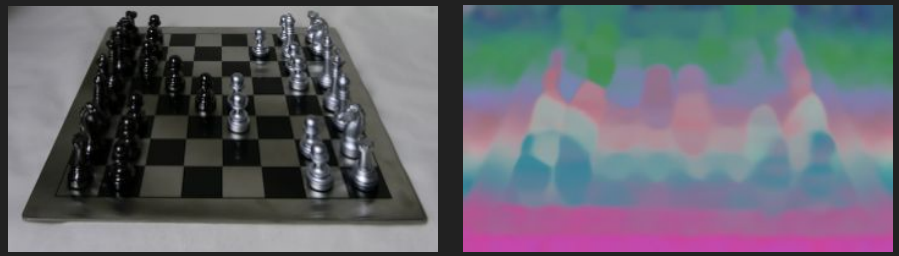
Leron Julian

Depth Estimation Techniques

Stereo



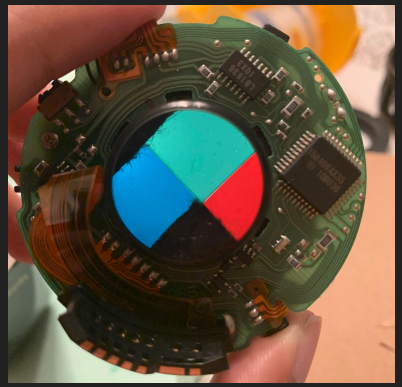
Depth From Defocus



Depth From Disparity Using Color Filter (This Project)

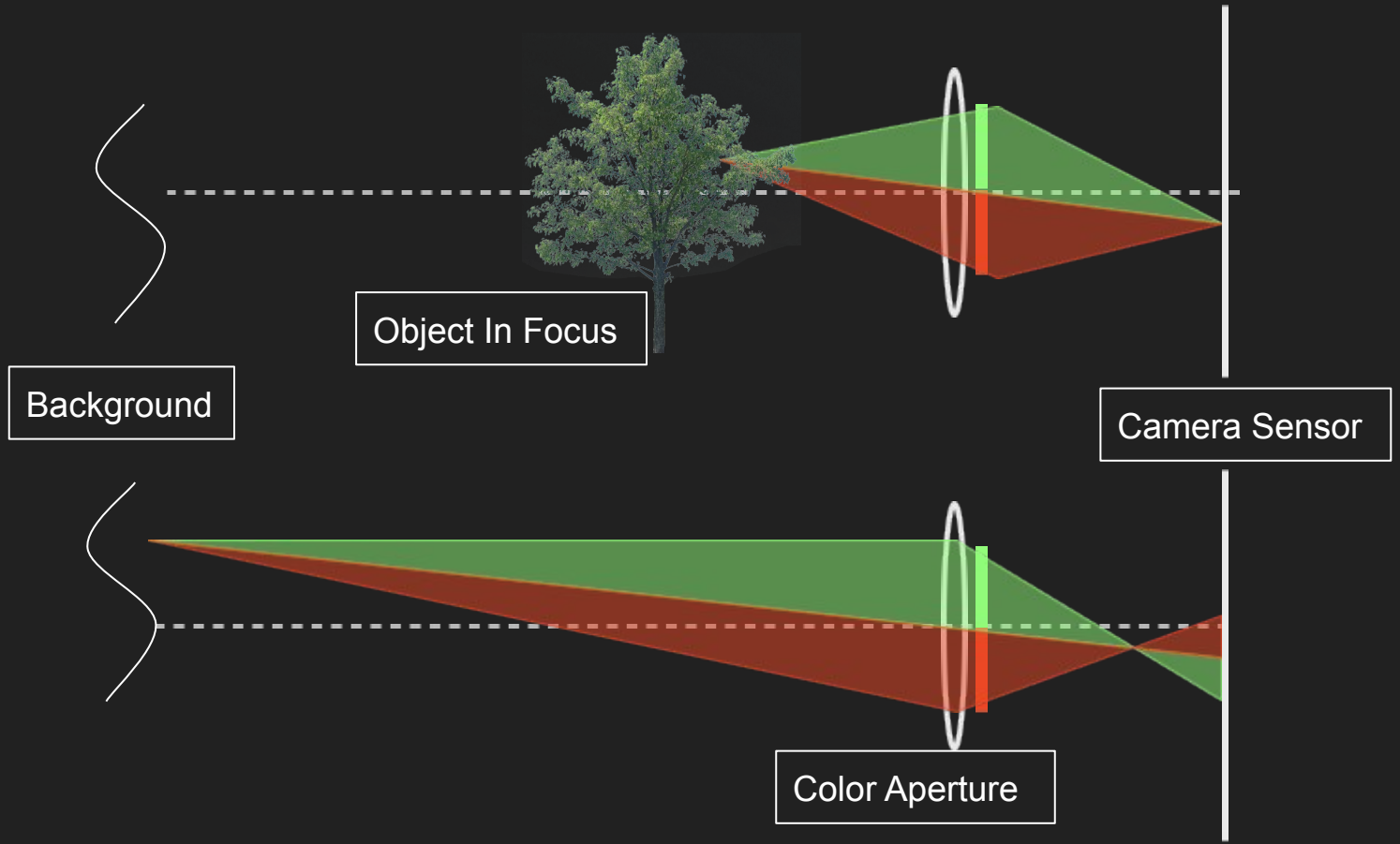
Constructing Depth Using Color Misalignment

- Red, Green, Blue Color Aperture placed in a certain arrangement
- A scene point farther than focused depth =
  - Right-Shift in R
  - Up-Shift in G
  - Left-Shift in B
- Color shifts come from geometric shifts.





How? (Red & Green Example)



Sample Images





Depth Estimation

- Let d be the hypothesized disparity between the RGB Channels:
- Need to measure the quality of between:

$$I_r(x+d,y), I_G(x,y-d), I_B(x-d,y)$$

- Colors in natural images form elongated clusters in RGB space

Depth Estimation

- Consider a set of pixels with hypothesized disparity d. Set can be represented as:

$$S_I(x, y; d) = \{I_r(x + d, y), I_G(x, y - d), I_B(x - d, y) | \ (s, t) \in w(x, y)\}$$

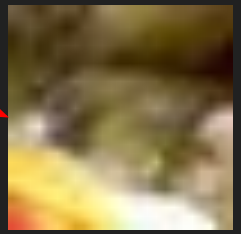
- Goal is to find a disparity d that minimizes the color alignment measure

$$L(x, y; d) = \frac{\lambda_0 \lambda_1 \lambda_2}{\sigma_r^2 \sigma_g^2 \sigma_b^2}$$

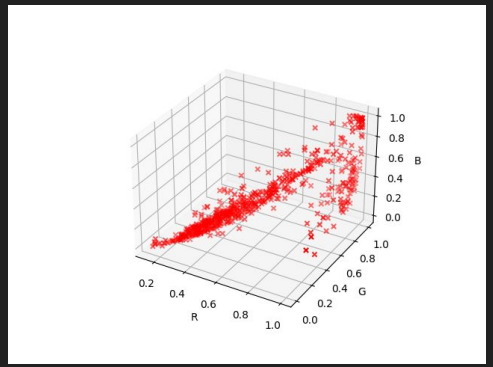
- As L gets small when cluster is elongated meaning the RGB components are correlated

RGB Color Model

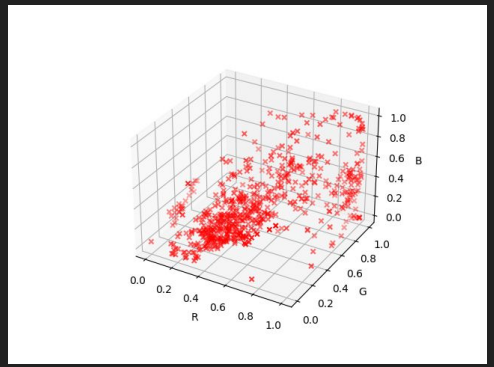




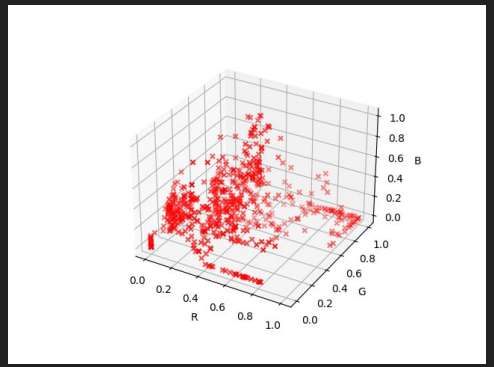
- As the disparity d increases, the cluster becomes more isotropic.



d = 1



d = 3



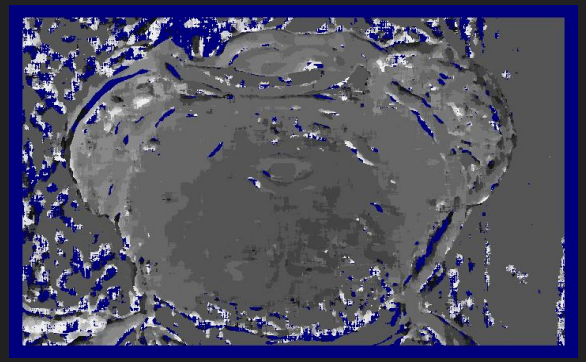
d = 5

Depth Map

- Can now solve for the disparity d to by minimizing L for a predetermined range of disparities.
- Can construct a depth map using local estimates (prone to error) or energy minimization graph-cuts:



Captured Images



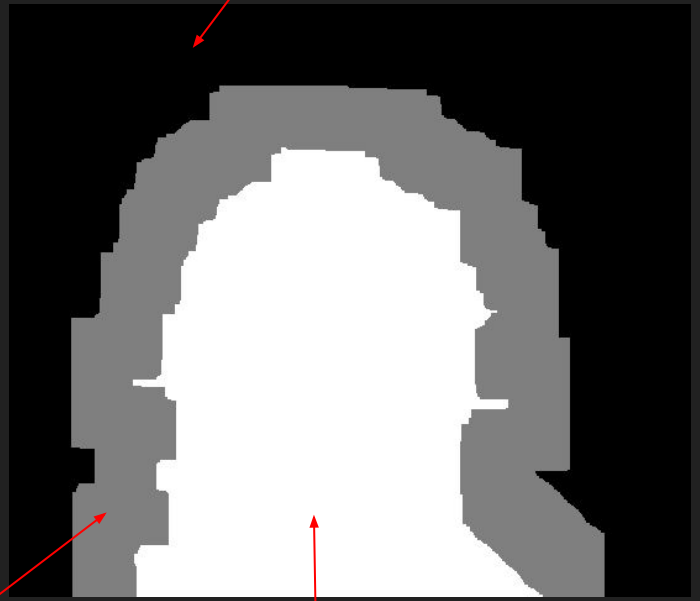
Local Estimates



Estimate with Graph-Cuts

Trimap





Matting

- Matting Equation: $I(x,y) = \alpha(x,y)F(x,y) + (1 - \alpha(x,y))B(x,y)$





Matte Optimization Flow

- Optimize the matte with various consistency methods based on foreground and background color
- Obtain the optimal matte





Applications of Matting

- Changing the background of images
- Refocusing
- Artificial Background Blur
- Video Matting

References

- Bando et. al, “Extracting Depth and Matte using a Color-filtered Aperture”. SIGGRAPH 2008.

- Levin et. al, “A closed form solution to natural image matting”. CVPR 2006

Depth from Defocus in the Wild

Alice Lai, Adriana Martinez

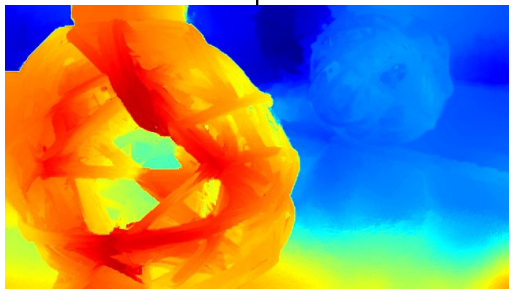
Goal: Two-frame depth from defocus using tiny blur condition

Original Image

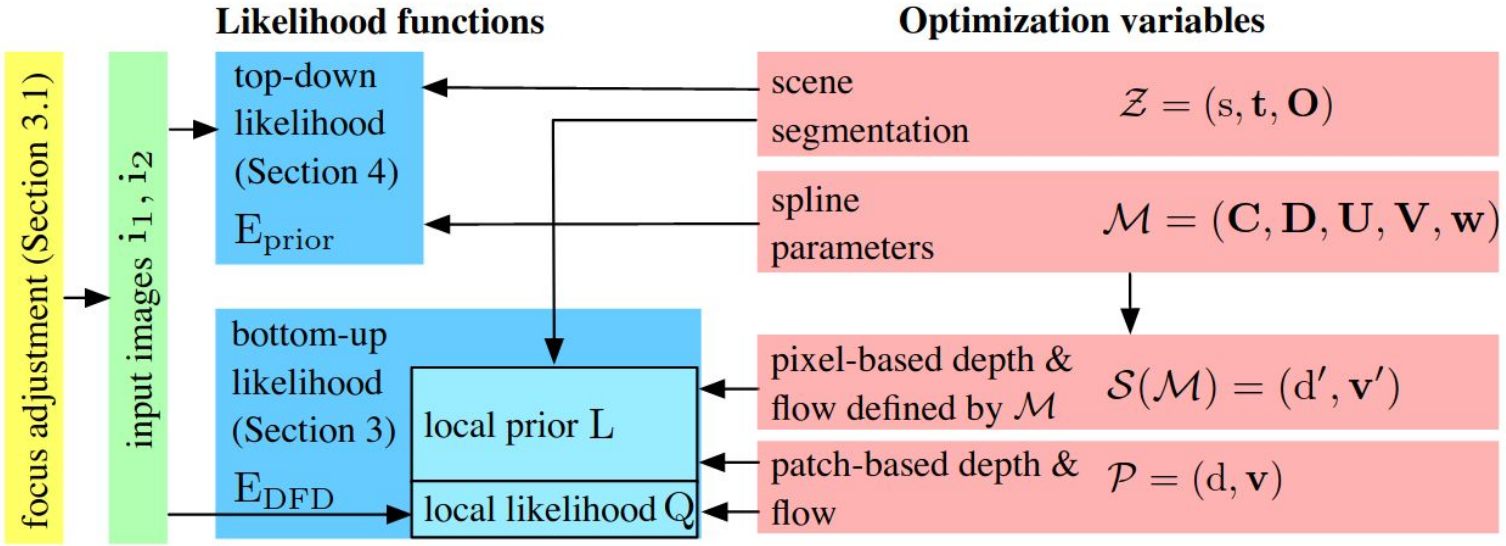


Image w/Blur

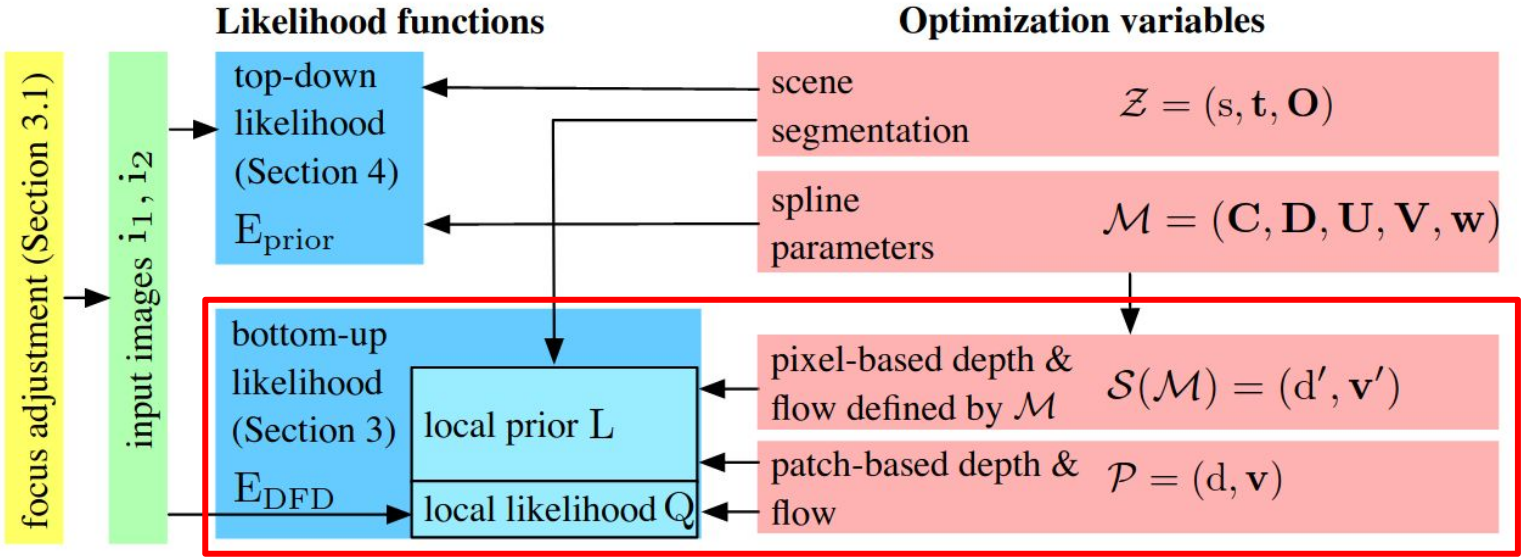




Idea: Combine local depth/flow estimation w/ spline-based scene understanding



Idea: Combine local depth/flow estimation w/ spline-based scene understanding



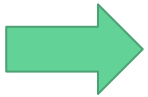
Local Depth/Flow Estimation (Bottom-up Likelihood)

Defocus Equalization Filters

Ensures brightness constancy and equalizes the variances between the two images

$$\mathrm{g}_1^d = \mathcal{F}^{-1}\left[\ \mathcal{F}[\mathrm{k}_2^d] \,/\, \sqrt{\mathcal{F}[\mathrm{k}_1^d]^2 + \mathcal{F}[\mathrm{k}_2^d]^2}\ \right]$$

$$\mathrm{g}_2^d = \mathcal{F}^{-1}\left[\ \mathcal{F}[\mathrm{k}_1^d] \,/\, \sqrt{\mathcal{F}[\mathrm{k}_1^d]^2 + \mathcal{F}[\mathrm{k}_2^d]^2}\ \right]$$



Local Likelihood Q

$$\min_{\alpha} \sum_{\mathbf{q}\in\Omega(\mathbf{p})} \frac{[(\mathrm{i}_1 * \mathrm{g}_1^d)(\mathbf{q}) - \alpha \cdot (\mathrm{i}_2 * \mathrm{g}_2^d)(\mathbf{q}+\boldsymbol{v})]^2}{2\sigma_i^2}$$

Minimize sum-of-squared difference between two equalized images!

Local Depth/Flow Estimation (Bottom-up Likelihood)

Local Likelihood Q

Fit quadratic function to make it analytical for global optimization

$$\min_{\alpha} \sum_{\mathbf{q}\in\Omega(\mathbf{p})} \frac{[(\mathrm{i}_1 * \mathrm{g}_1^d)(\mathbf{q}) - \alpha \cdot (\mathrm{i}_2 * \mathrm{g}_2^d)(\mathbf{q}+\boldsymbol{v})]^2}{2\sigma_i^2}$$



$$\mathrm{Q}_{\mathbf{p}}(d,\boldsymbol{v}) =$$

$$\sigma_{\mathbf{p}}^{-2}(d-\mathrm{d}_{\mathbf{p}}^*)^2 + (\boldsymbol{v}-\mathbf{v}_{\mathbf{p}}^*)^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{p}}^{-1}(\boldsymbol{v}-\mathbf{v}_{\mathbf{p}}^*) + \mathrm{q}_{\mathbf{p}}$$

Local Prior L_{qp}

Local smoothness prior due to overlapping patches

$$\mathrm{L}_{\mathbf{qp}}(d,\boldsymbol{v},d',\boldsymbol{v}',\mathcal{Z}) =$$

$$\begin{cases} \frac{(d-d')^2}{2\sigma_d^2} + \frac{|\boldsymbol{v}-\boldsymbol{v}'|^2}{2\sigma_{\mathbf{v}}^2} & \text{if } \mathbf{q} \in \Omega_f(\mathbf{p}) \\ \tau_o & \text{otherwise} \end{cases}$$

Local Depth/Flow Estimation (Bottom-up Likelihood)

Minimize over LDFD loss to optimize *(d,v)* patch estimates using Markov Random Field (MRF)

$$\min_{\mathbf{s}} \sum_{\mathbf{q}} \underbrace{\mathrm{D}_{\mathbf{q}}(\mathrm{s}_{\mathbf{q}})}_{\textit{data term}} + \sum_{\mathbf{p},\mathbf{q}\in\mathcal{N}_8} \underbrace{\mathrm{V}_{\mathbf{pq}}(\mathrm{s}_{\mathbf{p}},\mathrm{s}_{\mathbf{q}})}_{\textit{smoothness term}}$$

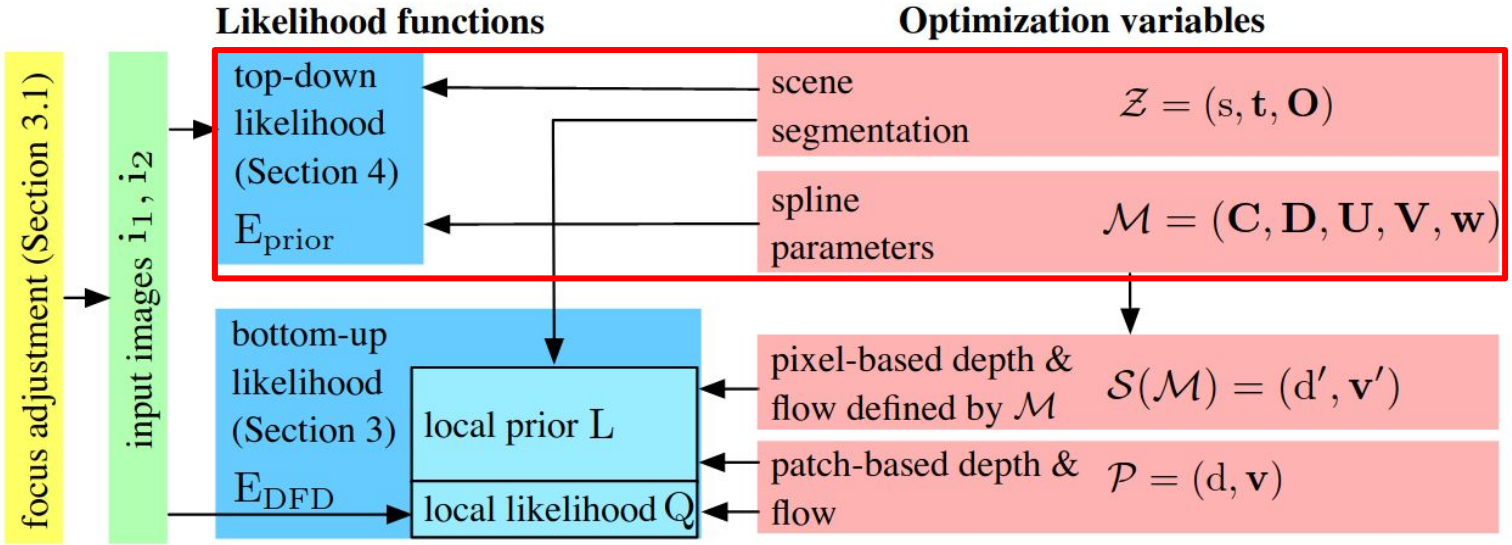
$\mathrm{Q}_{\mathbf{p}}(d, \boldsymbol{v})$

$\mathrm{L}_{\mathbf{qp}}(d, \boldsymbol{v}, d', \boldsymbol{v}', \mathcal{Z})$

Q_p evaluated at every pixel **q** for some *(d,v)* patch pixel **p** estimate (**p** denotes center pixel)

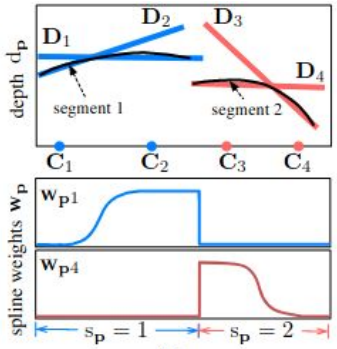
M x M array of values that encodes smoothness between every pixel w.r.t. every pixel

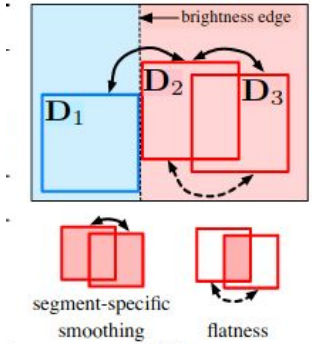
Idea: Combine local depth/flow estimation w/ spline-based scene understanding



Global DFD (Top-Down Likelihood)







Control Point $\mathbf{C_n}$:
- depth plane **D**
- 2D affine transformation **U, V**

Pixel **q** (including control points):
- weight vector
- feature vector
- segment label

Inputs:
Control Points,
Feature Map,
Patch Likelihoods

Update weight vectors

Update occlusion map

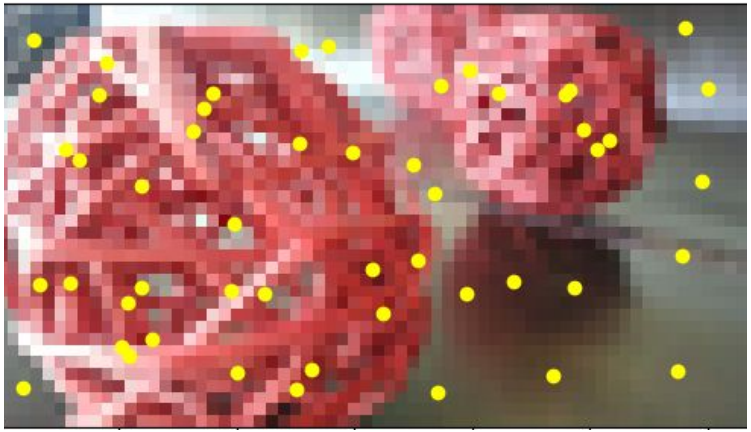
Update depth planes, 2D affine transformations, segment labels of control points

Update control points' feature vectors

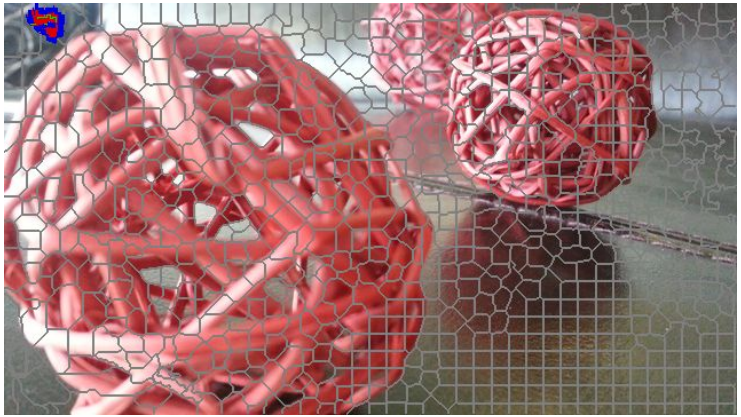
Output:
Pixel-based depth and flow
Patch-based depth and flow
Scene segmentation
Spline parameters

Global DFD (Top-Down Likelihood)

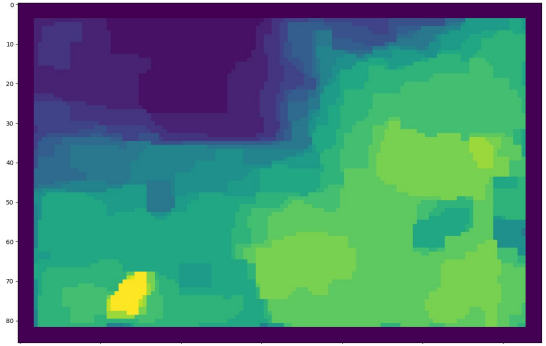
Control Points



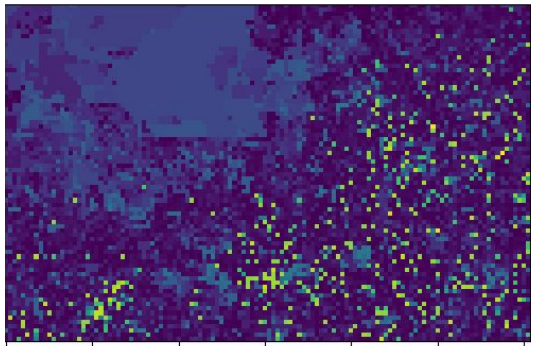
Scene Segmentation



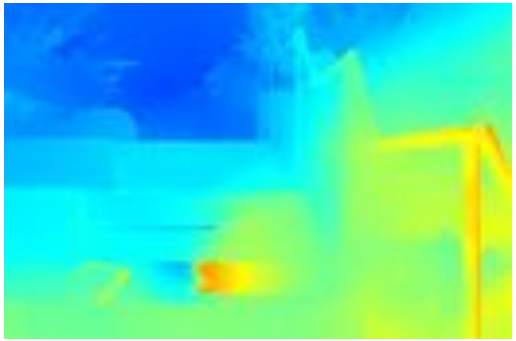
Results



Arbitrary L_{qp} smoothness depth map



Patch-based L_{qp} smoothness depth map

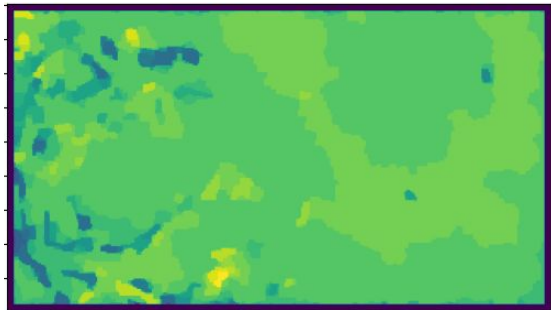


Ground truth depth

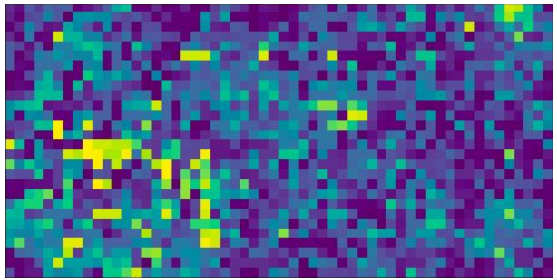


Original input

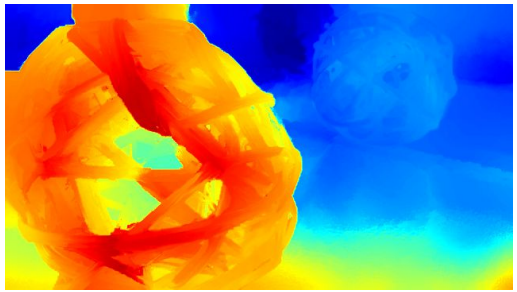
Results



Arbitrary L_{qp} smoothness depth map



Patch-based L_{qp} smoothness depth map



Ground truth depth



Original input

Acknowledgements

Dr. Ioannis Gkioulekas for the tireless support of our project through many OH sessions and Slack messages

Dr. Kiriakos Kutulakos for sharing his research group's dataset and contact information of first authors

References:

Huixuan Tang, Scott Cohen, Brian Price, Stephen Schiller and Kiriakos N. Kutulakos, Depth from defocus in the wild. Proc. IEEE Computer Vision and Pattern Recognition Conference, 2017. https://www.dgp.toronto.edu/WildDFD/

Questions?

A Reconstruction Framework for Time Series Thermal Traces using Mixed Stereo Photography



Arjun Lakshmipathy

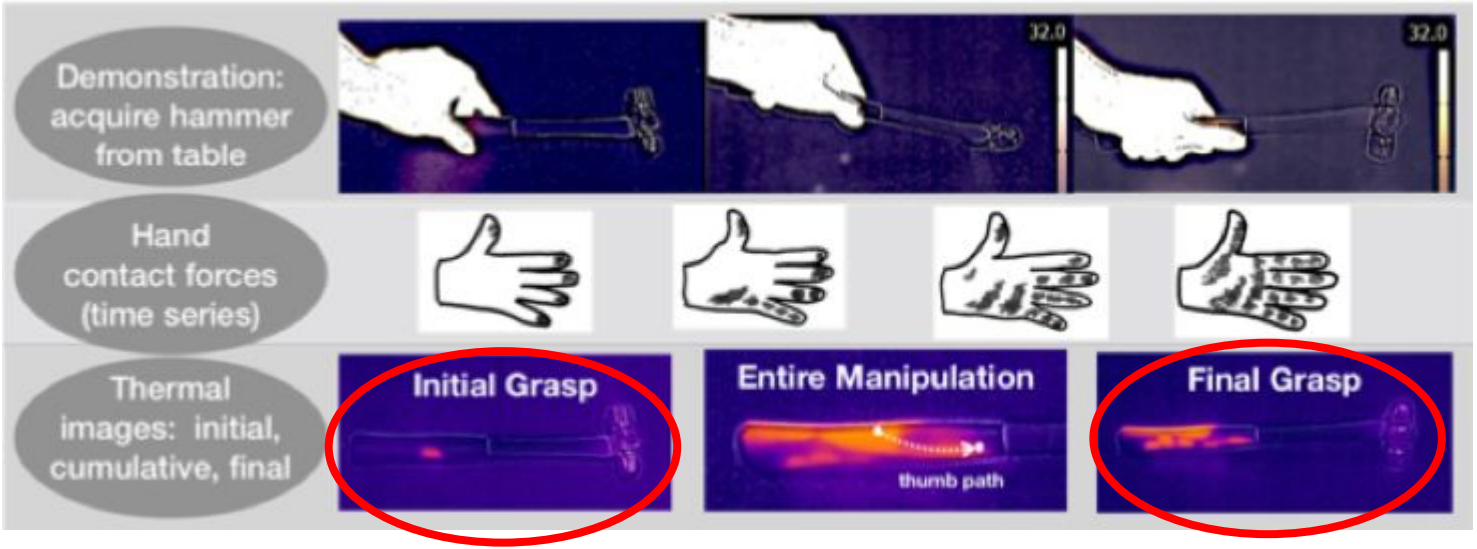
Computational Photography (15-862)

Fall 2020



Context

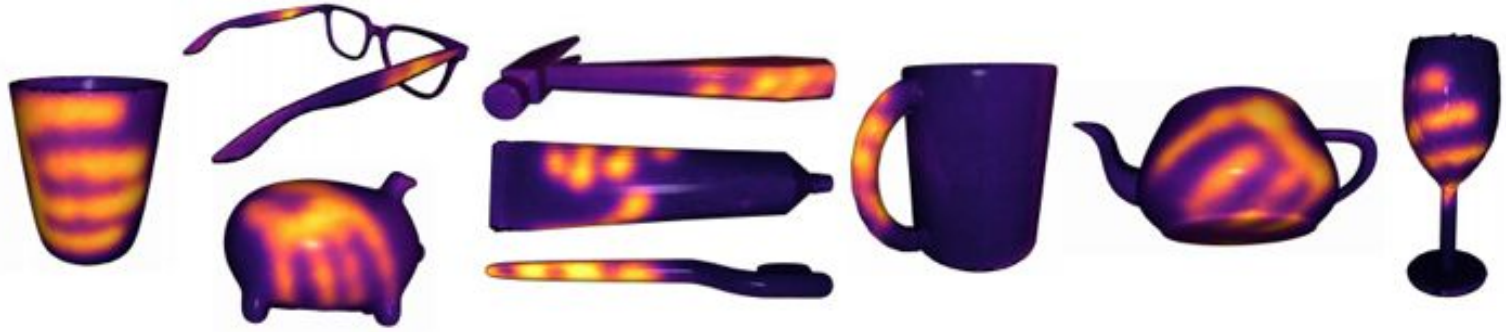
Larger Thrust: Automated Design of Custom, Low-Cost Dextrous Hands from Demonstration



Principal Reference

ContactDB: Analyzing and Predicting Grasp Contact via Thermal Imaging

Samarth Brahmbhatt[1], Cusuh Ham[1], Charles C. Kemp[1] and James Hays[1,2]
[1]Institute for Robotics and Intelligent Machines, Georgia Tech [2]Argo AI
{samarth.robo,cusuh}@gatech.edu, charlie.kemp@bme.gatech.edu, hays@gatech.edu



The Approach

- Difficult to simulate, not so difficult to capture (with the right tools)
- Thermography can detect residual heat traces imparted on objects through human grasping
- With depth + different viewpoints, we can estimate contact texture on arbitrary geometries
- Use multiple cameras and fuse
- Approach: Manipulate physical object, rotate object and capture both thermal and depth images from multiple viewpoints, combine into contact map

The Problems

3 problems to solve:

- Image alignment
- Pose and location estimation of target object within the capture scene
- Texture mapping

Materials and Setup

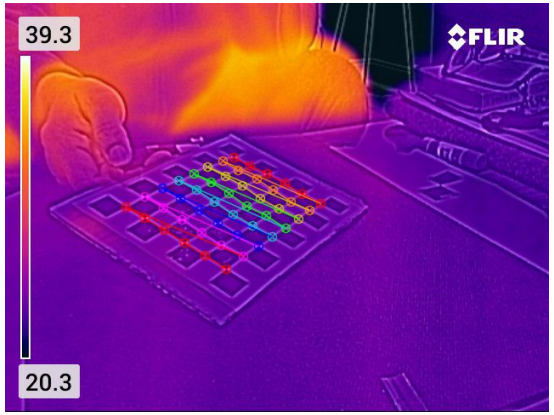
- 1 Intel RealSense D415
- 1 Flir C5
- 2 tripods + 1 clamp mount
- ?? Hand warmers
- 1 dressed up turntable
- 1 calibration target
- 1 ~~house~~ lab space

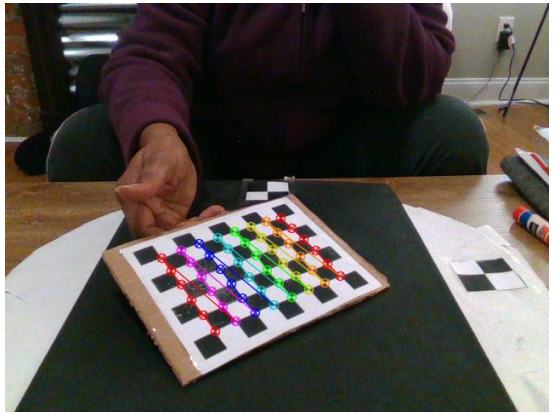
Turntable and second tripod ~~taken~~ borrowed from CMU Motion Capture Lab



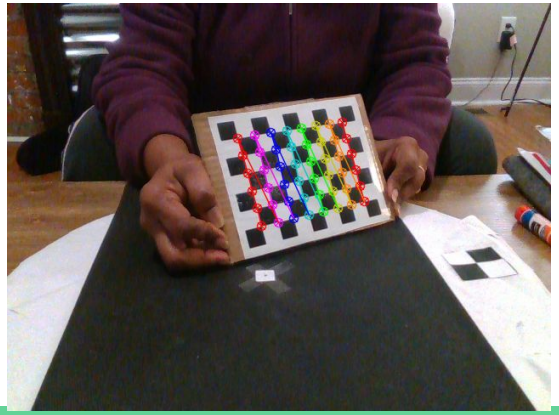
Process Step 1: Calibration

- Thermal intrinsics
- Stereo Calibration
- ~ 10 images apiece

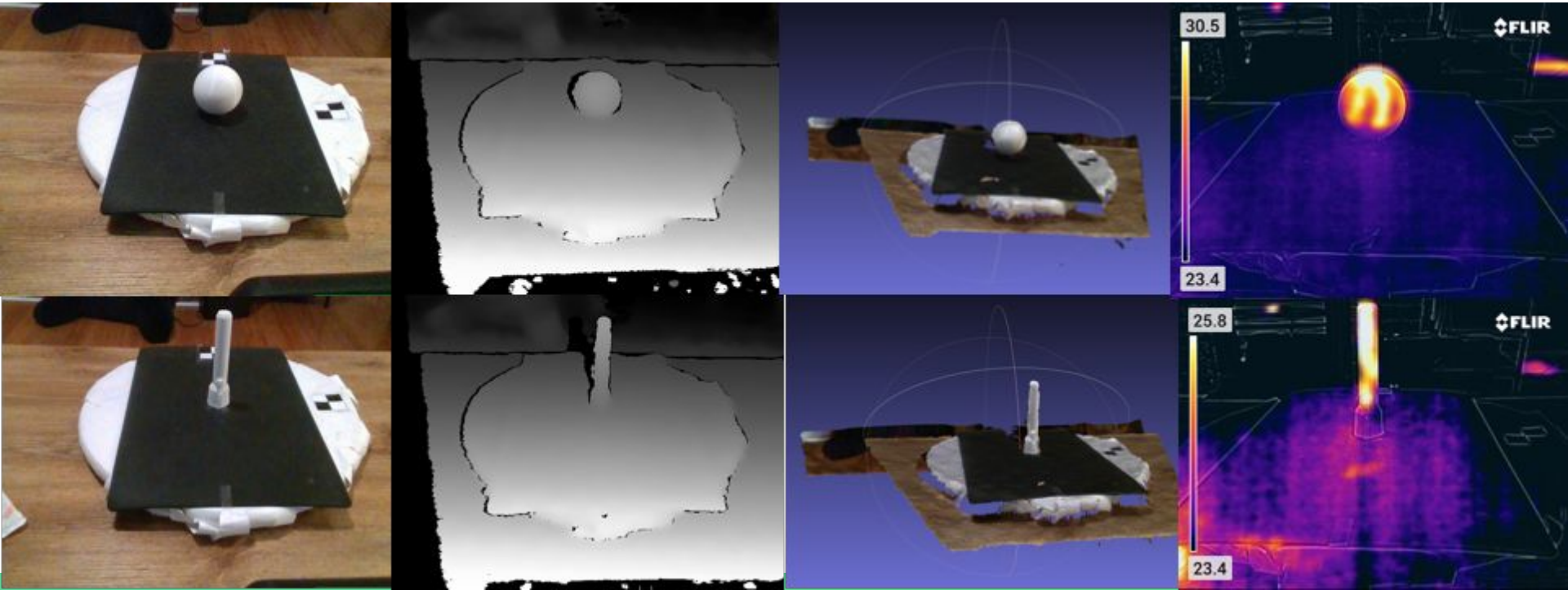






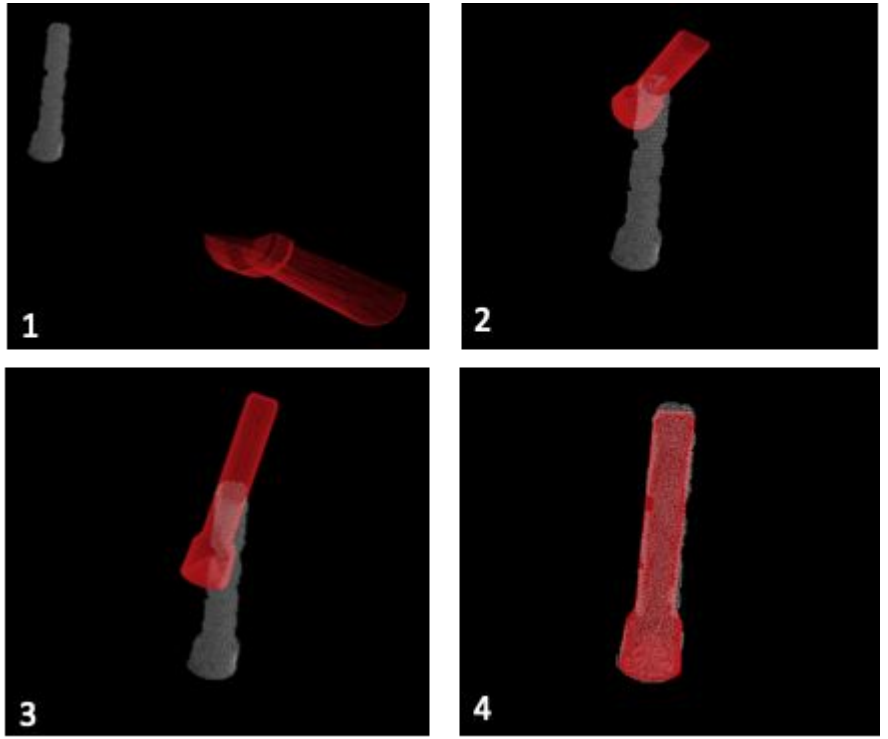


Process Step 2: Capture

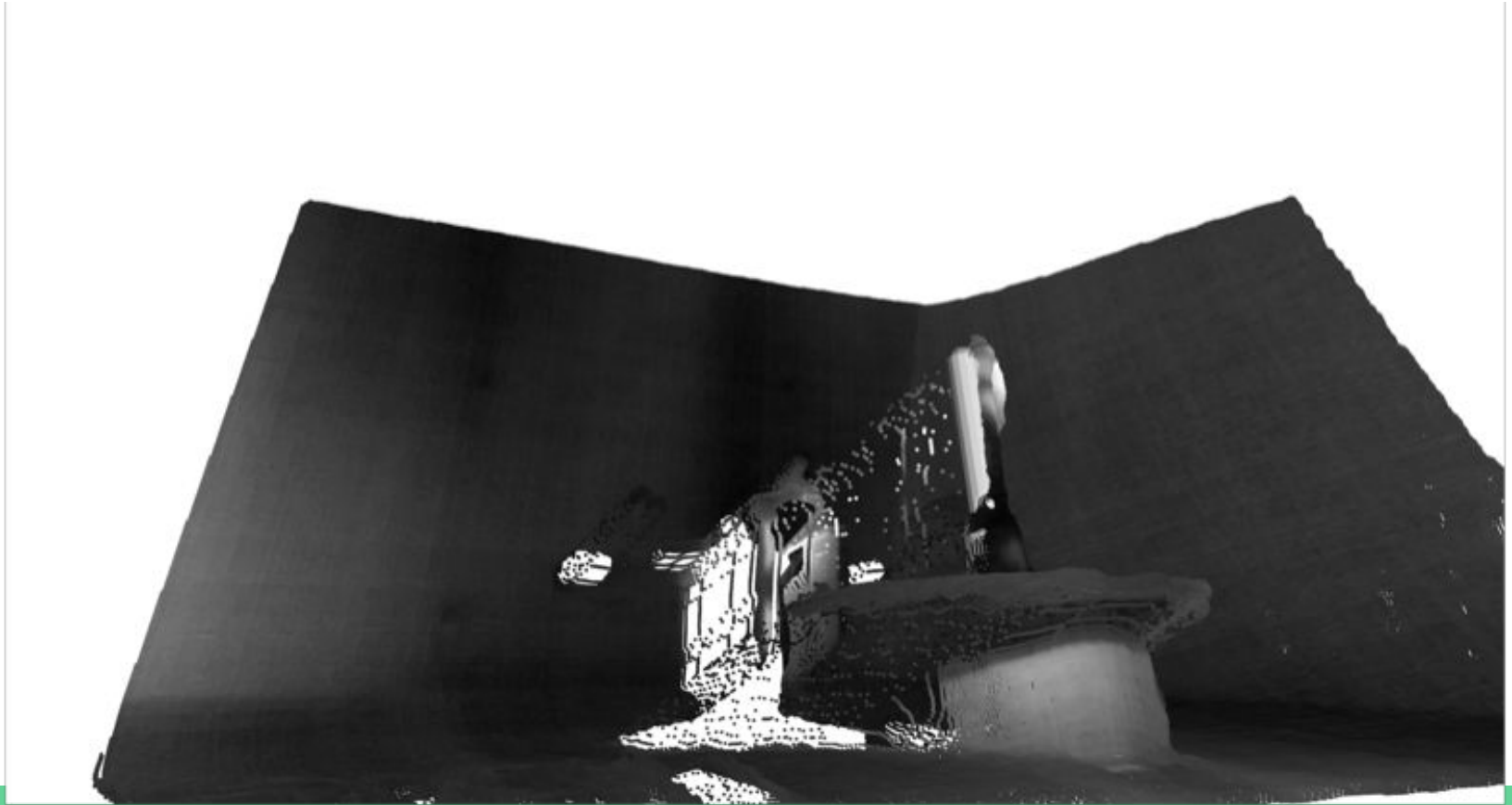


Process Step 3: Object Pose Estimation

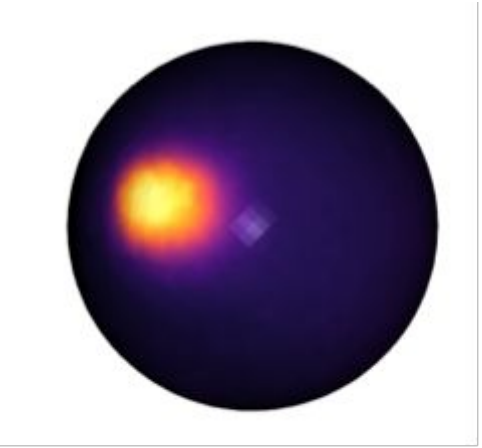
- Entirely in Depth CCS
- Load 3D model, convert to point cloud
- Segment scene to extract object
- Solve R and T using Iterative Closest Point Method (ICP) [2]

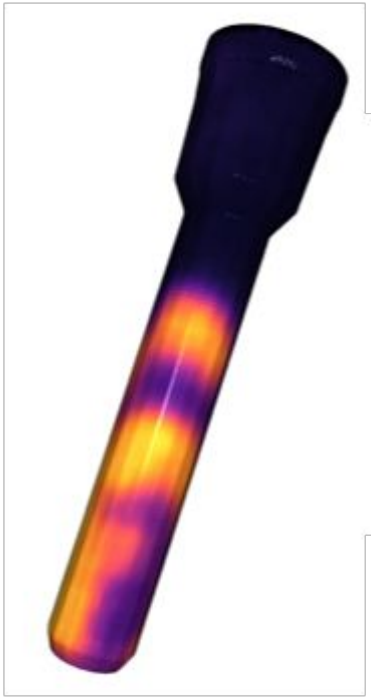


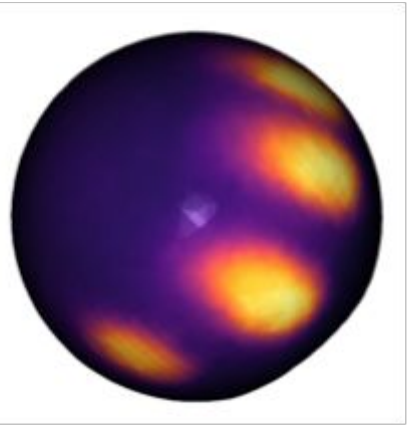
Process Step 4*: Alignment

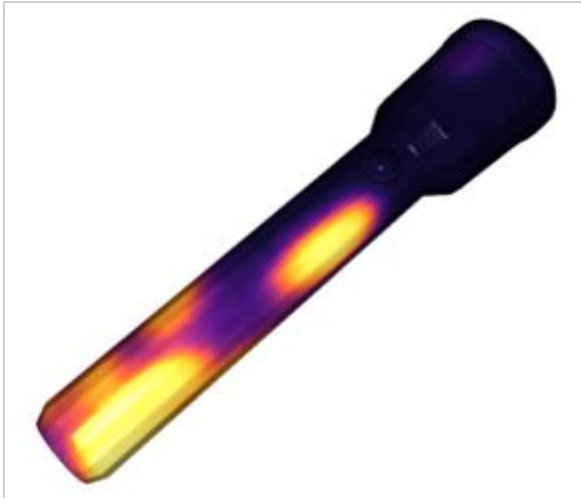


Process Step 5*: Texture Mapping via Color Map Optimization [3]









Concluding Remarks

- Process: Calibration -> Capture -> Pose Estimation -> Alignment -> Texture Mapping
- Works reasonably…in principle. Still some items to resolve for full operation
- Next steps: work through bugs and nuances with each step, collect initial and final grasps for multiple objects from multiple subjects. Use as basis for next research thrust.

*Synthesized from dataset captures only / not my own yet

Thanks for listening! Questions?

References

[1] S. Brahmbhatt, C. Ham, C. C. Kemp, and J. Hays, “Contactdb: Analyzing and predicting grasp contact via thermal imaging,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8709–8719, 2019.

[2] PJ Besl and Neil D McKay. A method for registration of 3- d shapes. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 14(2):239–256, 1992

[3] Qian-Yi Zhou and Vladlen Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. ACM Transactions on Graphics (TOG), 33(4):155, 2014.

HDR Image Reconstruction using Hallucinated exposure stack

Shamit Lal(shamitl), Sajal Maheshwari(sajalm)

Problem

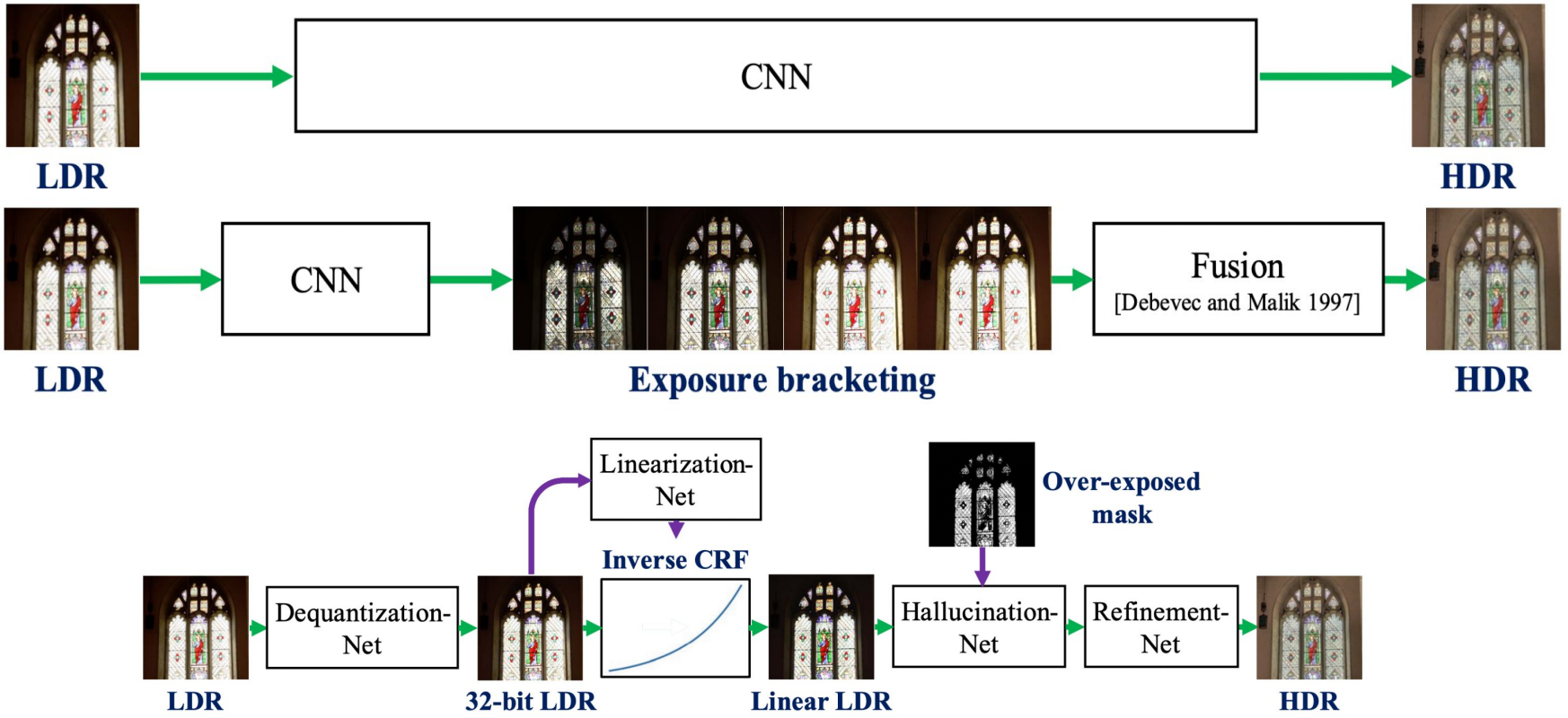




Traditional approach

- Collect a stack of images at different exposures
- If the images are non-linear, linearize them.
- Merge the linear exposure stack.
- If we want to display the image, we can also do a tonemapping operation

Deep learning based methods



What if ?

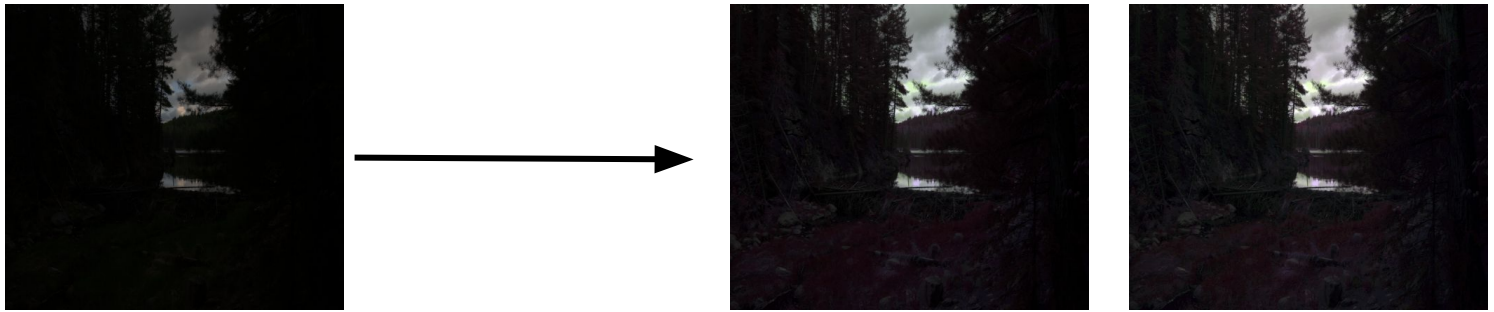
- We can merge these ideas and ：
  - Create a linearized and de-quantized exposure stack using a single input image.
  - Use this linearized stack and weights for each of the pixels and generate an output HDR image.
  - Instead of traditional merging method, let a deep network decide the weights for each of the pixels in the stack.

Our approach

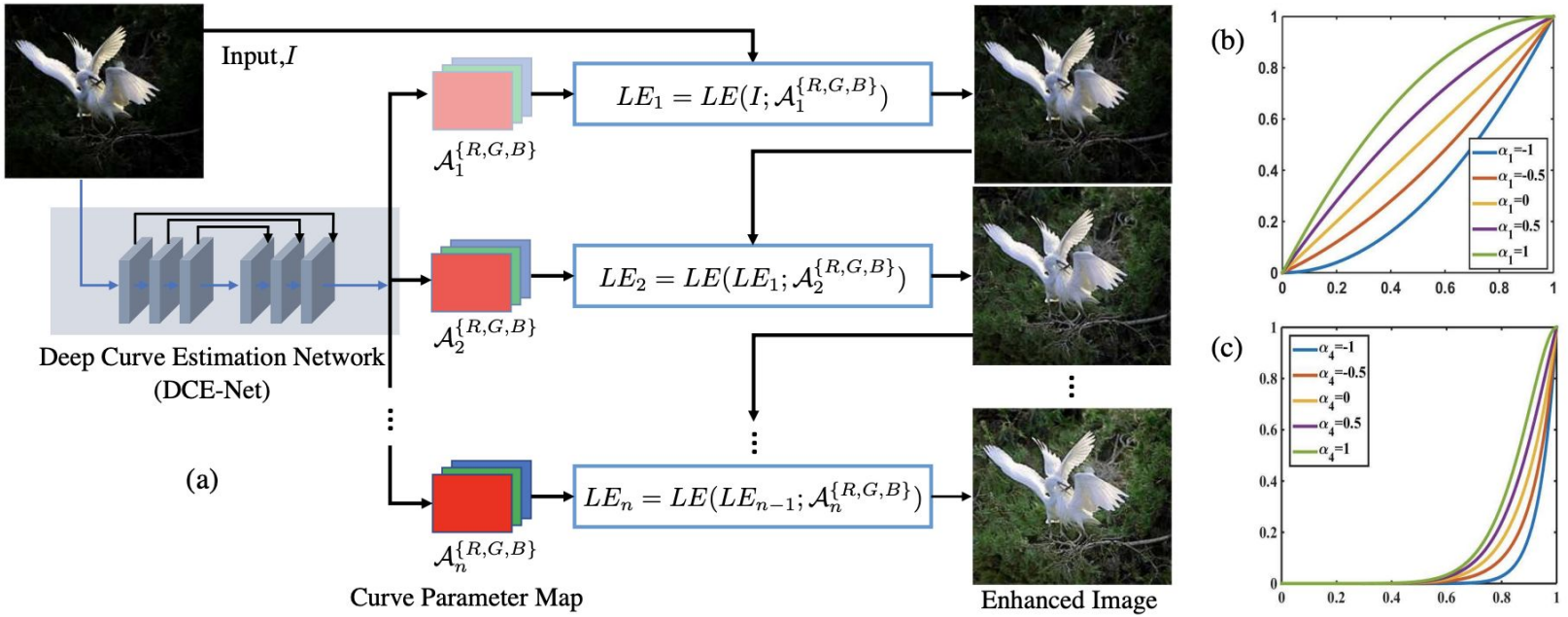
- Step 1 : Exposure stack generation
  - Images should be sharp.
  - The changes should be in brightness, not in color.
  - No multiple iterations
  - Possible approaches
    - Naive encoder-decoder
    - Mapping based model
    - Modified encoder-decoder

Naive encoder-decoder

- A simple single encoder- multi decoder model with different decoder heads having supervision corresponding to different images of the exposure stack.
- Result :
  - Blurry images
  - Images with artefacts like color inconsistency
  - Not great results as the output images looked very similar to the original input.



Mapping based approach



Mapping based approach

- The approach described in the previous slide uses multiple iterations. Therefore, although being a good starting point, it can not be used in our experiments.
- We now explain our approach, which can directly give us a mapping of higher degree.
- We output a 256 dimension vector (for each channel). Now, we make this vector monotonic and bound it from 0 to 1, and then for intensity value at each pixel, pick the output intensity value to be the value at the index of the input intensity.

0	0.2	0.4	0.7	.	.	.	1

Mapping based approach - Problem!

- This operation is non-differentiable. How to solve this ?
- During training, use an extra input of 256 x H x W dimensions, with each of the entries containing a one-hot vector. Multiply each of these values with the output vector (at each pixel value) earlier element-wise and sum the values along the channel dimension.
- Additionally, all the outputs should be identical, which we incorporate by adding loss to make all the pixel-wise embeddings closer to the mean embedding.
- Better and sharper results!

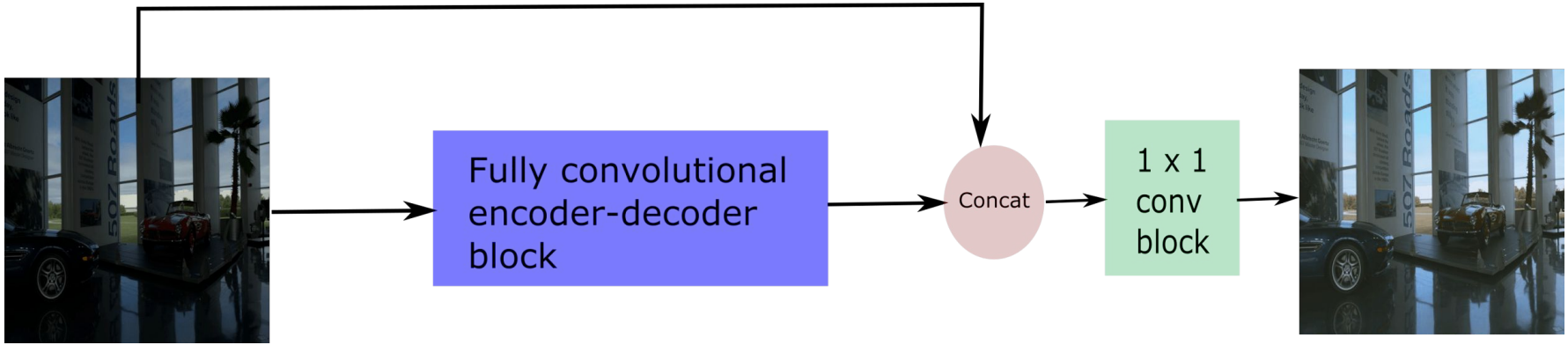
Mapping based approach - Results



- The results are much better now, however, we can see some color cast. This might be because we are creating the function for all channels. We are currently trying to run this method only on the luminance channel.

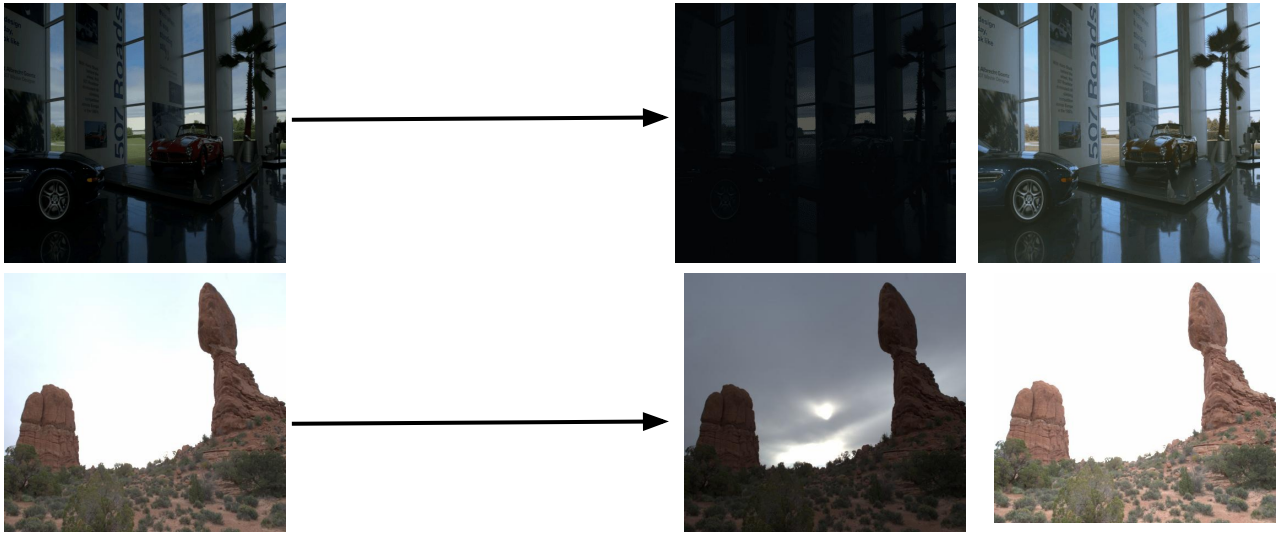
Modified encoder-decoder

- Instead of using a naive encoder-decoder trying to learn the output, try to learn the difference between the output and input instead (has been done before for many many problems!)



Modified encoder-decoder

- The modified encoder-decoder ensures that the results are close to the input but also have the brightness of the output where supervision is provided.
- The 1x1 convolution ensures that the output we get is sharp and not blurry outputs.



Merging HDR

- Once we get the stacks, we can either merge the images to get the HDR image using traditional approaches or a learning based method, which can again consist of an encoder-decoder with shared weights for each of the outputs.
- We have not yet run any tone-mapping model, so the HDR images generated look a bit dark. However, the images are decent when viewed in HDR viewers like openhdr(https://viewer.openhdr.org/). However, this is not working as expected and we believe there is still some training/tuning required.



Unsupervised methods ?

- We have currently been providing supervision at every stage - what happens if we try to generate the exposure in an unsupervised fashion ?
- Currently running experiments for unsupervised methods



References

1.Single-Image HDR Reconstruction by Learning to Reverse the Camera Pipeline, Yu-Lun Liu, Wei-Sheng Lai, Yu-Sheng Chen, Yi-Lung Kao, Ming-Hsuan Yang, Yung-Yu Chuang, Jia-Bin Huang (https://arxiv.org/abs/2004.01179)

2.Deep Reverse Tone Mapping, Yuki Endo, Yoshihiro Kanamori, and Jun Mitani(http://www.cgg.cs.tsukuba.ac.jp/~endo/projects/DrTMO/)

3.HDR image reconstruction from a single exposure using deep CNNs,Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K. Mantiuk, Jonas Unger(https://arxiv.org/abs/1710.07480)

4. Zero-Reference Deep Curve Estimation for Low-Light Image Enhancement, Chunle Guo, Chongyi Lim Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, Runmin Cong(https://openaccess.thecvf.com/content_CVPR_2020/papers/Guo_Zero-Reference_Deep_Curve_Estimation_for_Low-Light_Image_Enhancement_CVPR_2020_paper.pdf)

5. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks , Jun-Yan Zhu， Taesung Park, Phillip Isola ,Alexei A. Efros(https://junyanz.github.io/CycleGAN/)

6. ExpandNet: A Deep Convolutional Neural Network for High Dynamic Range Expansion from Low Dynamic Range Content, Demetris Marnerides, Thomas Bashford-Rogers, Jonathan Hatchett, Kurt Debattista(https://arxiv.org/abs/1803.02266)

Shape Estimation

Qiqin Le, Qiao Zhang

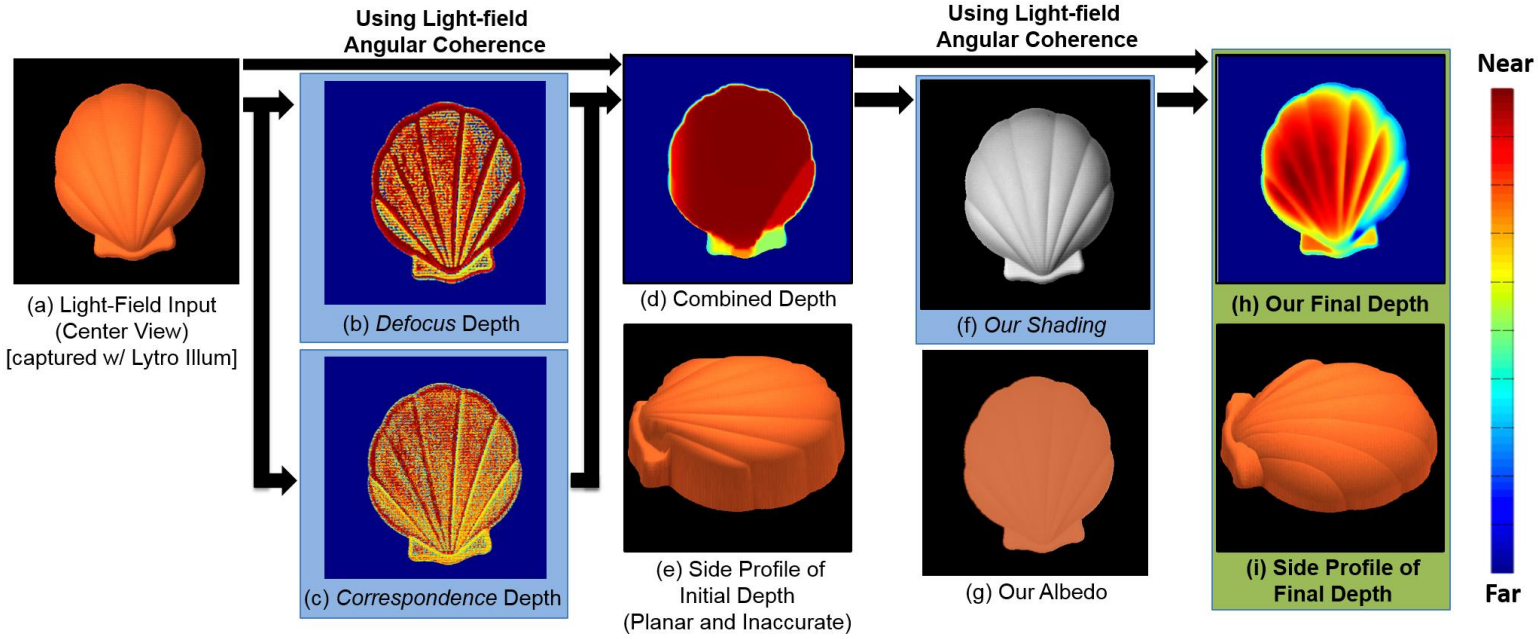
Overview

- **Limitations:**
  - **Depth from defocus:**
    - Affected by occlusions
    - Ambiguity
  - **Depth from correspondence:**
    - Correspondence errors
    - Matching ambiguity at repeating patterns and noisy regions
    - Affected by occlusions
  - **Combining depth from defocus and correspondence**
    - Require complicated algorithms / Camera modifications / Multiple image exposures

- **Improvement:**
  - Use angular coherence to improve robustness.

- **Pipeline:**
  - Combine defocus and correspondence metrics for dense depth estimation and then uses the additional cue of shading to refine details in the shape.

Pipeline



Tao, Michael W., et al. "Shape estimation from shading, defocus, and correspondence using light-field angular coherence."

Image Source



http://lightfield.stanford.edu/lfs.html





Our images taken by the Lytro Illum camera

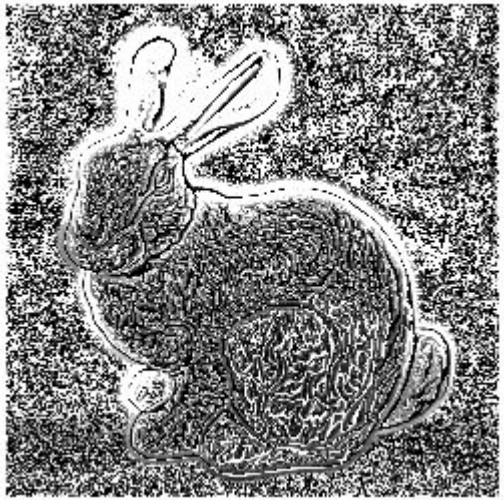
Defocus Depth

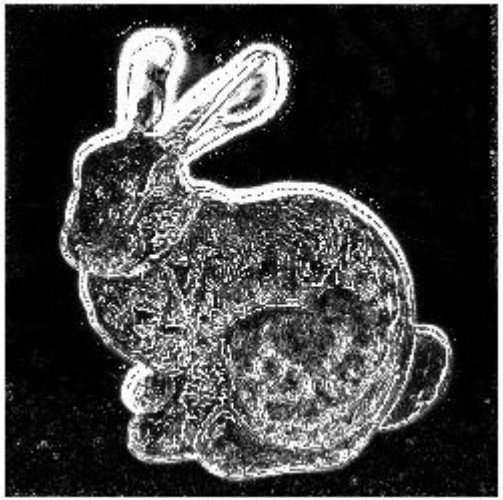
$$\bar{L_\alpha}(x,y) = \frac{1}{N_{(u,v)}} \sum_{(u',v')} L_\alpha(x,y,u',v') \tag{8}$$

$$D_\alpha(x,y) = \frac{1}{|W_D|} \sum_{(x',y') \in W_D} \left| \bar{L}_\alpha(x',y') - P(x',y') \right| \tag{9}$$

Defocus Depth

Depth Map & Confidence Map & Shape Estimation



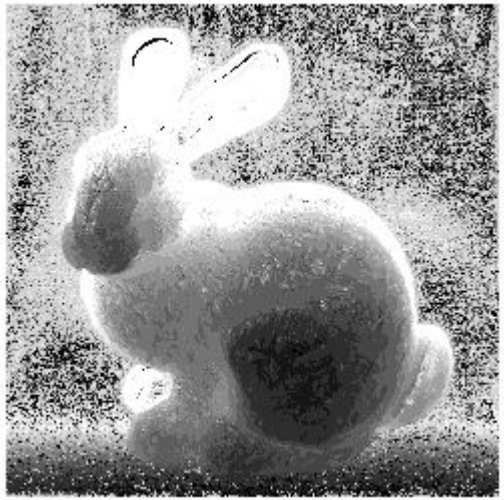


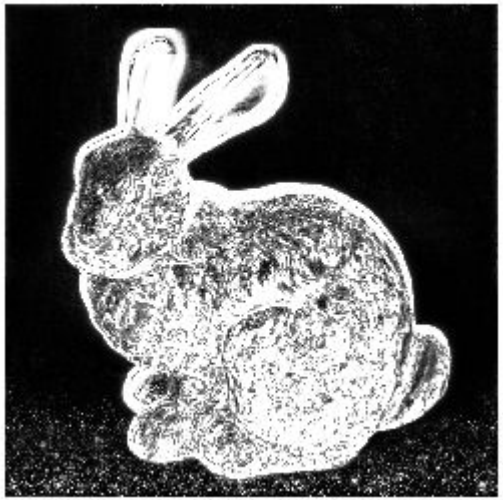
Correspondence Depth

$$C_\alpha(x,y) = \frac{1}{N_{(u',v')}} \sum_{(u',v')} |L_\alpha(x,y,u',v') - P(x,y)| \quad (10)$$

Correspondence Depth

Depth Map & Confidence Map





Combined Depth from Defocus and Correspondence

$$E(Z^*) = \sum_{(x,y)} \lambda_d E_d(x,y) + \lambda_v E_v(x,y) \tag{11}$$

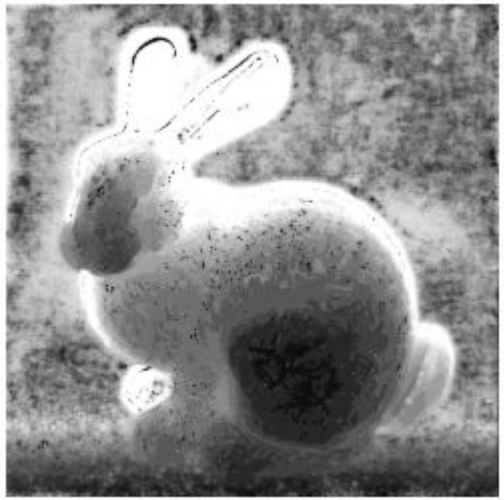
$$E_d(x,y) = Z_{\text{conf}}(x,y) \cdot ||Z^*(x,y) - Z(x,y)||^2 \tag{12}$$

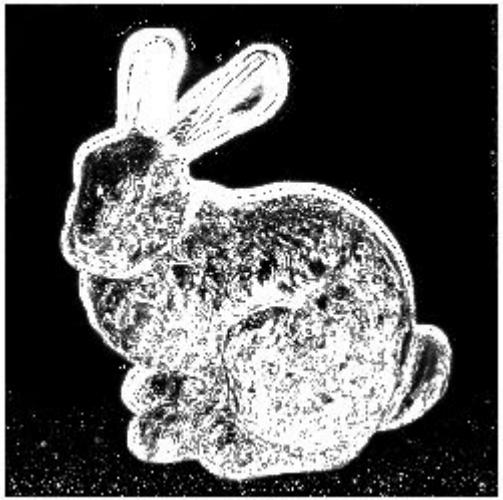
$$E_v(x,y) = \sum_{i=1,2,3} ||(Z^* \otimes F_i)(x,y)||^2 \tag{13}$$

$$F_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} F_2 = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} F_3 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \tag{14}$$

Combined Depth from Defocus and Correspondence

Depth Map & Confidence Map





Shading Depth

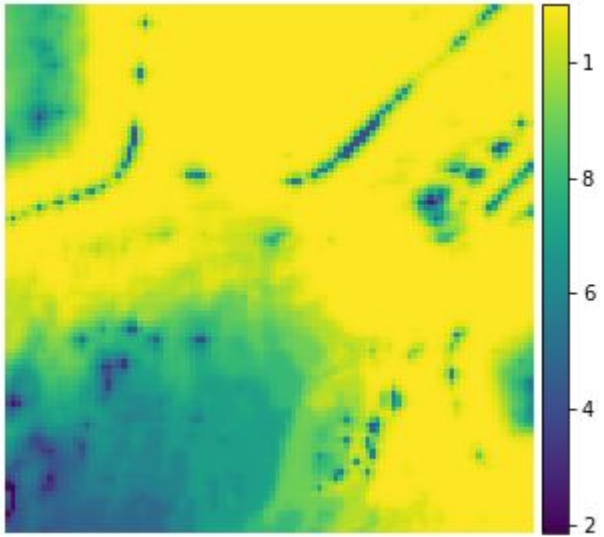
Albedo



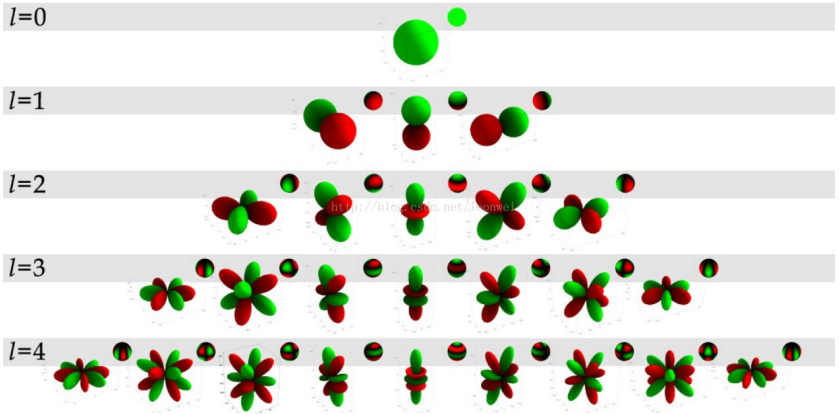
$$E(s) = \sum_{t=(x,y,u,v)} E_{ls}(t) + E_{la}(t) + E_{ns}(t) + E_{na}(t) + E_{ac}(t). \quad (16)$$

Shading Depth

Spherical Harmonic

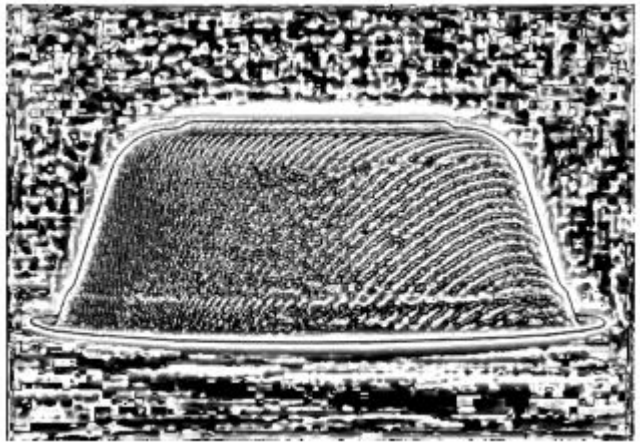


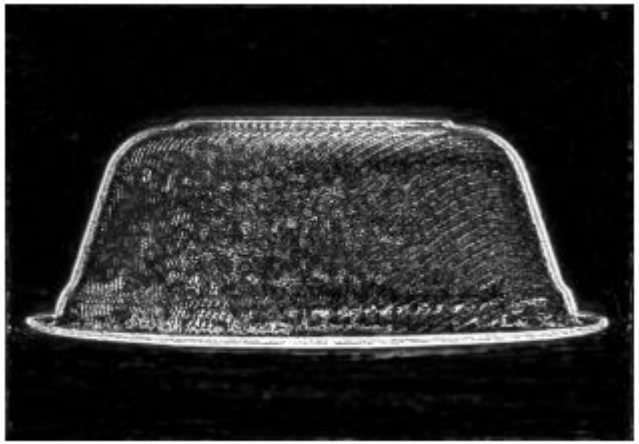
$$E_s(x,y) = w_s(x,y) \cdot || \sum_{k=0}^{8} l_k H_k(Z^*(x,y)) - S||^2 \quad (23)$$



Defocus Depth

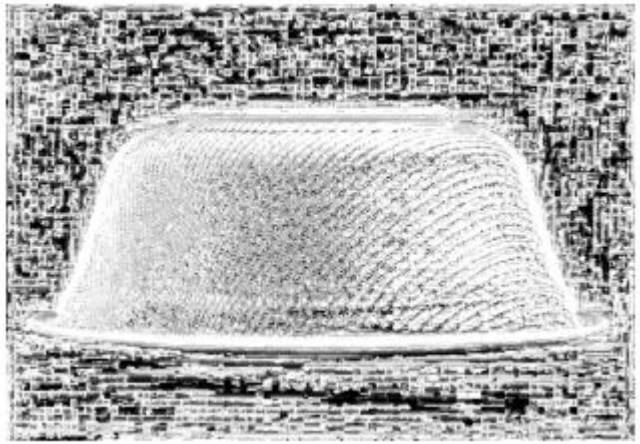
Depth Map & Confidence Map

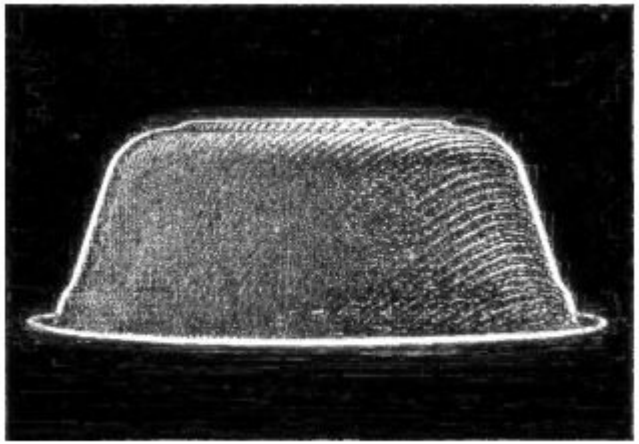




Correspondence Depth

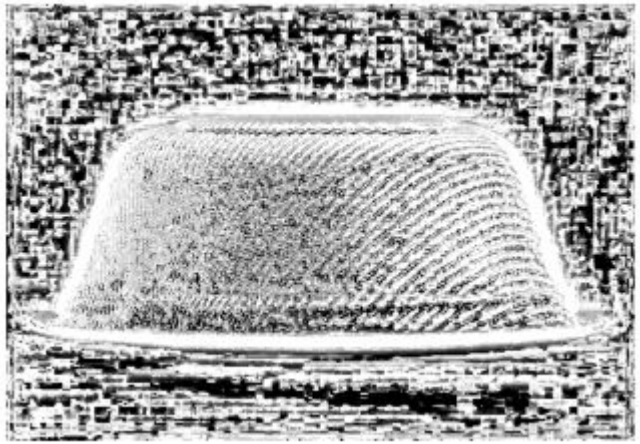
Depth Map & Confidence Map

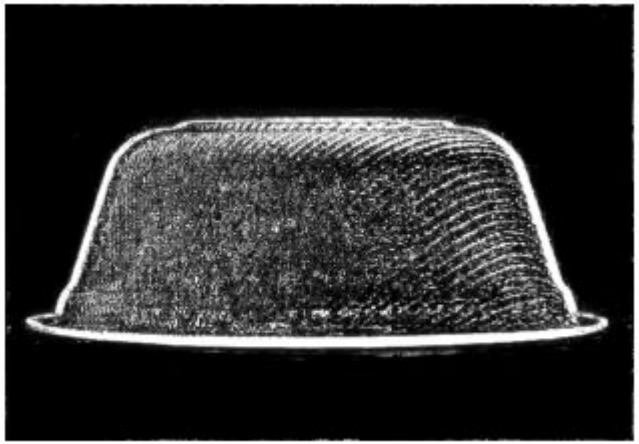




Combined Depth from Defocus and Correspondence

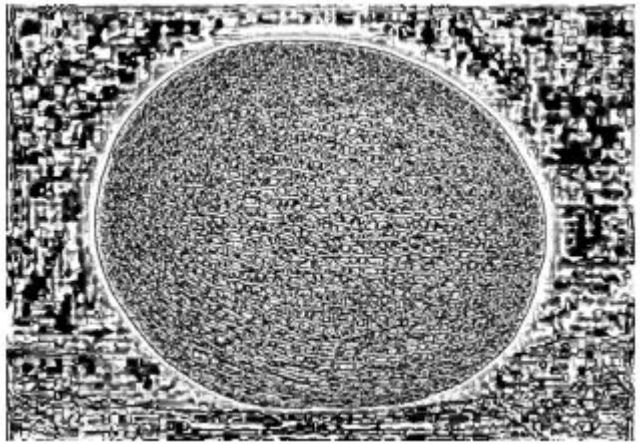
Depth Map & Confidence Map

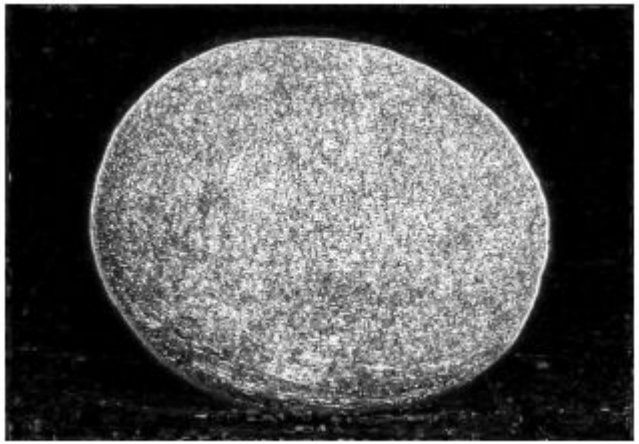




Defocus Depth

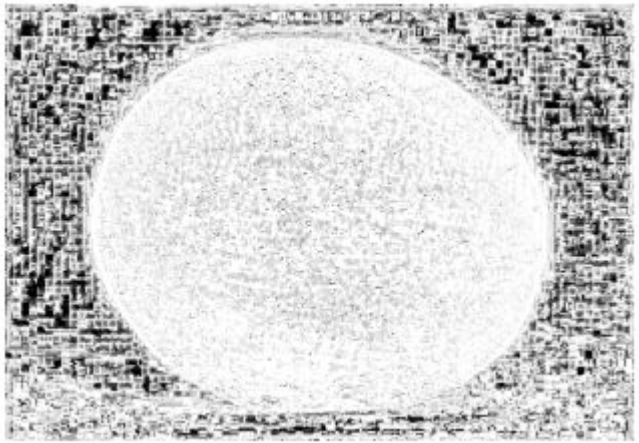
Depth Map & Confidence Map

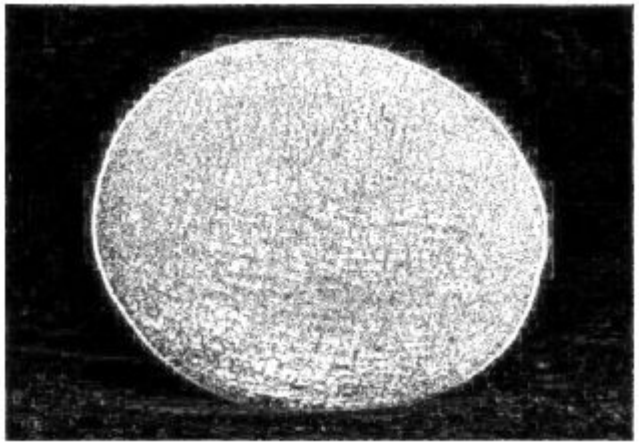




Correspondence Depth

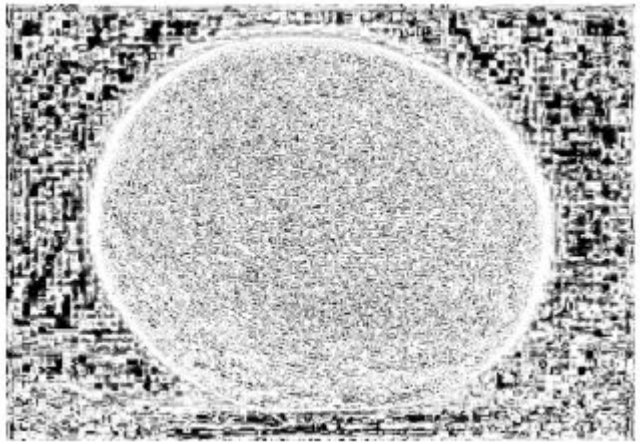
Depth Map & Confidence Map

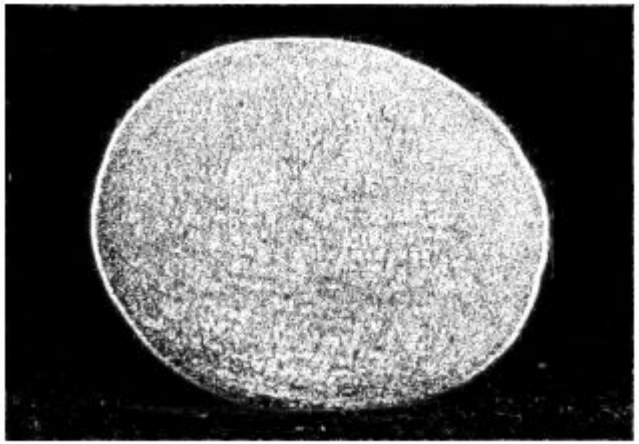




Combined Depth from Defocus and Correspondence

Depth Map & Confidence Map





References

[1] Tao, Michael W., et al. "Shape estimation from shading, defocus, and correspondence using light-field angular coherence." IEEE transactions on pattern analysis and machine intelligence 39.3 (2016): 546-560.

[2] Tao, Michael W., et al. "Depth from combining defocus and correspondence using light-field cameras." Proceedings of the IEEE International Conference on Computer Vision. 2013.

Q & A

Normal Estimation for Transparent Objects

Amy Lee

Overview + Motivation

- Naive photometric stereo methods do not work well for specular or transparent objects.
- Improvement idea:
  - Use a reference chrome sphere to estimate the light direction for each image.
  - Estimate the normals at highlighted points on the object using the reference light direction.

Pipeline

Image Set



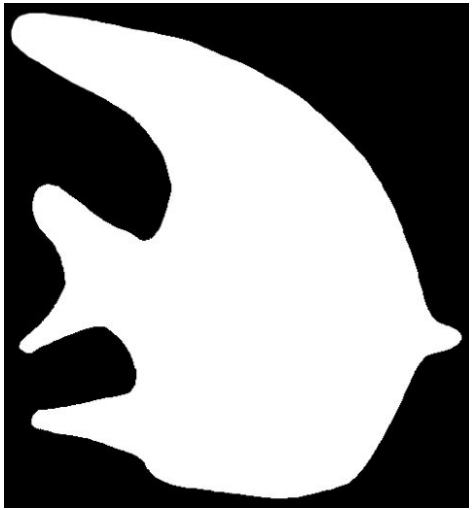
- Video of static scene, moving spot light (dense set of images under variable illumination)
- Object set next to chrome sphere
- Assume orthographic camera

Image Segmentation

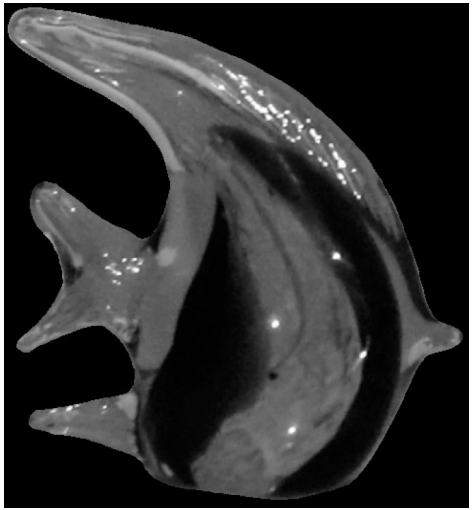
Image



Mask



Final

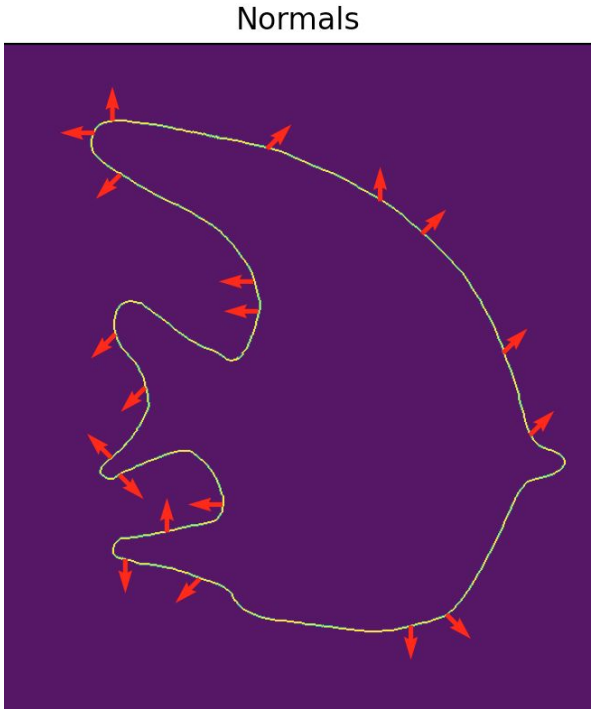


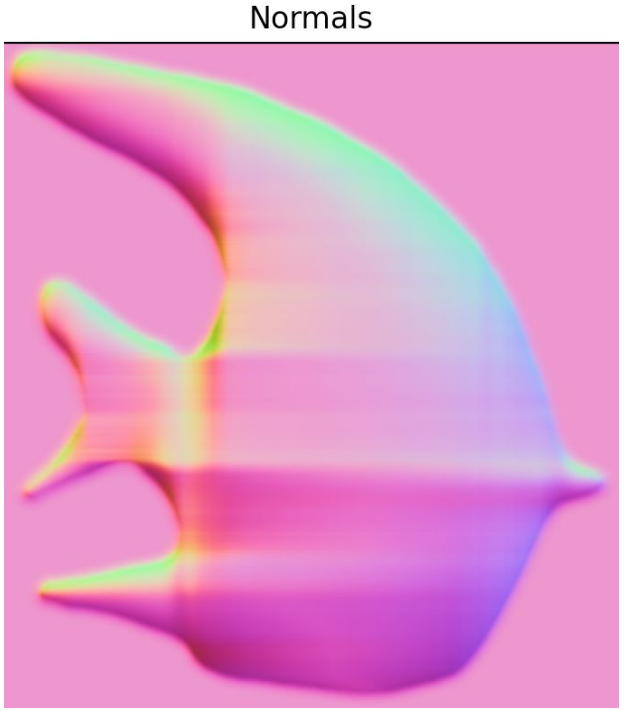
Initial Silhouette-based Surface Normals

Key Ideas:

- Surface normals are parallel to the image plane at the silhouette of the object.
- Interpolating the silhouette normals gives a decent approximation of the surface.
- Caveat: Initial normals heavily influence quality of final result.

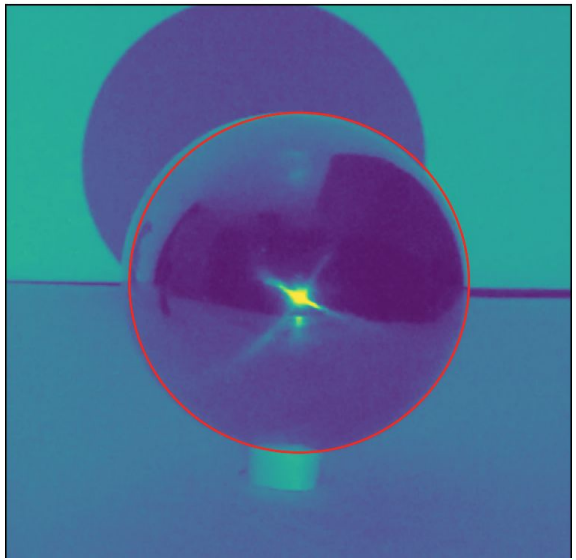
Initial Silhouette-based Surface Normals



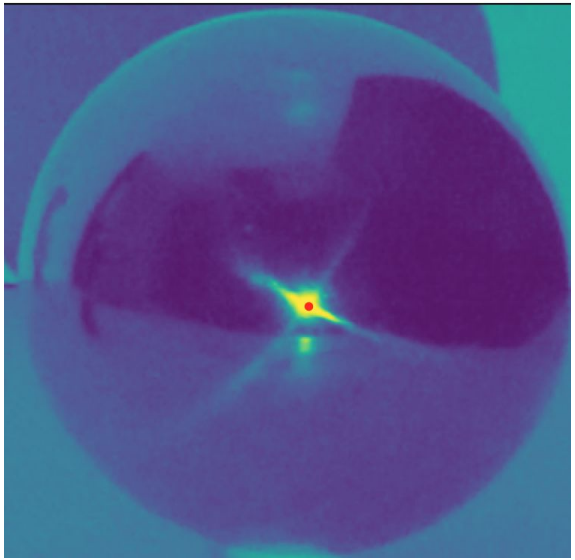


Orientation Consistency w/ Reference Sphere

Estimated Sphere Location(s)

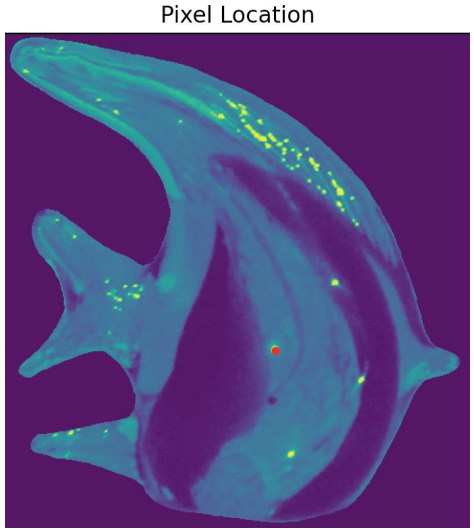


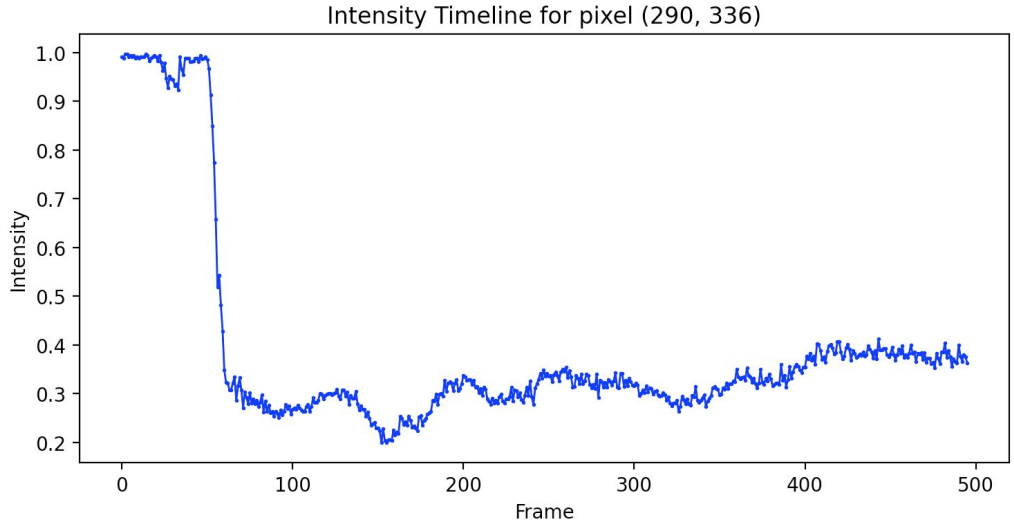
Estimated Highlight Location



Normal Clustering

- Can infer that highlights occur when intensity exceeds some threshold (e.g. >0.9). Highlight occurrences form clusters in time.





Normal Clustering

- May have multiple clusters due to interreflections (not true direct highlights).
- Find top 2 “salient” clusters (i.e. the 2 clusters with the most observations) corresponding to specular highlights for every pixel.
- Record the mean normal corresponding to each cluster (computed using reference sphere).

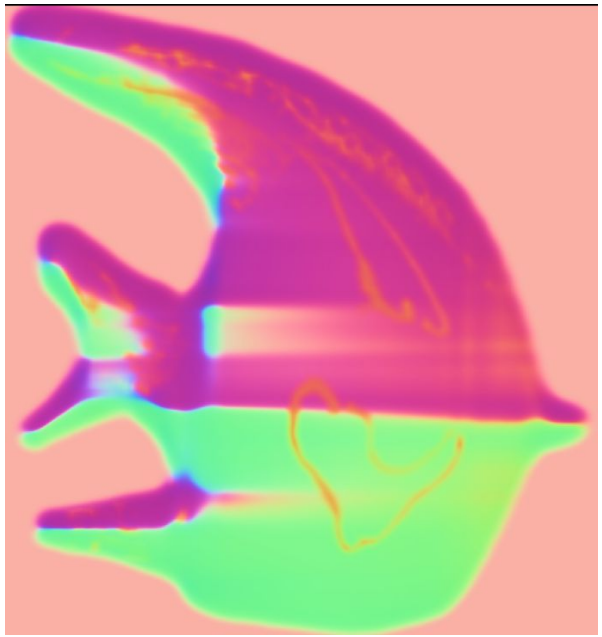
Finding Optimal Normal Clusters

Graph-cut cost minimization using max flow algorithm.

- At most 2 nodes per pixel (1 node per salient normal cluster).
- Want to label nodes as being either true or false highlights. If true, use reference normal.
- Edges between adjacent pixels' nodes.
- Cost (energy): difference with initial normal (E_1) + neighboring normals (E_2)

$$E(\mathbf{S}) = \sum_{i \in \mathcal{V}} E_1(s_i) + \lambda \sum_{\{i,j\} \in \mathcal{E}} E_2(s_i, s_j)$$

Final Calibrated Normals



Quality of results are heavily influenced by the initial normals

References

1. Yeung, Sai-Kit & Wu, Tai-Pang & Tang, Chi-Keung & Chan, Tony & Osher, Stanley. (2015). Normal Estimation of a Transparent Object Using a Video. Pattern Analysis and Machine Intelligence, IEEE Transactions on. 37. 890-897. 10.1109/TPAMI.2014.2346195.
2. H.-S. Ng, T.-P. Wu, and C.-K. Tang, “Surface-from-gradients without dis-crete integrability enforcement: A Gaussian kernel approach,”IEEE Trans.Pattern Anal. Mach. Intell., vol. 32, no. 11, pp. 2085–2099, Nov. 2010.

Thank you for your time!

Event-based Object Tracking-by-Detection

Jessica Lee

Why Event-Based Vision?

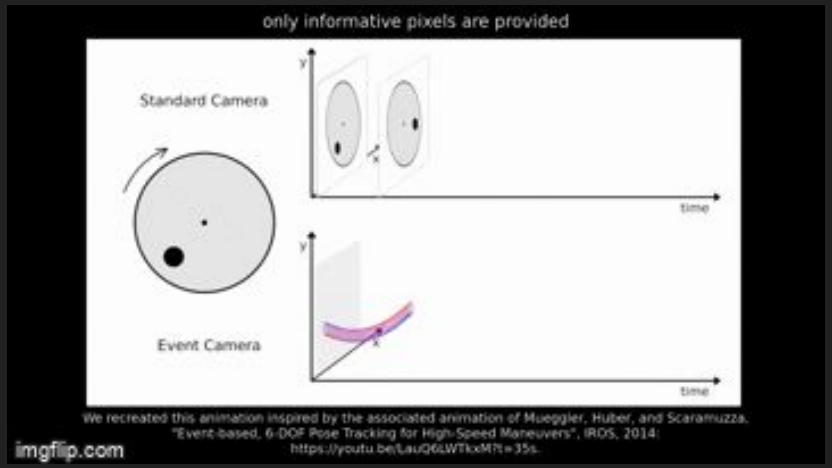
Event-based vision measures log-based changes in brightness for each pixel in the sensor *independently*

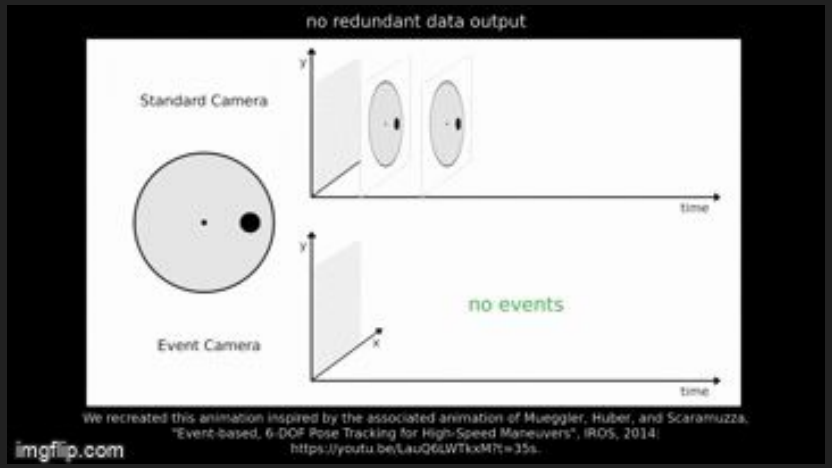
Data point: (ts, x, y, p)

Polarity - moving in/out of a scene

In comparison to a typical camera, which captures via rolling shutter synchronously [1].

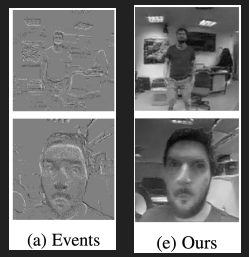
- Little motion blur
- High-dynamic range





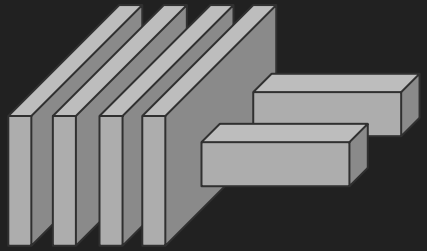
Prior Work on Event-Based Object Detection

Image Reconstruction [2]

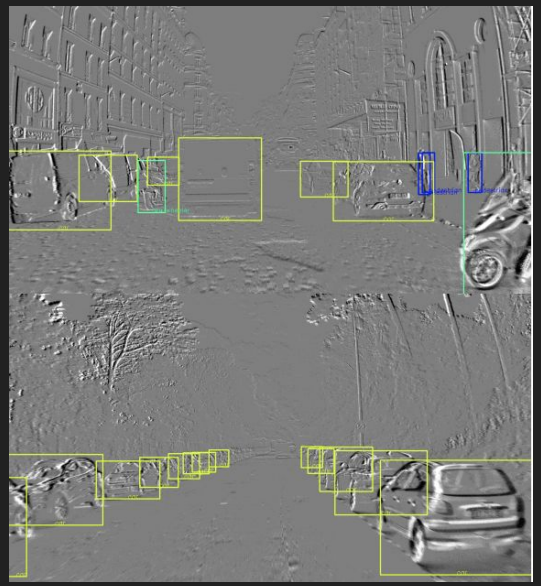


Event Tensors:

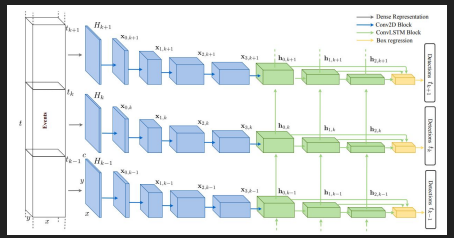
- Event Volumes [5]
- Histograms
- Time Surfaces



backbone heads



RED [3] *(SoTA)*



Our Approach - Tracking by Detection

Inputs

Volume *t*

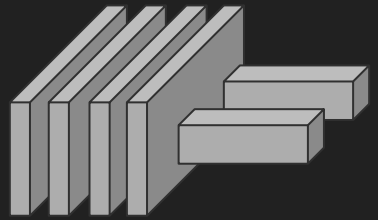


Volume *t-1*



Tracks *t-1*





CenterTrack [4]

Outputs

Tracks *t*



Sizes *t*





Offsets *t*

Event Volumes [5]

$$k_b(a) = \max(0, 1 - |a|)$$

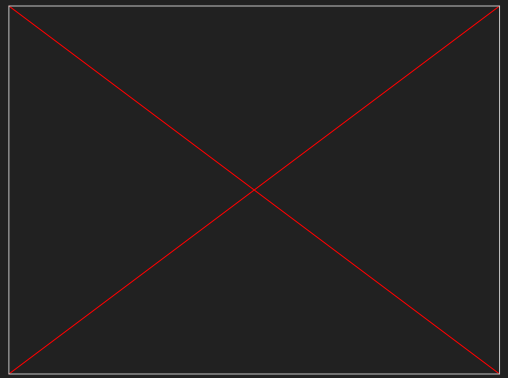
$$t_i^* = (B-1)(t_i - t_0)/(t_N - t_0)$$

$$V(x, y, t) = \sum_i p_i k_b(t - t_{i^*})$$

Current Results & Future Work

Results:

- Optimized Event Volume Data Loading - From 4 to 0.001 seconds
- Currently, training is not converging
- Labels are incredibly sparse for the event data - taken every 5 seconds
    - Doesn’t allow to learn tracking since we need adjacent labels



Future Work:

- Balance positive and negative samples to 50/50
- Trim videos to only use the event data that has labels
- Sample from random videos to make a diverse batch

Citations

[1] E. Mueggler, B. Huber, and D. Scaramuzza, *Event-based, 6-DOF Pose Tracking for High-Speed Maneuvers*. IROS 2014

[2] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza. *High speed and high dynamic range video with an event camera*. IEEE Transactions on Pattern Analysis and Machine Intelligence 2019.

[3] E. Perot, P. Tournemire, D. Nitti, J.Masci, and A. Sironi. *Learning to Detect Objects with a 1 Megapixel Event Camera*. NeurIPS 2020.

[4] X. Zhou, V. Koltun, and P. Krähenbühl. *Tracking Objects as Points*. ECCV 2020.

[5] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, *Unsupervised event-based learning of optical flow, depth, and egomotion.* CVPR 2019.

Distortion-Free Wide-Angle Portraits on Camera Phones

Ji Liu, Zhuoqian Yang

12/17/2020

Introduction and Method Overview

Introduction

1. Modern mobile phones are equipped with wide angle cameras
2. Wide angle cameras introduce artifacts caused by perspective projection
3. An algorithm is needed to do correction on-site immediately after photo taking





Introduction

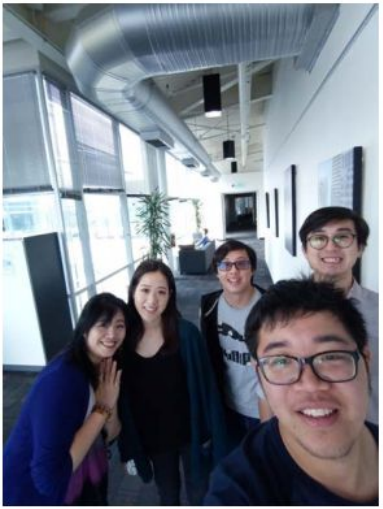
Before correction



After correction



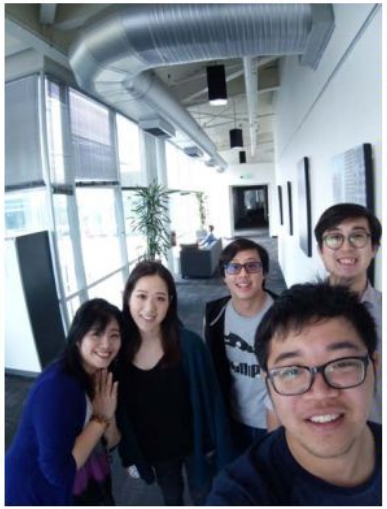
Method Overview



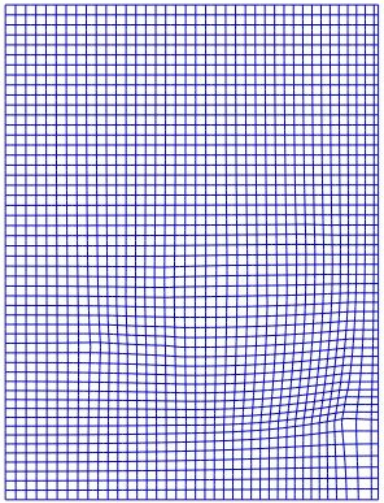
(a) Input (97° FOV)



(b) Subject mask of (a)



(c) Stereographic projection



(d) Optimized mesh



(e) Distortion-free output

Results on Images of Our Own

Naive Stereographic Result



input



flow



naive stereographic

Subject Mask Segmentation Result



input



face mask

Correction Result



Before correction



After correction

Method - Subject Mask Segmentation

1. Use Detectron 2 to extract person mask
2. Use Dlib to extract face bounding box
3. Intersect the two to get face mask







Method - Stereographic Projection

Stereographic projection can picture the 3D world onto a 2D plane with minimal conformal distortion, at the expense of changing the curvature of long lines.

We enforce the stereographic projection locally on face regions to correct perspective distortion.

Given the camera focal length f , we compute the stereographic projection from the input using a radial mapping:

$$r_u = r_0 \tan\left(0.5 \arctan\left(\frac{r_p}{f}\right)\right) \quad r_0 = \frac{d}{2\tan(0.5\arctan(\frac{d}{2f}))} \quad d = \min(W, H)$$

Method - Mesh Placement



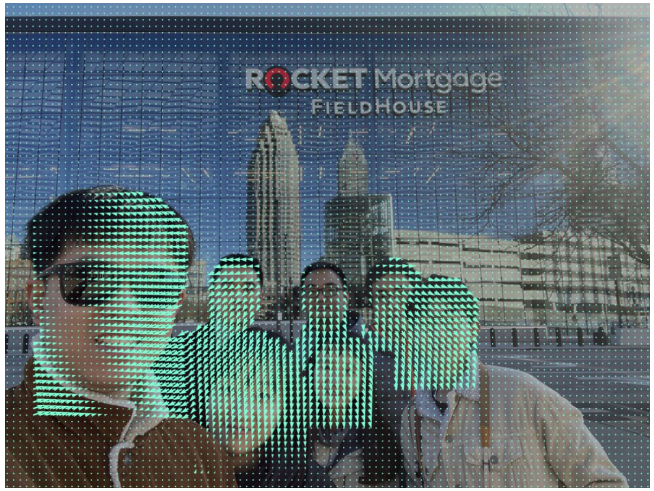
(a) Input (97° FOV) (b) Inset of (a) (c) Naive blending

$$w_i = \begin{cases} 0 & \text{if } \mathbf{p}_i \notin \text{face mask}, \\ 1 & \text{if } \mathbf{p}_i \in \text{face mask}. \end{cases}$$





Before and after naive stereographic, we can observe strong artifacts around the face regions.





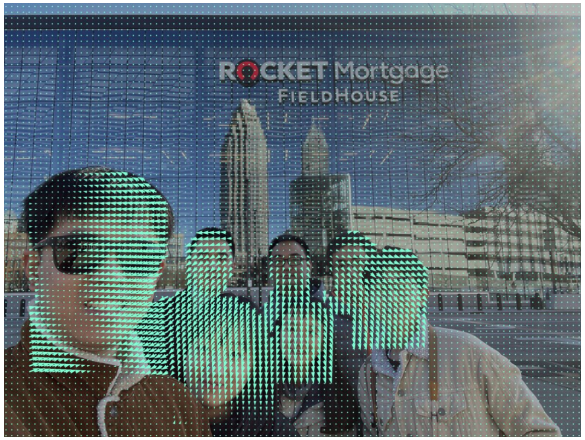
Method - Local Face Undistortion

We minimize the following energy function to determine an optimal mesh.

$$\{\mathbf{v}_i^*\} = \underset{\{\mathbf{v}_i\}}{\operatorname{argmin}} E_t(\{\mathbf{v}_i\})$$

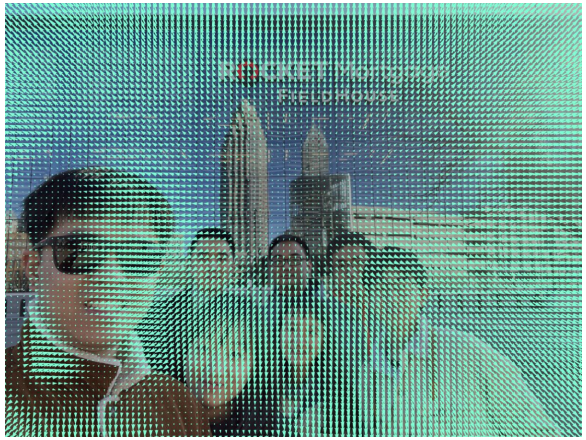
where Et is the weighted sum of several energy terms including Face Objective Term, Line-Bending Term and Regularization Term.

Method - Regularization Term



$$E_r = \sum_i \sum_{j \in N(i)} \|\mathbf{v}_i - \mathbf{v}_j\|_2^2$$

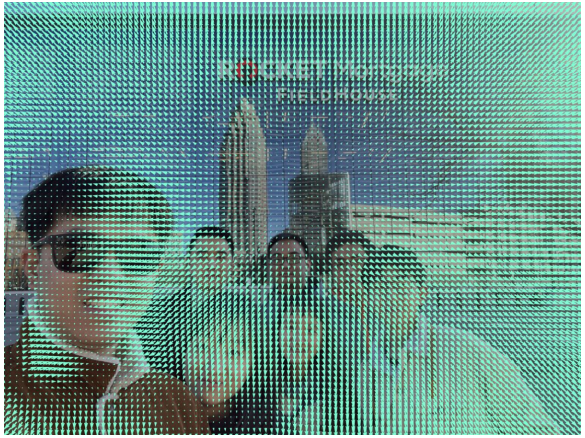
we regularize the mesh by encouraging smoothness between 4-way adjacent vertices using a regularization term







Method - Line-Bending Term



$$E_b = \sum_i \sum_{j \in N(i)} \|(\mathbf{v}_i - \mathbf{v}_j) \times \mathbf{e}_{ij}\|_2^2$$

On the boundary between the face and background, straight lines may be distorted because the two regions follow different projections. We preserve straight lines by encouraging the output mesh to scale rather than twist by adding a line-bending term. The line-bending term penalizes the shearing of the grid, and therefore preserves the edge structure on the background





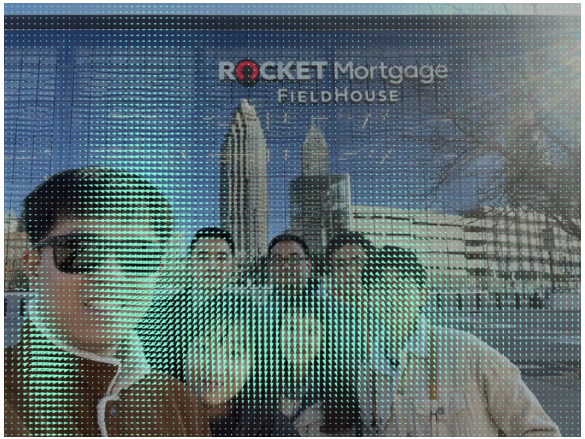


Method - Mesh Boundary Extension



Similar to gradient-domain image processing, a simple boundary condition by forcing vi = pi on the mesh boundary would do the job. However, it creates strong distortions when faces are close to image boundary.

$$\begin{cases} v_{i,x} = p_{i,x}, & \text{if } i \in \text{left or right boundary,} \\ v_{i,y} = p_{i,y}, & \text{if } i \in \text{top or bottom boundary.} \end{cases}$$





$$E_a = E_l + E_r + E_t + E_b ,$$

$$E_l = \mathbb{I}(v_{i,x} > 0) \cdot \|v_{i,x}\|_2^2, \forall i \in \partial_{\text{left}} ,$$

$$E_r = \mathbb{I}(v_{i,x} < W) \cdot \|v_{i,x} - W\|_2^2, \forall i \in \partial_{\text{right}} ,$$

$$E_t = \mathbb{I}(v_{i,y} > 0) \cdot \|v_{i,y}\|_2^2, \forall i \in \partial_{\text{top}} ,$$

$$E_b = \mathbb{I}(v_{i,y} < H) \cdot \|v_{i,y} - H\|_2^2, \forall i \in \partial_{\text{bottom}} ,$$

This distributes the distortion to padded vertices, and reduces artifacts near the boundary of the output image.



Method - Similarity Constraint

$$E_{s,k} = \sum_{i \in \mathbf{B}_k} w_i m_i \|\mathbf{v}_i - (\mathbf{S}_k \mathbf{u}_i + \mathbf{t}_k)\|_2^2 + \lambda(\mathbf{S}_k)$$



$$E_f = \sum_k E_{s,k}$$

$$\mathbf{S}_k = \begin{bmatrix} a_k & b_k \\ -b_k & a_k \end{bmatrix}$$





Constrain the transformation around each facial area to be a similarity transformation that preserves scale

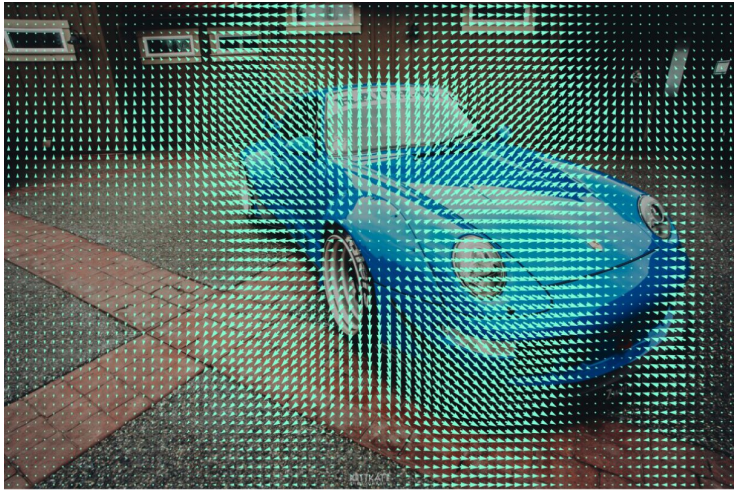


Extension of the Method to Other Object Categories













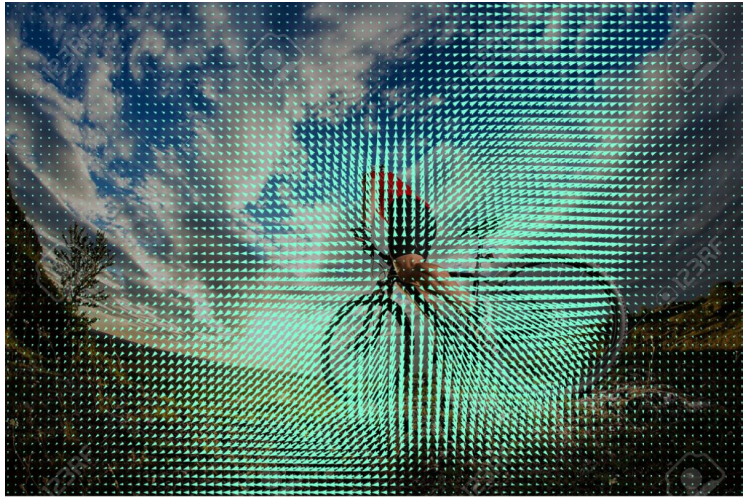












Thank you!

Mirror, Mirror On the Wall: Detecting Mirrors and Correcting Point Clouds

Sachit Mahajan
sachitma

Carnegie Mellon University

Motivation



- Mirrors are common in indoor scenes
- Mirrors appear as doors or entranceways to visual sensors.
- For scene reconstruction- point clouds have to be be manually corrected by giving the mirror boundaries
- Robots performing exploration cannot differentiate between doors and mirrors, and end up planning paths through mirrors or forming incorrect maps



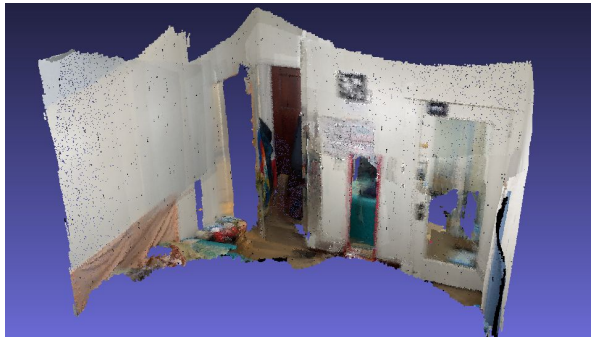


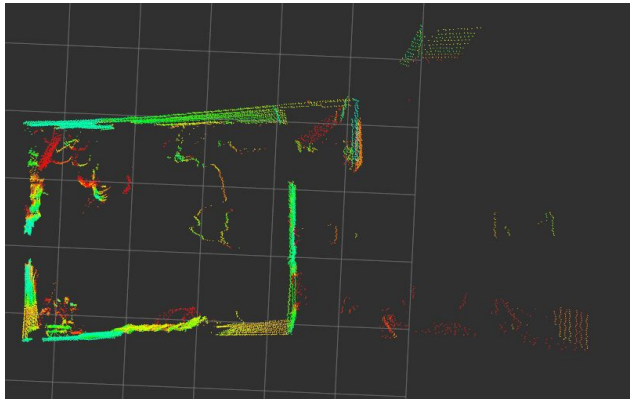
Matterport3D: Learning from RGB-D Data in Indoor Environments

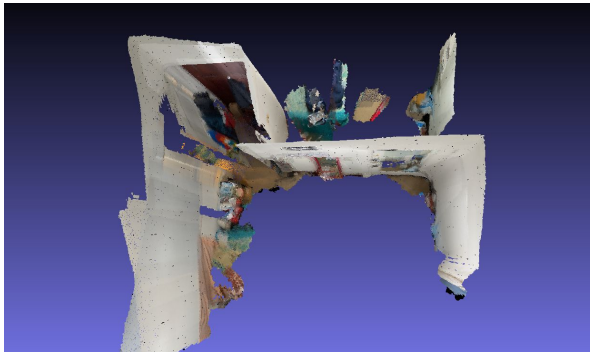
Carnegie Mellon University

Motivation





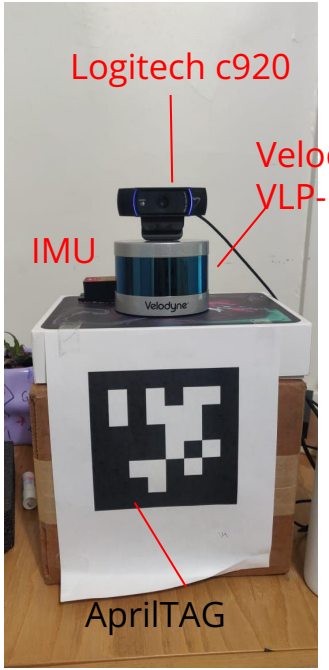


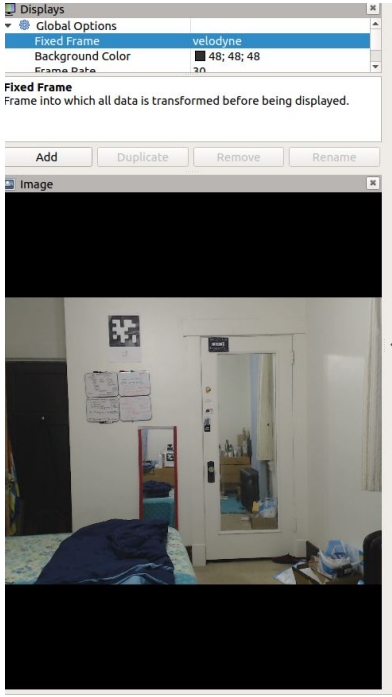


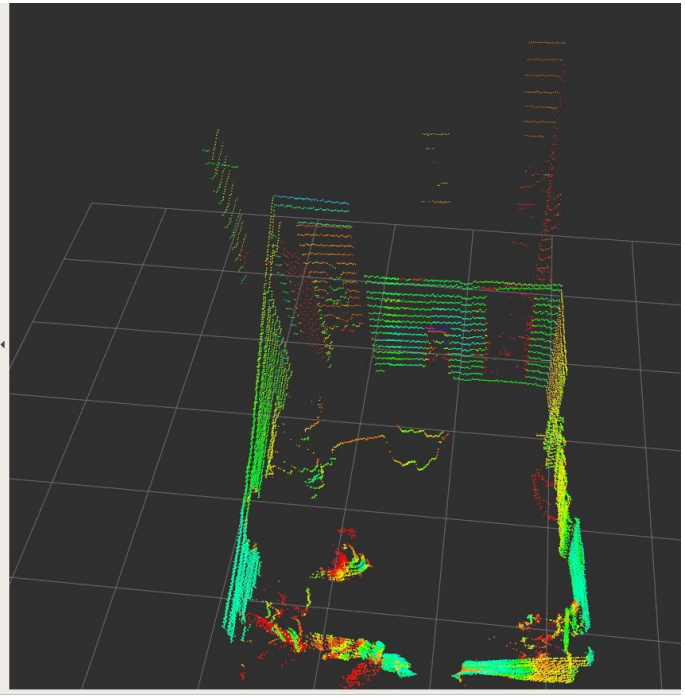
Taken from Phone (camera+Lidar) and ‘3d Scanner App’

Carnegie Mellon University

Setup

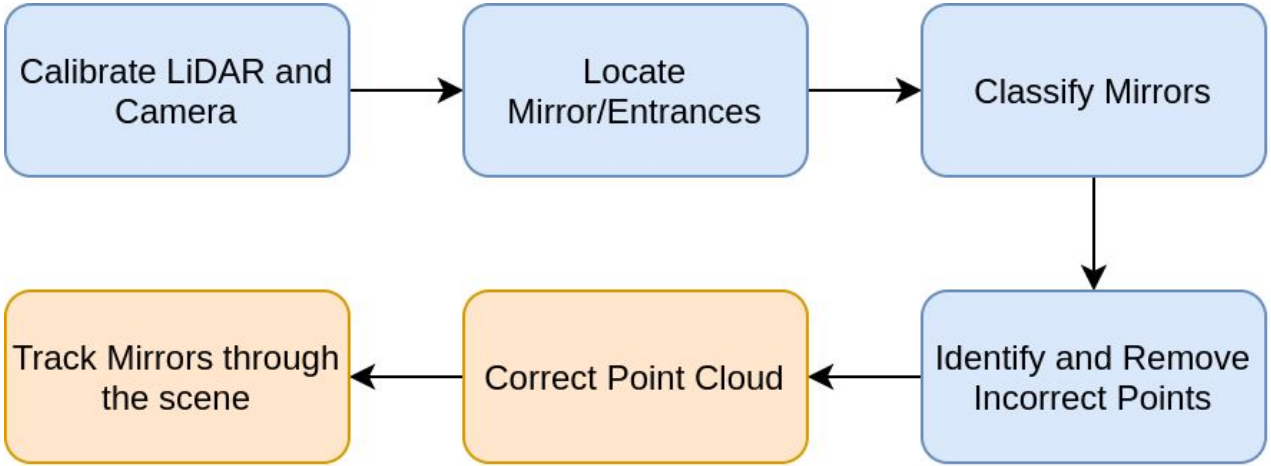




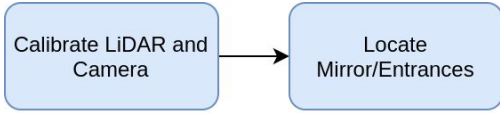


Carnegie Mellon University

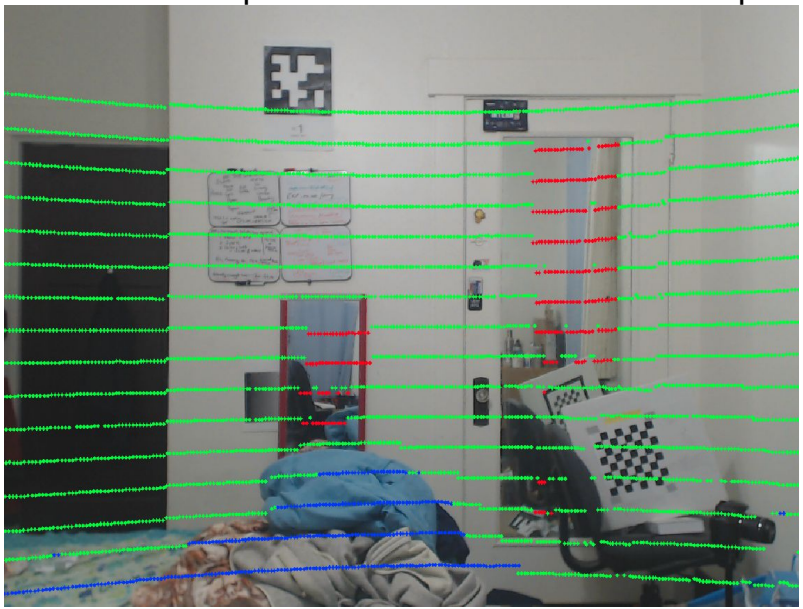
Methodology

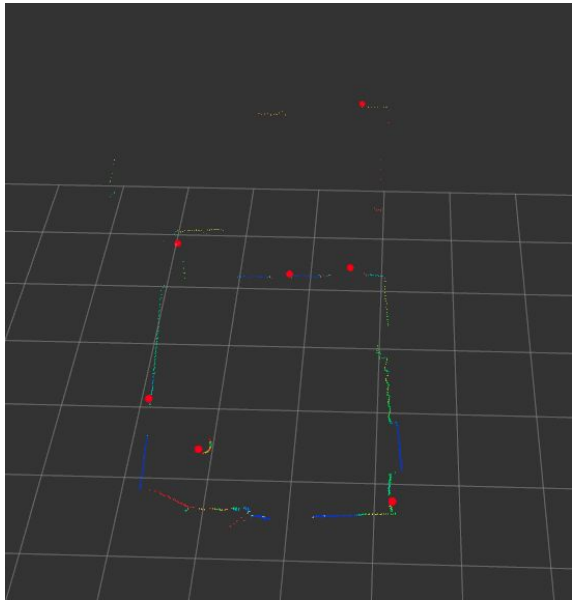


Calibrate Extrinsics and Locate Mirror/Entrances

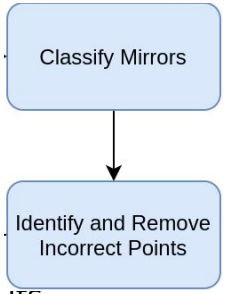


- Preprocess pointcloud: Median Filtering + Bilateral Filtering
- Use depth discontinuities to detect possible mirrors





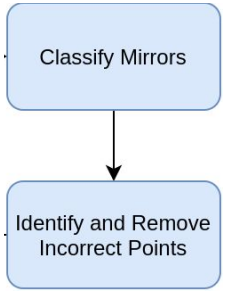
Classify mirrors and Remove Points

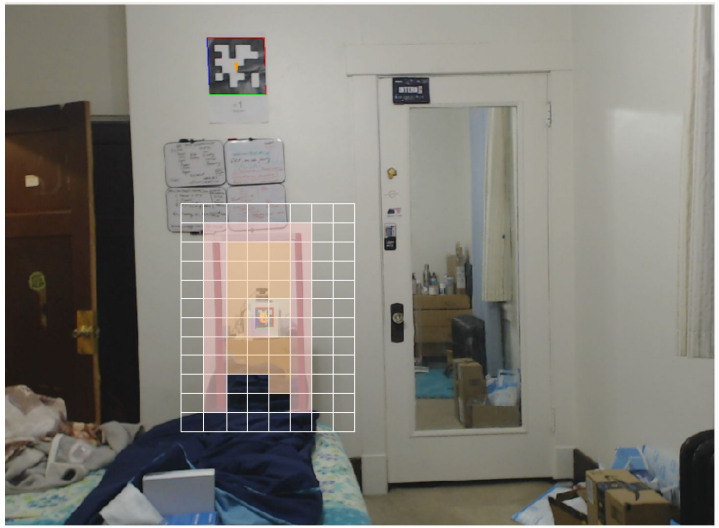


Two different methods

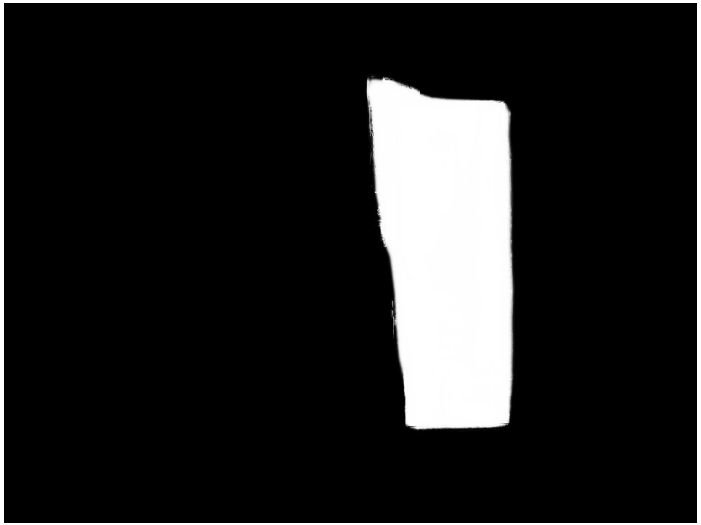
1. Using self-reflection (mirrored fiducial marker) to identify mirror plane
   - Detect the 6dof pose of AprilTag, this gives the location of one point on the mirror plane and its orientation
   - **Novel method:** Divide the image into grids, use a region growing to identify mirror planes. Heuristics used to classify grid as mirror-
     i. Part of AprilTag lies in Grid
     ii. Depth greater than mirror plane and high depth discontinuities
     iii. Grid normals differ from that of mirror plane/ High normal variance (sum of dot products)
     iv. Image intensities have high variance (mirror boundaries)
   - Requires very accurate calibration
2. Segmentation mirrors in images by modeling semantical and low-level color/texture discontinuities -
   - MirrorNet from 'Yang, Xin and Mei, Haiyang and Xu, Ke and Wei, Xiaopeng and Yin, Baocai and Lau, Rynson W.H.Where Is My Mirror?'
   - Works for only one mirror in the scene, fails very often when there is no mirror in the scene

Classify mirrors and Remove Points





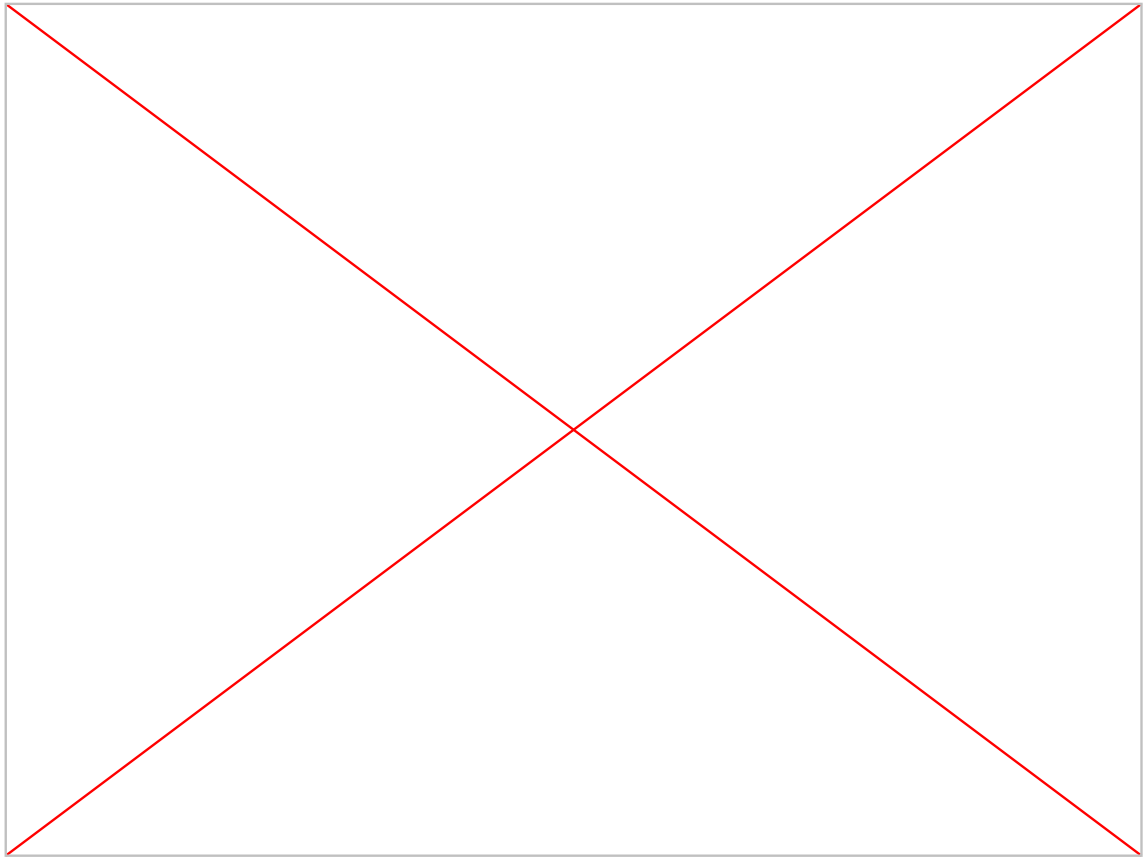
Mask Obtained From Region Growing



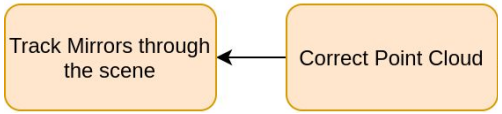
Mask Obtained From MirrorNet

Carnegie Mellon University

Results



Future Work



- Correct the errors in region growing,
- Region growing should be done using a coarse-to-fine approach such that mirror boundaries can be better estimated
- Instead of removing point clouds they can be corrected, since we know our pose and mirror plane we can find which 3D point they actually correspond to and reproject them.
- Once a mirror is found and estimated, it should be tracked through the scene even after the AprilTag is not visible

Image Deblurring Using Inertial Measurement Sensors

Kevin O’Brien
kobrien

Problem Motivation/Description



Blurry images stink! She gets it ---------->

Deconvolution is difficult

-Optimization with no closed-form solution

-Different optimization strategies perform well for some types of scenes, poorly for others

-Generally want to enforce sparse gradients in the image to preserve sharp edges

$$b = c * x + n$$

$$\min_x ||b - c * x||^2 + ||\nabla x||_1^1$$

Blind Deconvolution

Simultaneously solve for the sharp image I and the blur kernel K

- -This is an even harder optimization problem, solving for two things at once
- -Advances in blind deconvolution come from introducing additional constraints
  - -sharp edge prediction, color statistics

MAIN IDEA

When we're dealing with camera shake, use the camera's motion to help inform the blind deconvolution problem

Blur as a Homography

For small camera movements during exposure, blurring can be parametrized as small planar warps

$$I_t(u_t, v_t) = I(H_t(d)(u_0, v_0, 1)^T)$$

$$\vec{B} = \int_0^s \left[A_t(d)\vec{I}dt \right]$$

$$||\vec{B} - A(d)\vec{I}||^2/\sigma^2$$

Sensor Data Processing

Accelerometer readings give linear acceleration (m/s^2)
Gyroscope readings give rotational velocity (rad/s)

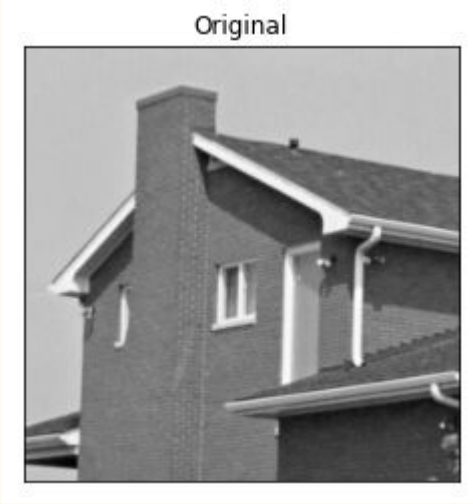
Procedure:

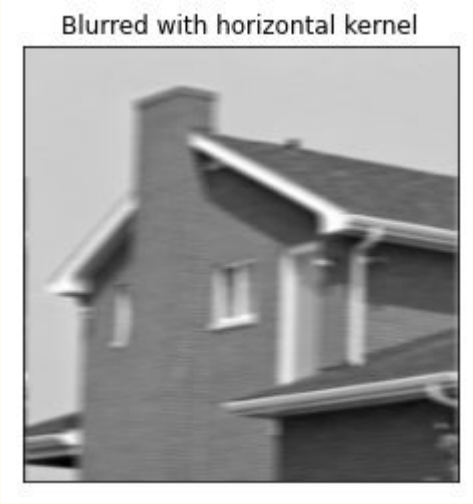
1. Integrate rotational velocity to get angular position, convert to rotation matrix
2. Rotate accelerometer readings into starting reference frame
3. Integrate accelerometer readings to get relative translations
4. Use recovered rotations/translations for homography calculation

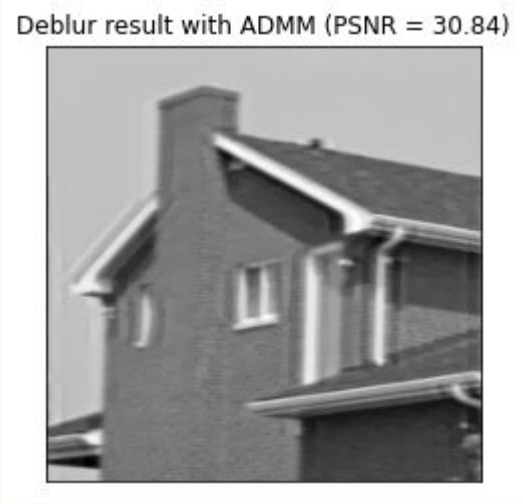
$$H_t(d) = [K(R_t + \frac{1}{d}T_t N^T)K^{-1}]$$

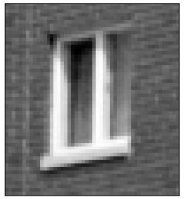
Implementation Status

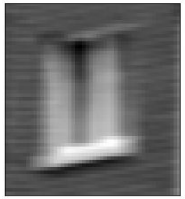
ADMM deblurring optimization with known blur kernel







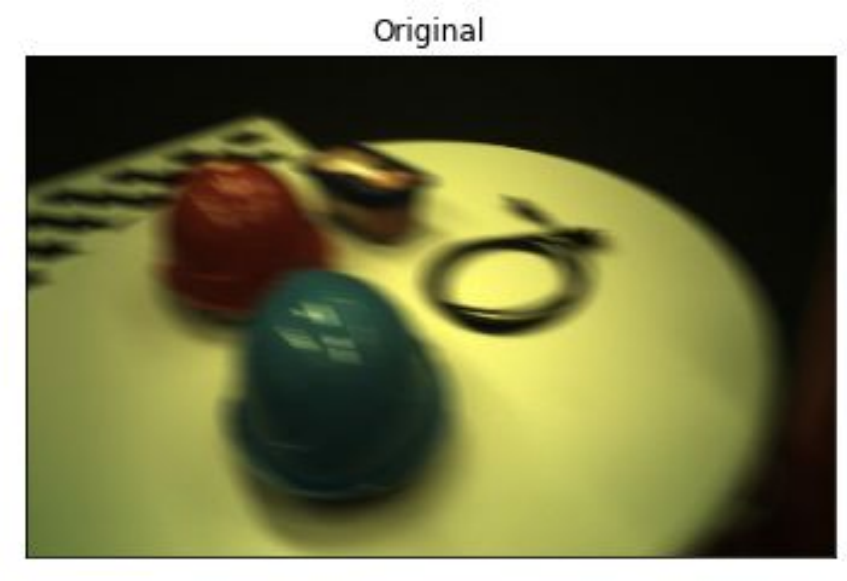


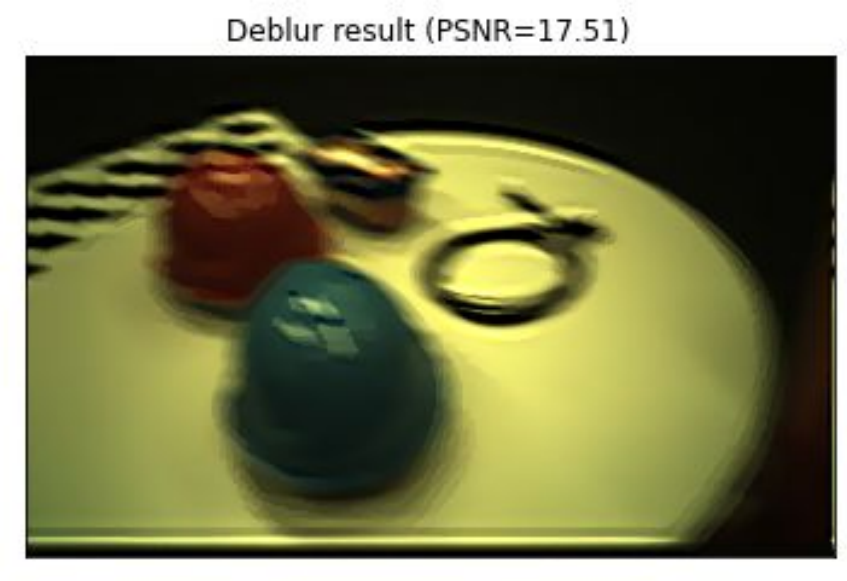




Implementation Status

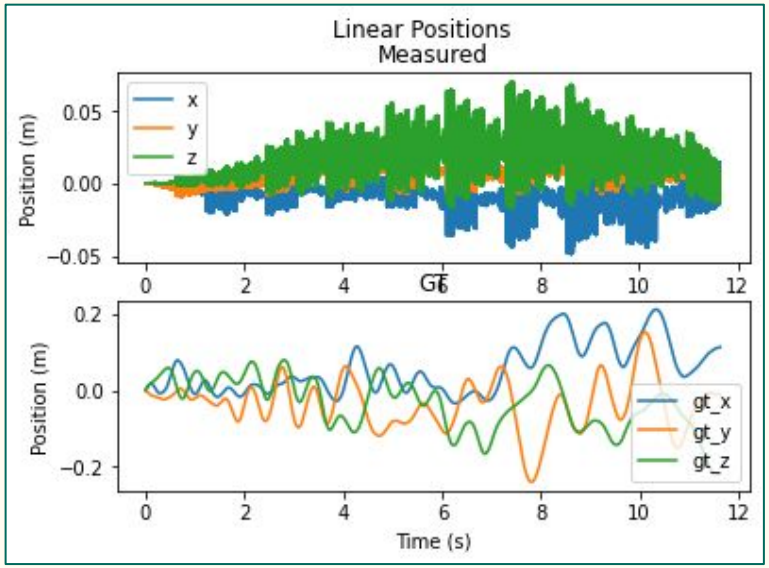
Deblurring using ground truth camera positions

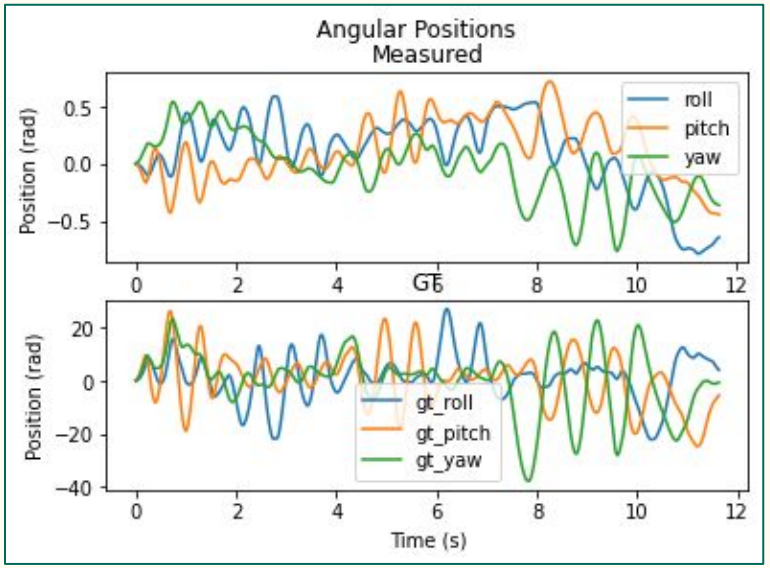




Implementation Status

Use raw sensor values next





Thank you!

Light field 3D reconstruction

Wenxuan Ou (Owen)

Abstract

- Objective: Obtain high resolution depth map from structured light field
- Primary reference:
  - Kim, Zimmer. “Scene Reconstruction from High Spatio-Angular Resolution Light Fields.” ACM transactions on graphics 32.4 (2013): 1–12. Web.
- Other materials:
  - Lin, Yeh. “Depth Estimation for Lytro Images by Adaptive Window Matching on EPI.” Journal of imaging 3.2 (2017): 17–. Web.
- Motivation: Extend the algorithm to structured lightfield (Lytro image), achieve 3D reconstruction in a single-shot.

Algorithm

- Depth estimation from epipolar image (EPI)
  - Depth relationship to disparity:

$$m = \frac{1}{d} = \frac{z}{fb}$$

  - Calculate the depth score of each possible disparity values, select the best estimate:

$$S(u,d) = \frac{1}{|R(u,d)|} \sum_{r \in R(u,d)} K(r - \bar{r})$$

- Refine depth estimation based on confident masking and radiance/color

$$C_e(u,s) = \sum_{u' \in N(u,s)} ||E(u,s) - E(u',s)||^2 \qquad C_d(u,\hat{s}) = C_e(u,\hat{s})||S_{max} - \bar{S}||$$

  - Compute depth only at confident locations, thresholding

Reference: Kim et al

Algorithm

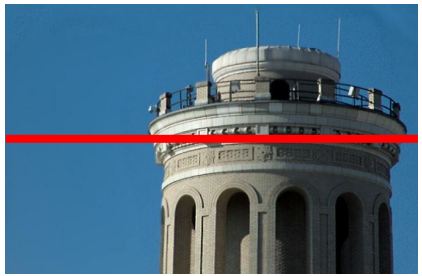
- Obtain EPI from Lytro image
  - Does not need rectification



Figure 2. The epipolar plane images. s and t denote the horizontal and vertical position within the multiple view image sets. With fixed t and spatial position y (which is inferred by the red line), we can get a slice representing the shift of a horizontal line from different views. We call this the *horizontal epipolar plane image (EPI)*. The green line infers the *vertical EPI*.

Reference: Lin et al

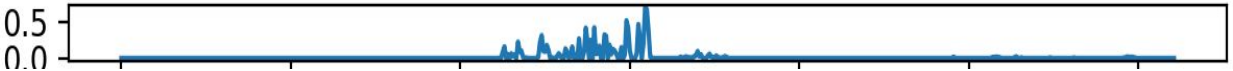
Results



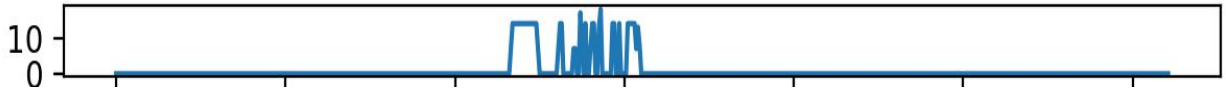
Original image



EPI

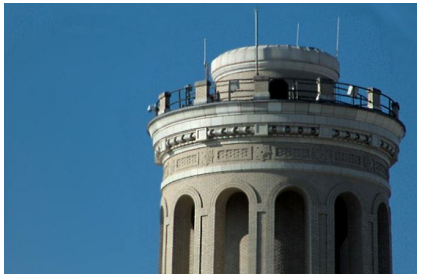


Confidence



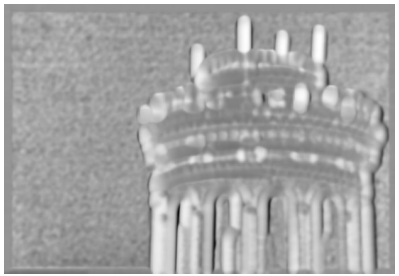
Disparity

Results



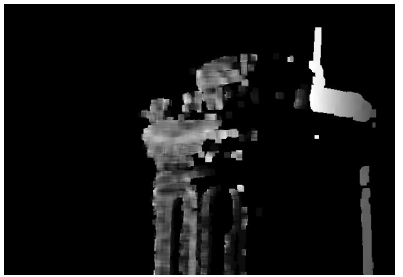


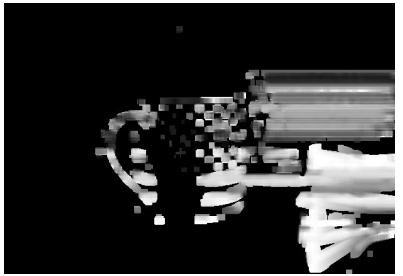
Original image





AFI depth map





My depth map

Discussion and conclusion

$$m = \frac{1}{d} = \frac{z}{fb}$$

- Lytro image does not have large enough disparity to generate reliable depth estimation. Need a larger camera transition to capture fine details.
- Depth from EPI is strongly influenced by texture of the objects. Homogeneous region leads to ambiguous depth score.
- Future extension:
    - The homogeneous problem can be potentially solved by doing a further step fine-to-coarse refinement.



Thank you !

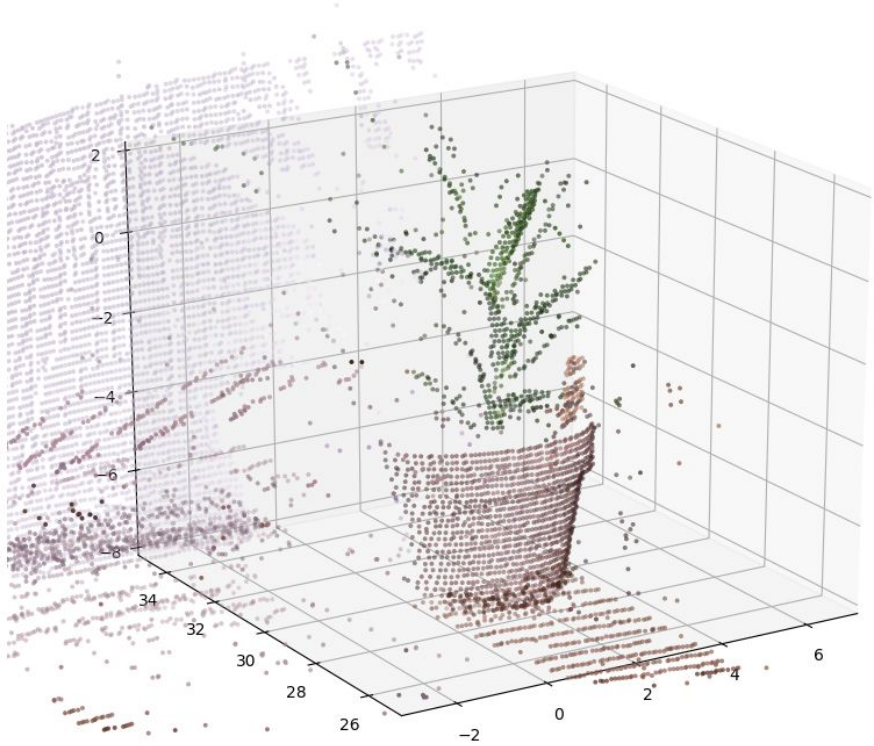
3D Scanning with Structured Light

George Ralph

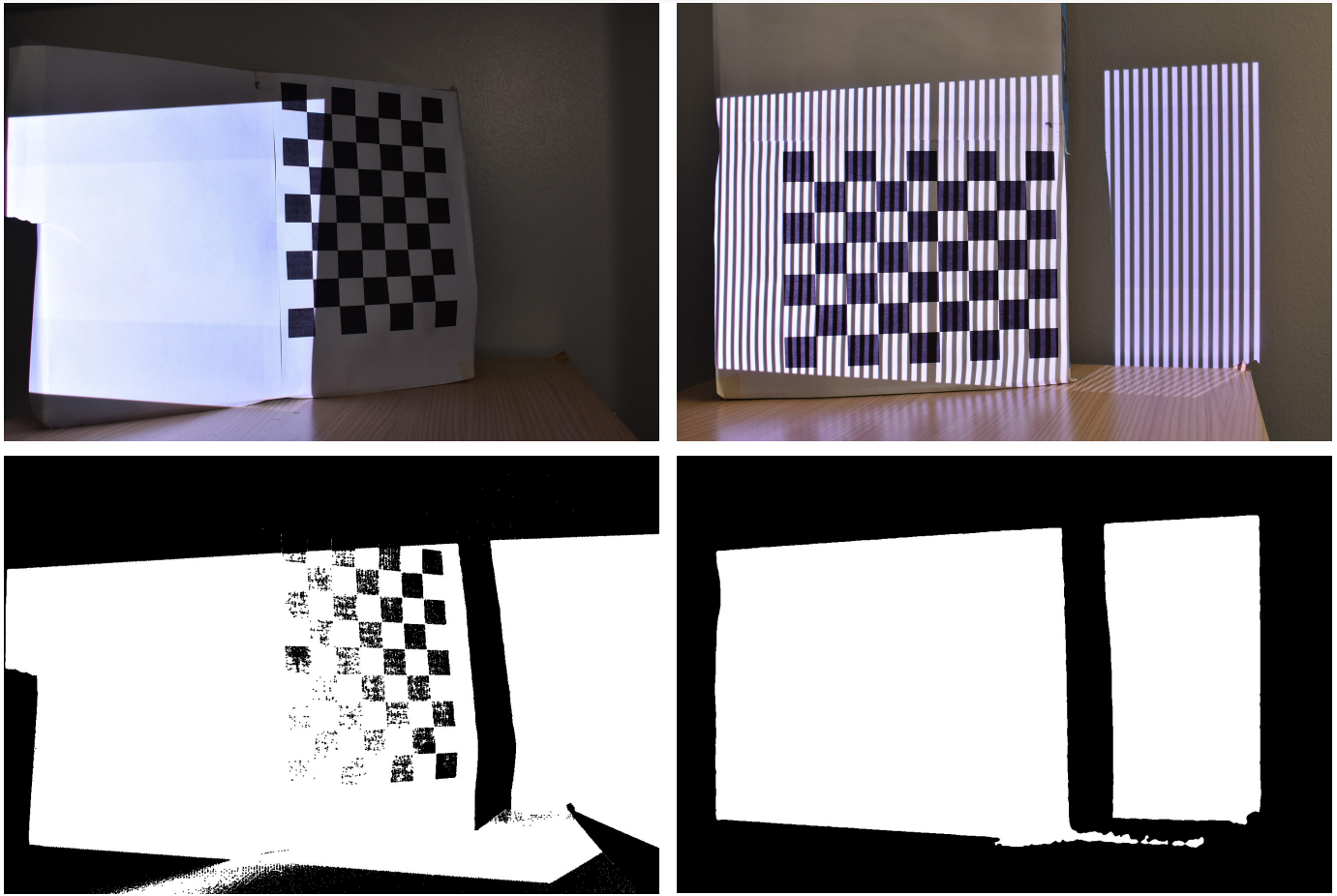
Background

- Structured light reconstruction is dependent on accurate pixel correspondences

- Global Illumination
- Defocus

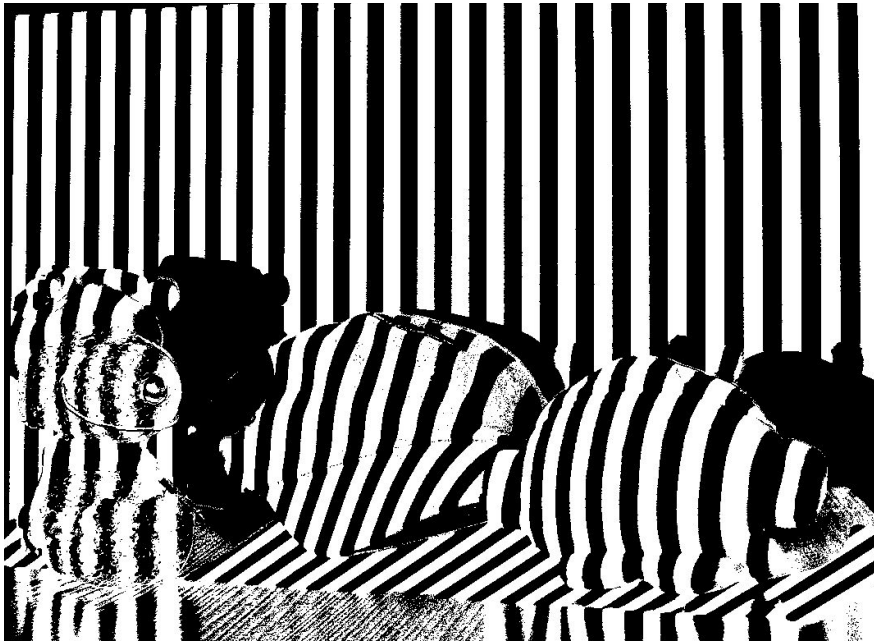


Reducing Global Illumination Effects

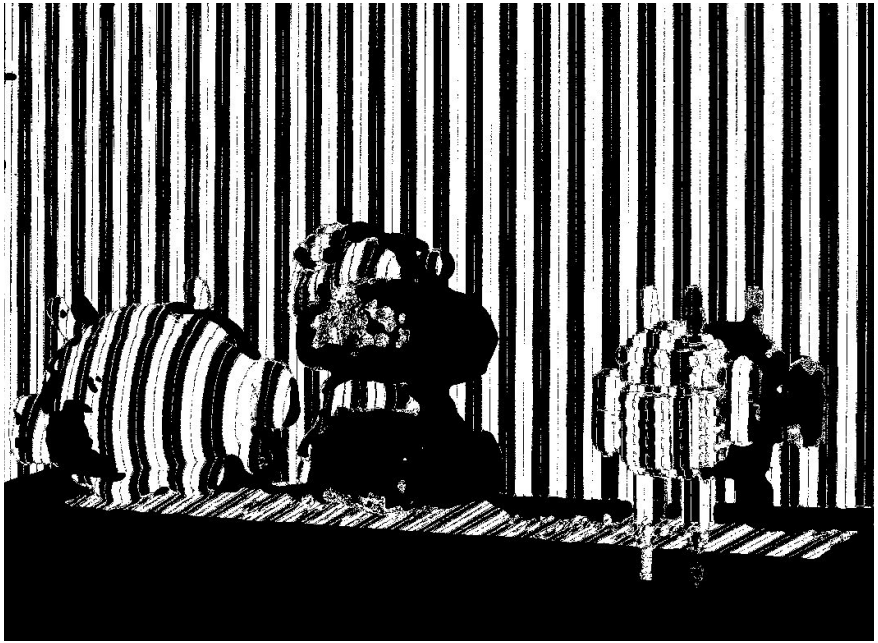


Pattern Binarization

Two-Image Binary Codes

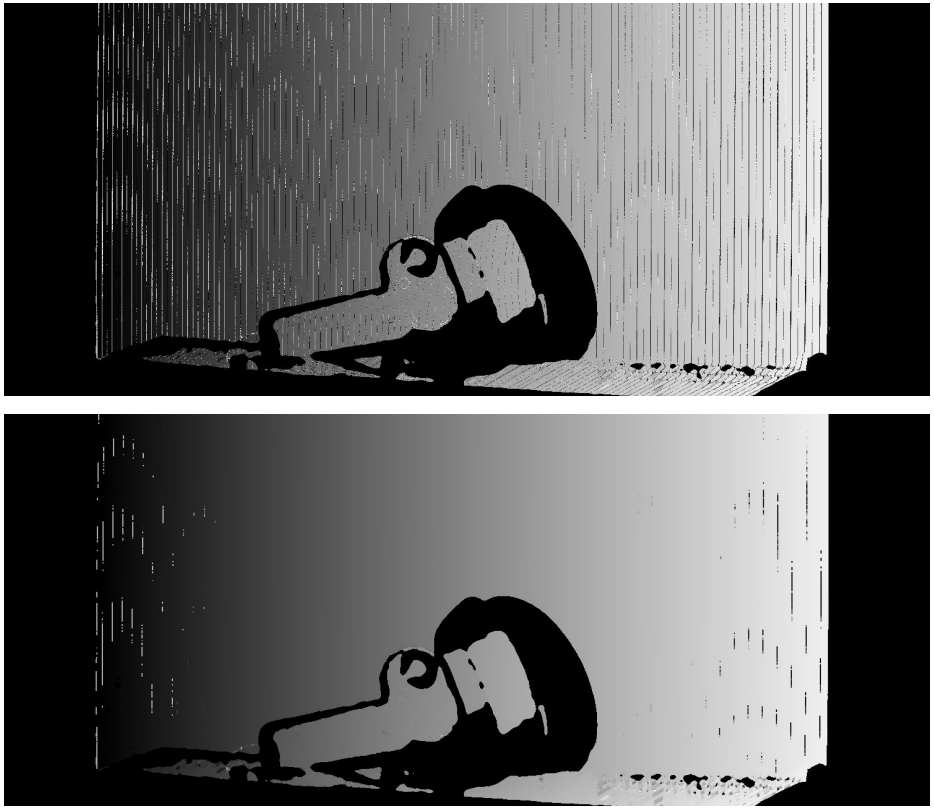


One-Image XOR Gray Codes

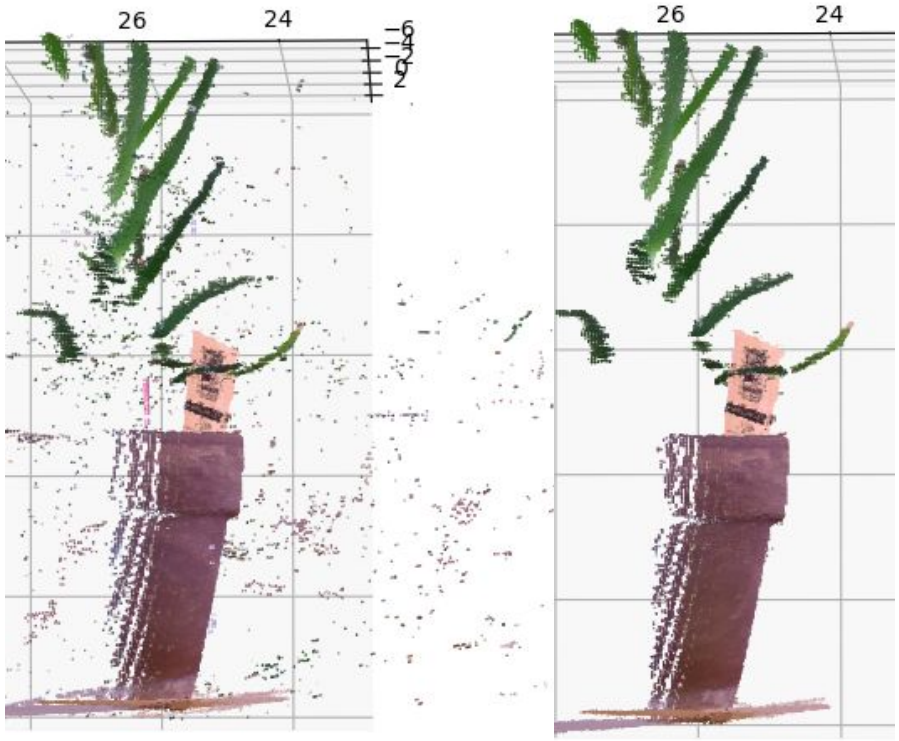


Other Filtering Techniques

Median Filtering of Correspondences



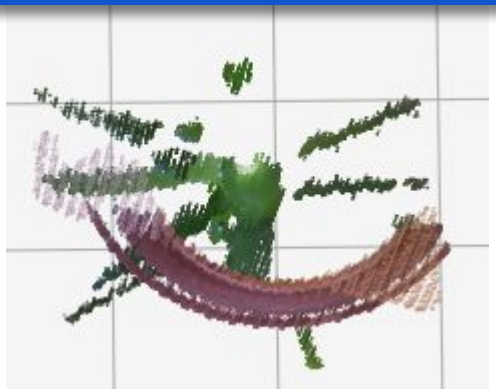
Point Cloud Filtering



Examples



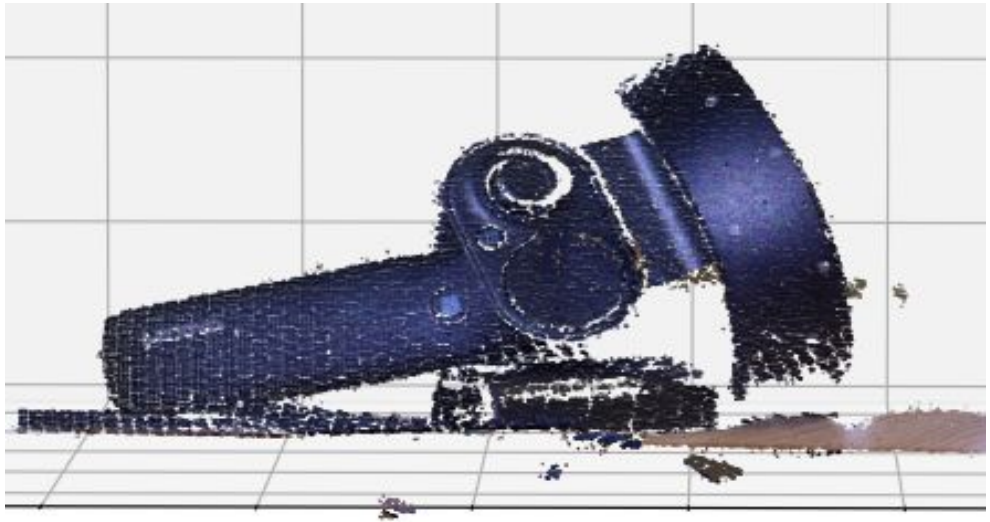


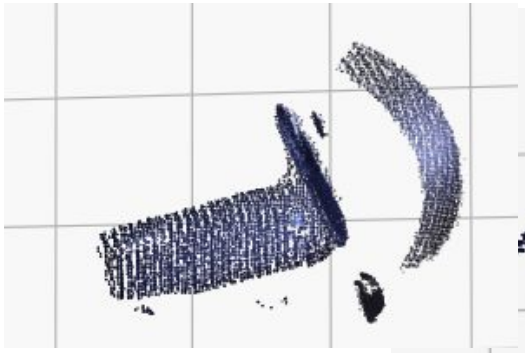


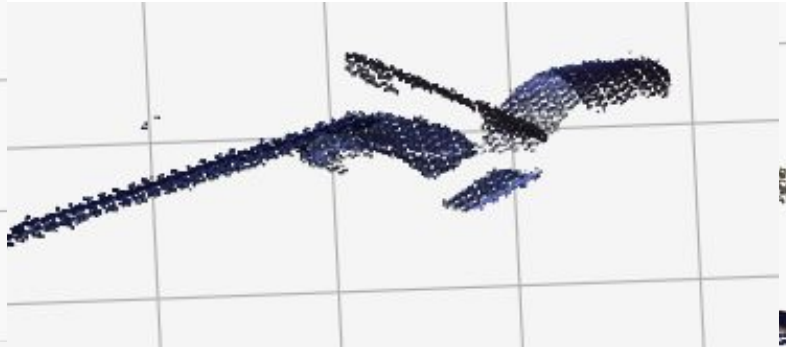


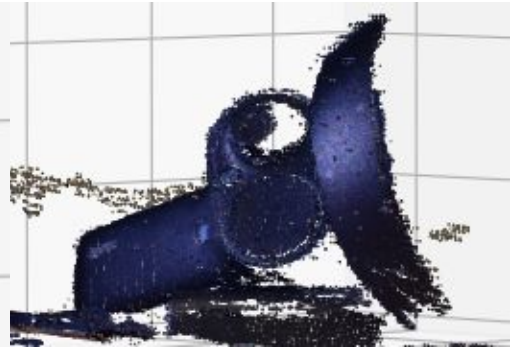
Examples





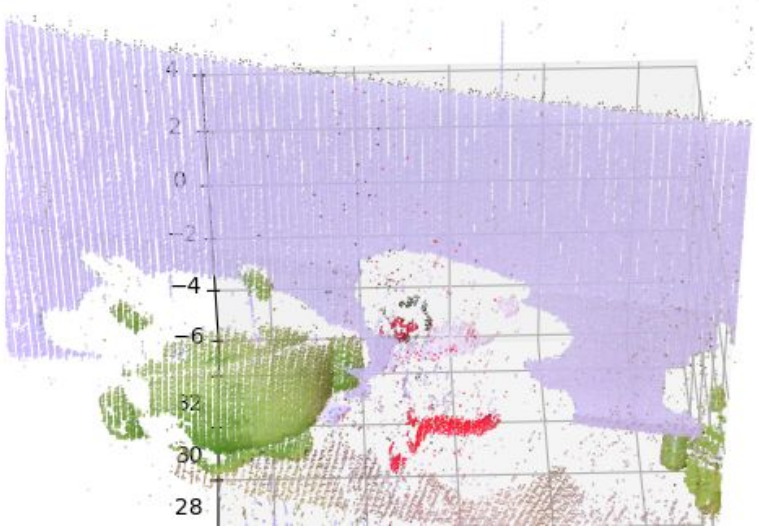


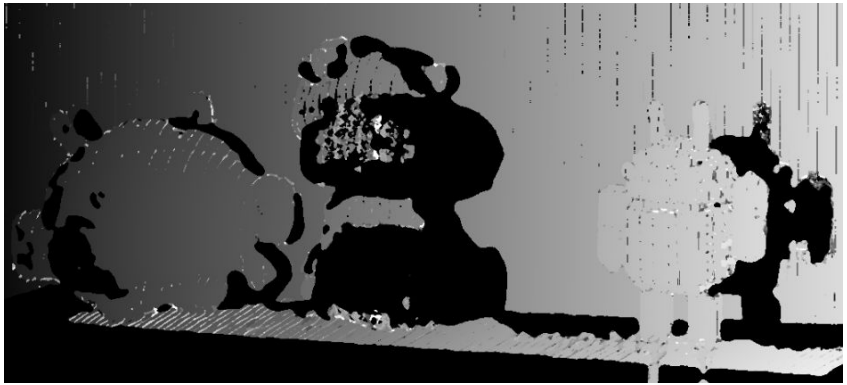


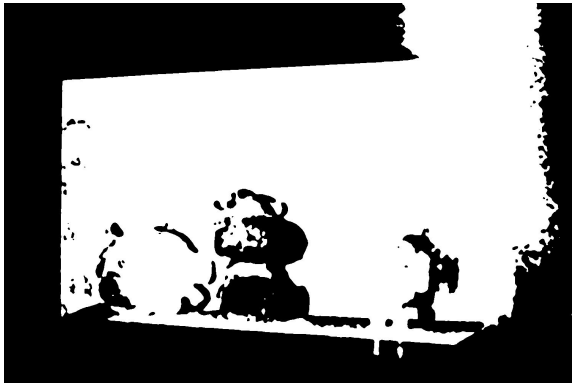


Examples





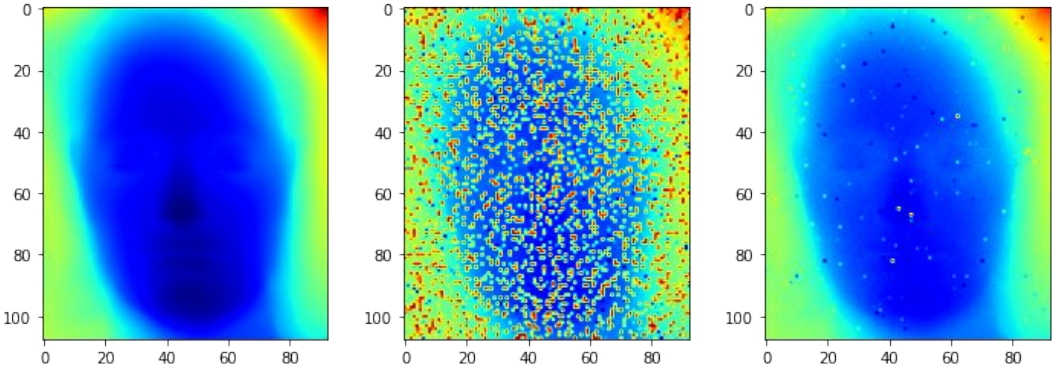




Thank You!

Feel free to ask questions.

Adaptive SPAD Imaging with Depth Priors



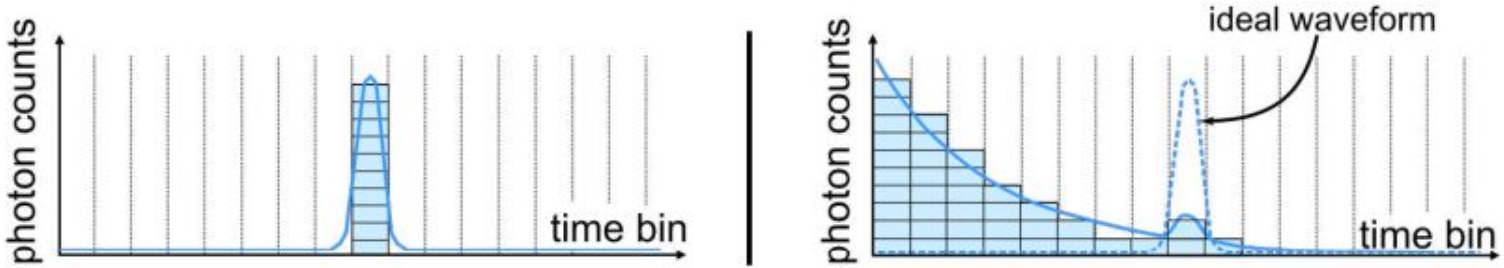
Po Ryan

Background

- High temporal resolution
- Only records first arriving photon
- Performs fine under no ambient light
- Pile-up effect under ambient light
  - Coates' Estimate used for pile-up compensation
  - Earlier bins carry more information
  - Earlier bins have better estimates

SPAD Sensor

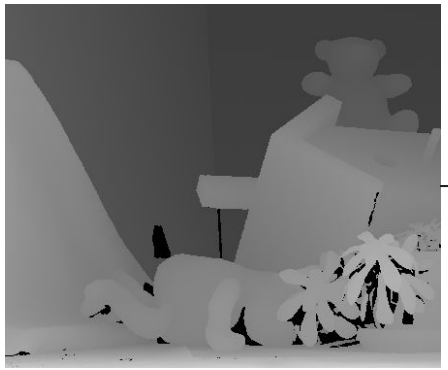




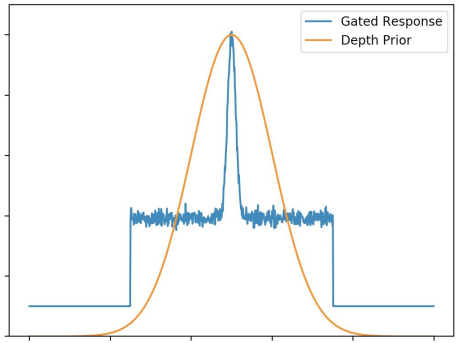
[Gupta et al. "Photon-Flooded Single-Photon 3D Cameras" CVPR 2019]

Method

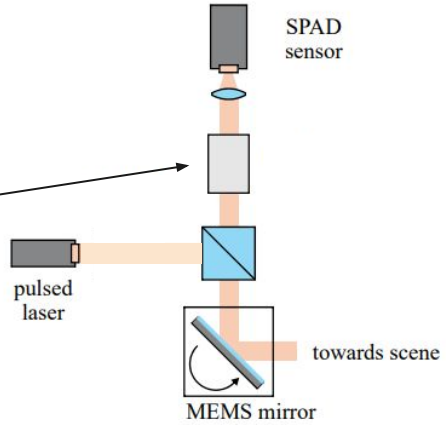
Depth Estimate



Establish Depth Prior



Adjust Gate and Active Time of SPAD



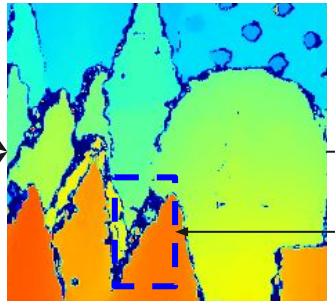
Stereo Depth Prior

Stereo Pairs

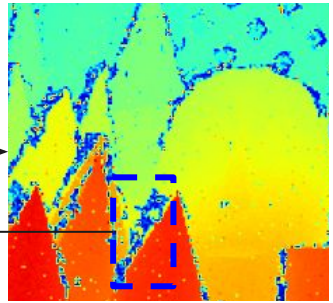




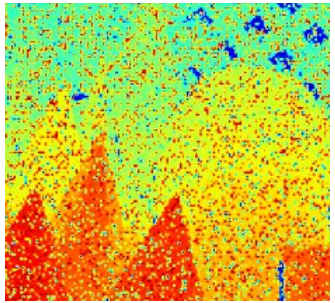
Depth Estimate



Adaptive SPAD



Conventional SPAD



- ❏ Poor estimates for occluded pixels
- ❏ Certainty of estimate unknown

NN-Based Probabilistic Single RGB Depth

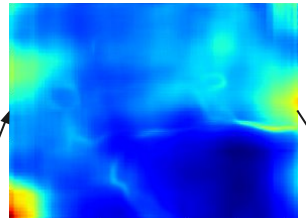
Single RGB Image



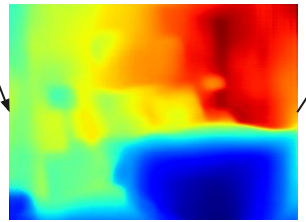
[Xia et al. "Generating and Exploiting Probabilistic Monocular Depth Estimates" CVPR 2020]

CNN

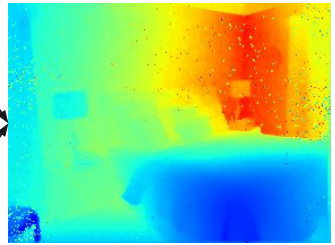
Uncertainty



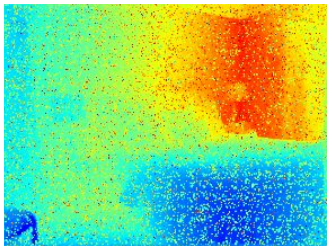
Depth Estimate



Adaptive SPAD



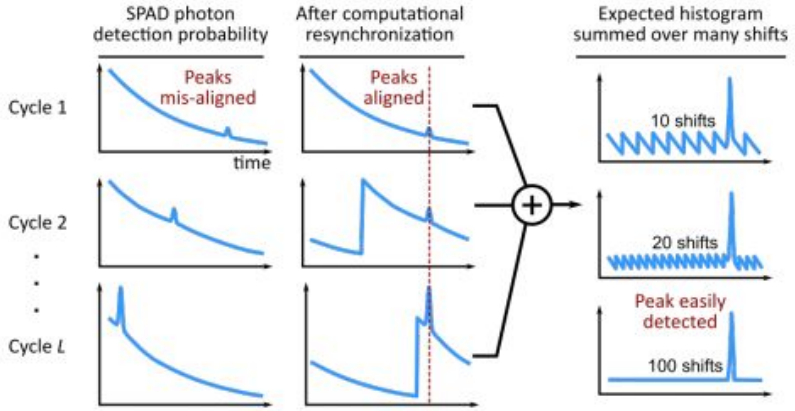
Conventional SPAD



- ❏ Estimates are probabilistic
- ❏ Shifting of gate and active time can be determined through uncertainty

Application to Async. Acquisition

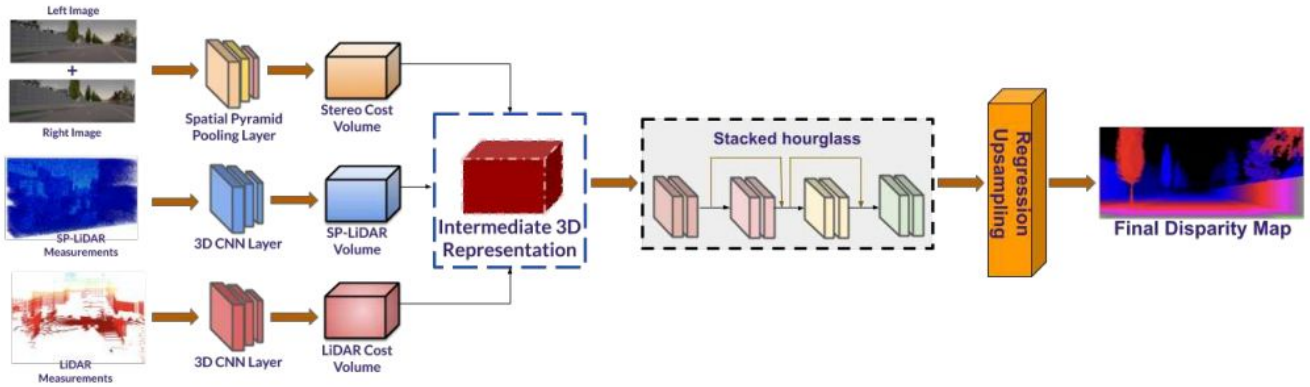
- Asynchronous Acquisition
  - Uniform shifting of gate
  - Every bin is near beginning of transient for at least one shift
- Depth prior can be utilized depth prior to determine shifting pattern instead of relying on uniform shifting schemes
- Depth prior based coding schemes yield at least 20% increase in accuracy



[Gupta et al. "Asynchronous Single-Photon 3D Imaging"]

Integration with NN-Based Pipelines

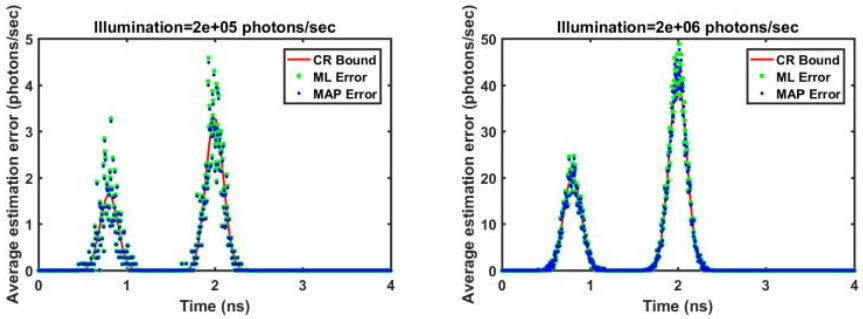
- Stand-alone adaptive SPAD may not give the best final measurements, but improves ***raw*** SPAD measurements
- Better ***raw*** measurements means that other multi-sensor fusion methods would also perform better



[Siddiqui et al. "An Extensible Multi-Sensor Fusion Framework for 3D Imaging" CVPR 2020]

SPAD Only Adaptation?

- Per bin probability of error can be estimated
  - CR-bound for variance of Coates' estimate
  - Chernoff Bound on probability of error for each bin
- Per bin probability is quite uniformly distributed making adaptive gating very ill-posed



[Pediredla et al. "Signal Processing Based Pile-up Compensation for Gated Single-Photon Avalanche Diodes"]

Thanks!

Carnegie Mellon University

Computational Photography Project CMU-15663

Tejas Zodage , Harsh Sharma

Handheld Photometric Stereo

Motivation

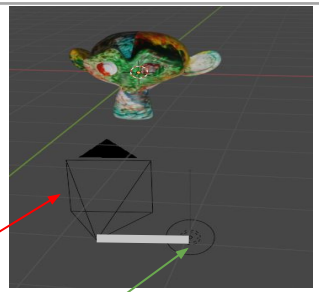
To develop a simple device for dense 3D reconstruction of static scene.

Problem definition

A set of images taken by camera with a fixed Point light source attached to it.

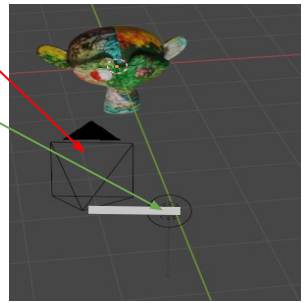
Add photometric constraints to multi-view stereo to get a dense depth map.

Problem definition



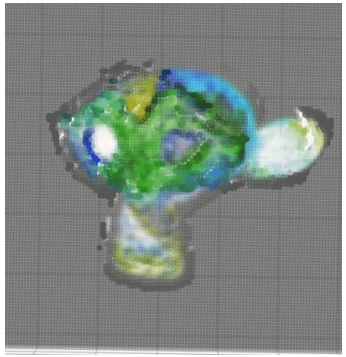
Camera

Point light source rigidly attached to the camera



Input: *m* images

algorithm



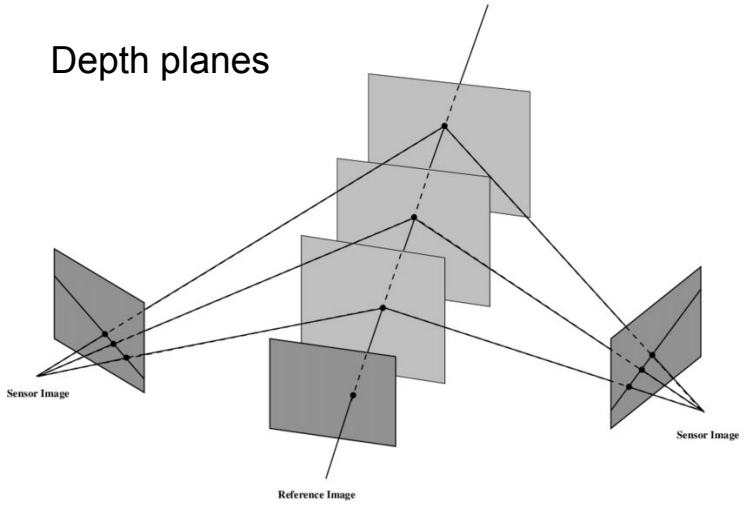
Output: depth image

Procedure

1. Use images from multiple views to **get a per-pixel depth map estimate**, using plane sweep stereo
2. **Get estimates for normals, albedos and ambient lighting** using RANSAC near-light photometric stereo.
3. **Refine the depth, normal and albedo estimates** by formulating an optimization problem with the photometric and depth smoothness costs.

Initial estimates

1. Plane sweep stereo for per pixel depth estimate-
   a. Choose a reference frame and warp images.
   b. Pixel locations are estimated to be on that plane where their NCC score is maximum.
2. RANSAC based near-light photometric stereo-
   a. Normals and albedos



Optimization

- Photoconsistency cost - account for varying lighting, since the light source is attached to the moving camera.
- Surface normal cost - Preferred depth estimates are those which are consistent with the surface normal estimates.
- Smoothness cost - penalize large discontinuities in the depth map

$$E(p, j, j') = E_p(p, j) + \lambda_n E_n(p, j) + \lambda_s E_s(j, j')$$

Model the problem into a graph-cut optimization problem and solve for the constraints jointly

Can’t we solve it like HW5?

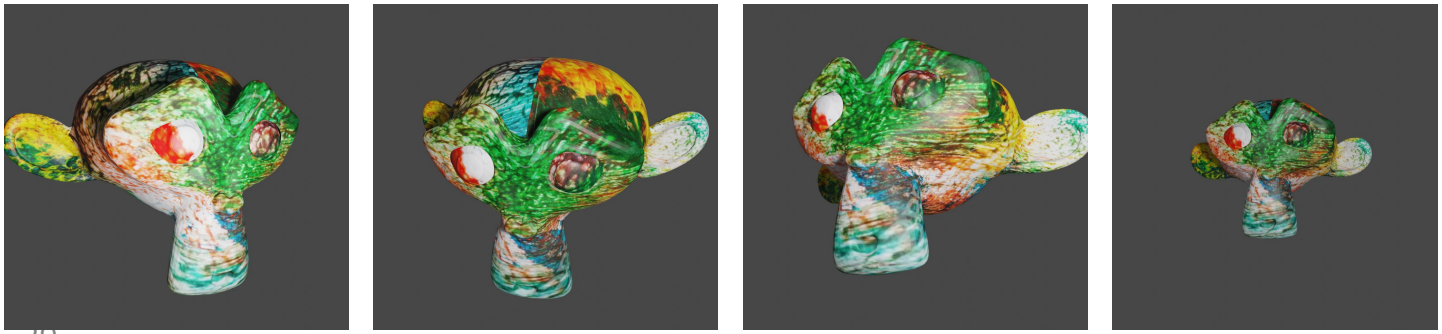
No!

Near-light photometric stereo	**Conventional photometric stereo**
Directional light source	Point light source
Orthographic camera	Perspective camera

Data generation

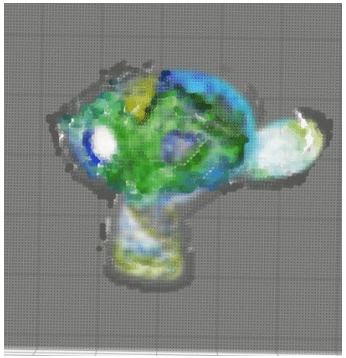
Blender + Mitsuba

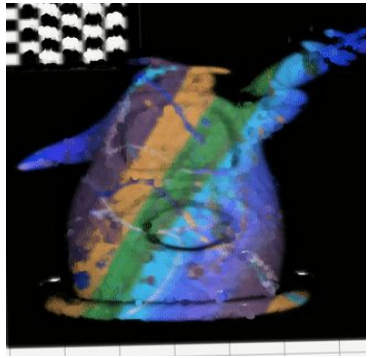
1. Generate scenes using blender (easy for visualization)
2. Render scenes using mitsuba (photorealistic rendering)



Results

Plane sweep stereo - depth map





Challenges

1. **Data generation -**
    a. No readily available dataset
2. **Graph-cut implementation (np hard problem) -**
    a. Suggested algorithm for graph cut by authors - alpha beta swapping algorithm.
    b. Takes hours for optimization on images of size > 100 px X 100 px

Future work

1. **Data generation -**
    a. Real world data.
2. **Graph-cut alternatives -**
    a. Look for better solutions to solve energy minimization on graphs.
    b. Deep learning based methods?

References

1. **Higo, Tomoaki, et al.** "A hand-held photometric stereo camera for 3-d modeling." 2009 IEEE 12th International Conference on Computer Vision. IEEE, 2009.
2. Plane sweeping stereo https://github.com/ReynoldZhao/Stereo/blob/master/src/plane_sweep_stereo.py
3. Graph cuts https://github.com/roflmaostc/Fast-Approximate-Energy-Minimization-via-Graph-Cuts

Image Morphing

Jiazheng Sun

Sample effect







Step 1 - set up

- Load texture (image)
- Coordinates system conversion: world to canvas (cartesian)
  - Move (0, 0) from top left to middle; flip y's direction
    - x / world_size * canvas_size - (canvas_size/2)
    - (canvas_size/2) - y / world_size * canvas_size

Step 2 - build control points

- Define basic anchor points

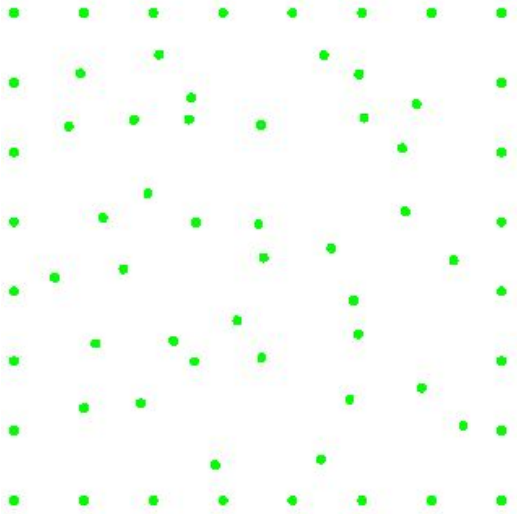


- Define control points and map them to anchor points using Barycentric

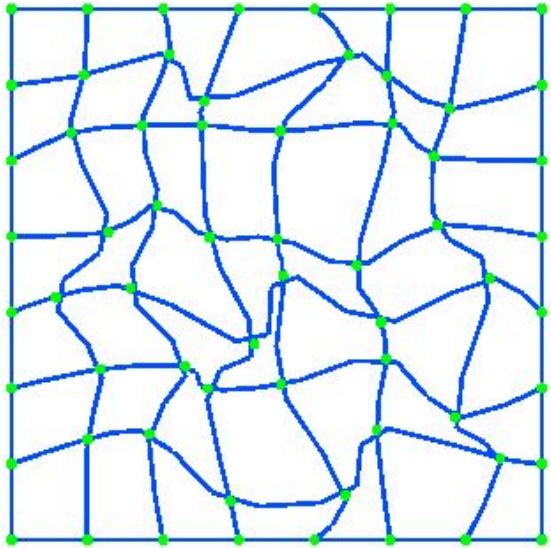


Step 3 - perturb, interpolate & blend

- Perturb the control points (random or preset trajectory)



- Interpolate with curves; blending guarantees C2 continuty at intersection



Curves & Blending

- **Neville interpolation:**

$$p_{i,i}(x) = y_i, \qquad 0 \le i \le n,$$

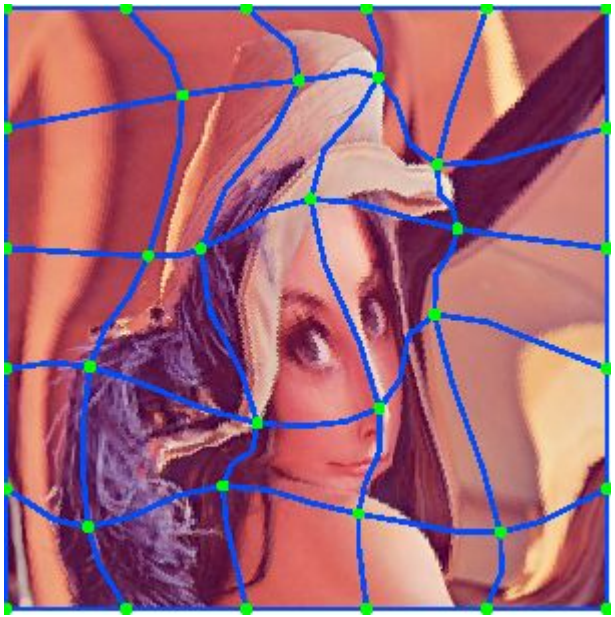
$$p_{i,j}(x) = \frac{(x - x_j)p_{i,j-1}(x) - (x - x_i)p_{i+1,j}(x)}{x_i - x_j}, \quad 0 \le i < j \le n.$$

- **Blending function:**

$$\mathbf{C}_i(\theta) = \cos^2\theta\, \mathbf{F}_i(\theta + \tfrac{\pi}{2}) + \sin^2\theta\, \mathbf{F}_{i+1}(\theta)\,,$$

Citation: *Cem, Yuksel. 2020. A class of C2 interpolating splies. ACM Transactions on Graphics, 39, 5, 2020*

Step 4 - texture mapping

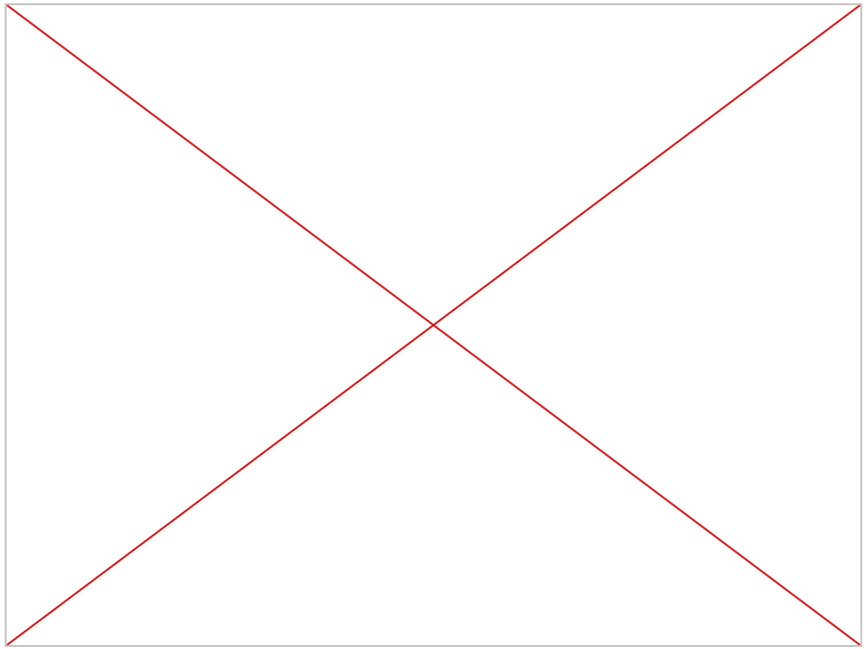




Additional functionalities

- Register key frames, and interpolate between frames to get customized animation.
- Customize anchor points.
- Directly select and drag control points.
- Precompute a trajectory for each control points, so they move automatically.

Precompute trajectory demo



3D Scene Flow Using Lightfield Images

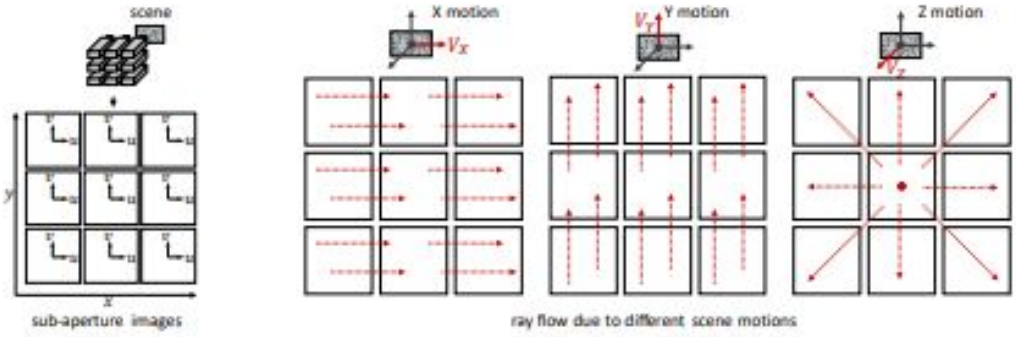
Kevin Wang

Lightfield Image Structure

- 4-dimensional L(x, y, u, v):
  - X and y are the camera coordinates
  - U and v are the pixel coordinates
- Given two lightfield images, we want to find 3D scene flow (Motion in X, Y, and Z)

Relating Ray Flow to Optical Flow

- Optical Flow Equation: $I_X V_X + I_Y V_Y + I_t = 0$
- Raw Flow Equation: $L_X V_X + L_Y V_Y + L_z V_z + L_t = 0$
  - L_z is found with relation to the pinhole camera position, partial derivatives of the X and Y positions and a fixed depth constant



Lucas-Kanade

- Because of the form of the ray flow equation, and it's similarity to the traditional optical flow equation, we can use existing methods to estimate depth motion.
- For Lucas-Kanade, finding local motion (assumes that pixels in the same area will have the same motion)

$$\mathbf{A} = \begin{bmatrix} L_X(\mathbf{x}_1) & L_Y(\mathbf{x}_1) & L_Z(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ L_X(\mathbf{x}_n) & L_Y(\mathbf{x}_n) & L_Z(\mathbf{x}_n) \end{bmatrix}, \mathbf{b} = \begin{bmatrix} -L_t(\mathbf{x}_1) \\ \vdots \\ -L_t(\mathbf{x}_n) \end{bmatrix}.$$

$$\mathbf{V} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}.$$

Horn-Schunck

- Assumes that the flow between pixels is “smooth” (global)

$$E(\mathbf{V}) = E_D(\mathbf{V}) + E_S(\mathbf{V}), \qquad \text{where} \tag{11}$$

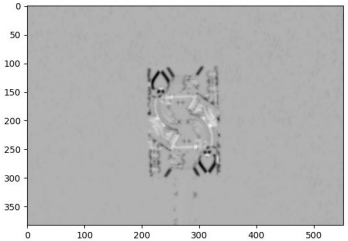
$$E_D(\mathbf{V}) = \int_\Omega (L_X V_X + L_Y V_Y + L_Z V_Z + L_t)^2 \, dx\,dy\,du\,dv\,,$$

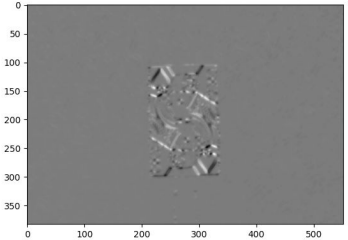
$$E_S(\mathbf{V}) = \int_\Omega \left(\lambda |\nabla V_X|^2 + \lambda |\nabla V_Y|^2 + \lambda_Z |\nabla V_Z|^2\right) dx\,dy\,du\,dv\,.$$

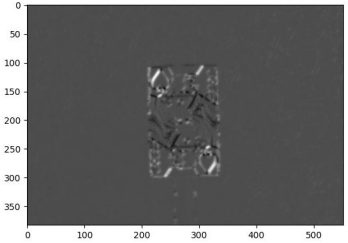
- Minimum can be found using the Euler-Lagrange Equations

Example Results

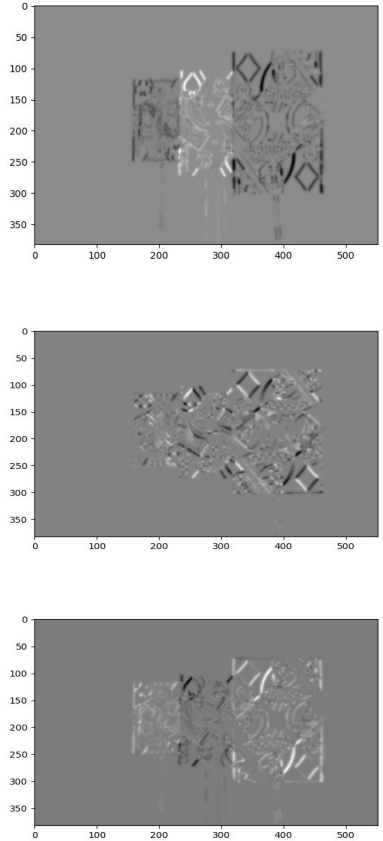











0
50
100
150
200
250
300
350
0
100
200
300
400
500

A Computational Approach for Obstruction-Free Photography

Wesley Wang
(wesleyw)

Overview:

- Take photos through occluding elements like windows and fences
- Remove those occlusive elements from the photos
- Capture video (or image sequence) while moving the camera
- Background scene must be far enough that it remains static

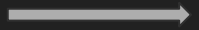


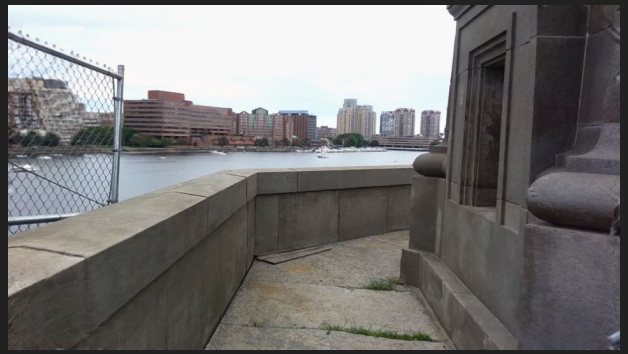


Overview:

- Occlusive Foreground Objects

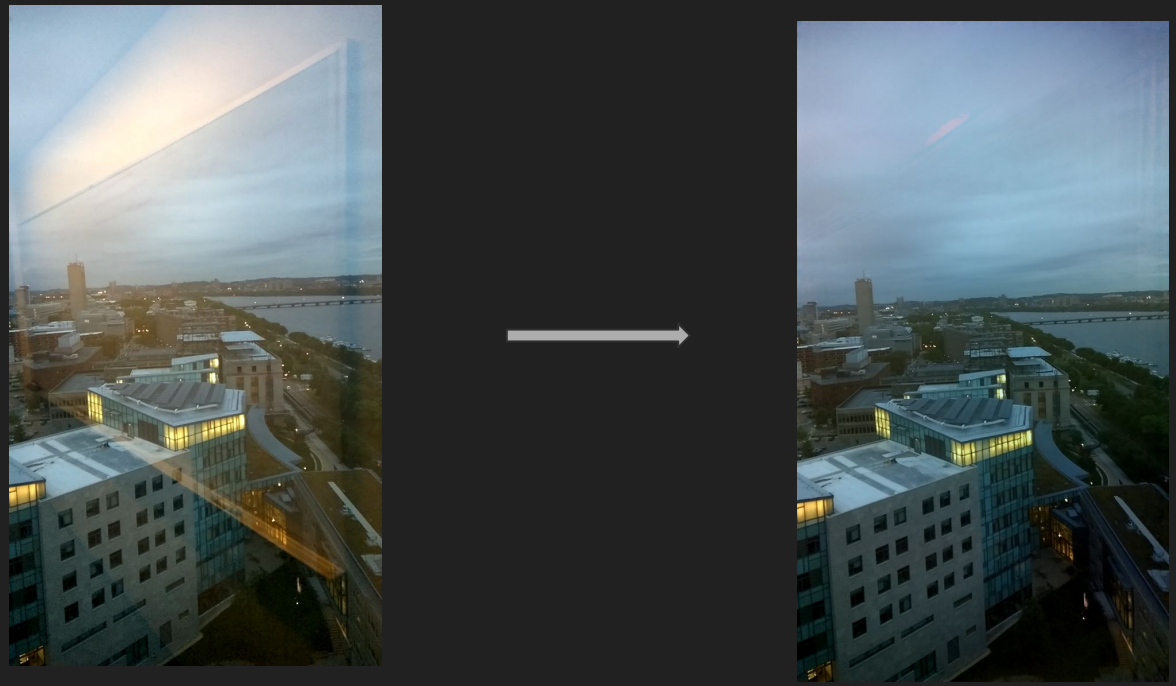






Overview:

- Window Reflections



Implementation:

- Treat image as composition of a background layer and obstruction layer

$$I = (1 - A) \circ I_O + A \circ I_B$$

- I_O is occlusion (foreground) layer
- I_B is background layer
- A is alpha mask where occlusion exists in image

Implementation:

- Solve for those image components using info from other frames
- Also solve for the warping W(V) of layers between frames

$$I^t = \mathbf{W}(V_O^t) I_O + \mathbf{W}(V_O^t) A \circ \mathbf{W}(V_B^t) I_B$$

- Solve optimization problem

$$\min_{I_O, I_B, A, \{V_O^t\}, \{V_B^t\}} \sum_t \| I^t - \mathbf{W}(V_O^t) I_O - \mathbf{W}(V_O^t) A \circ \mathbf{W}(V_B^t) I_B \|_1$$
$$+ \lambda_1 \|\nabla A\|_2^2 + \lambda_2 (\|\nabla I_O\|_1 + \|\nabla I_B\|_1) + \lambda_3 L(I_O, I_B) + \lambda_4 \sum_t \|\nabla V_O^t\|_1 + \|\nabla V_B^t\|_1$$

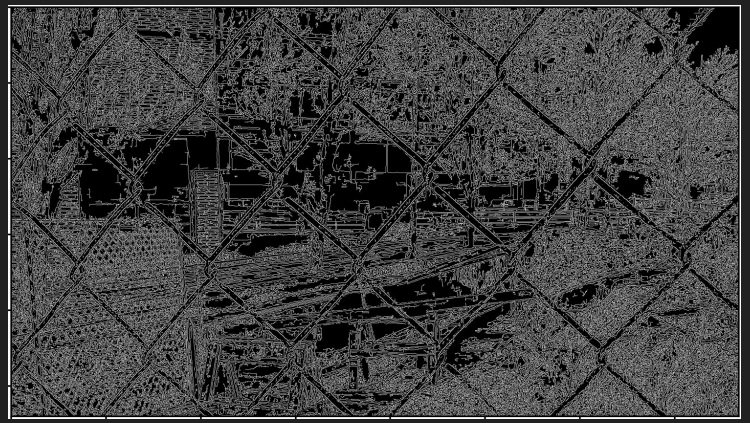
Subject to:

$$0 \leqslant I_O, I_B, A \leqslant 1,$$

Results:

- Motion estimation between frames after processing + edge detection





Questions?

Confocal Stereo

Hiroshi Wu

Confocal Stereo

- A method to extract depth information using multiple images of the same scene with varying focus and aperture
- https://people.csail.mit.edu/hasinoff/pubs/hasinoff-confocal-2006.pdf





In assignment 4...

1. Simulated a focal-aperture stack based on image from plenoptic camera
2. Constructed AFI (aperture-focus-image) for each pixel
3. Used direct variance evaluation to determine depth d at each pixel

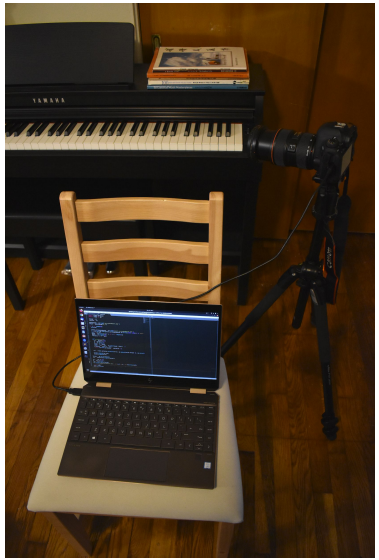
The whole process (as detailed in paper)

1. Aperture-focal stack image acquisition using a DSLR
2. Relative exitance estimation (radiometric calibration)
3. Image alignment (lens warp calibration)
4. AFI construction
5. Depth estimate by AFI model fitting

Although the core idea is the same, the whole process is more involved

1. Aperture-focal stack acquisition

- Depth from blur: bigger bokeh = better resolution
- To maximize bokeh:
  - Used a lens with f/2.8 aperture
  - Zoomed to the maximum focal length on that lens: 70mm
- Took photos of the same scene for 80 focus settings (0.38m ~ 3.4m) and 16 apertures (2.8 ~ 16)
- Used gphoto2 to automate the process; one set of data took 1 hour to capture





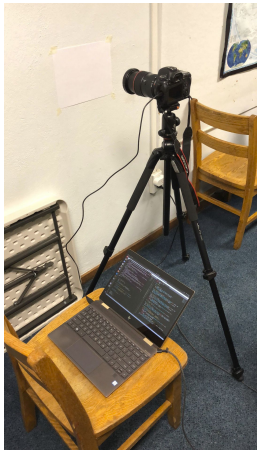
Example: f=50, a=2.8

2. Relative exitance estimation

- Since we compare pixel values across the AF stack, we need to ensure that same values mean the same thing
- E.g. vignetting
- We take photos of a uniform white paper using similar focus settings and aperture values and check the exitance at each pixel and each setting so we can adjust for brightness differences

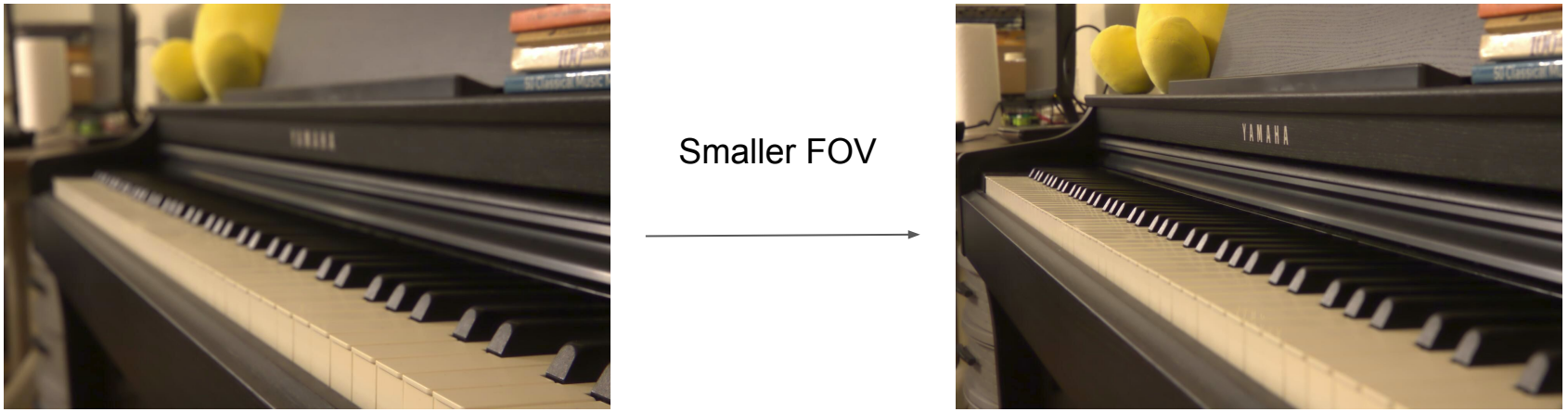
$$E_{xy}(\alpha, f) = \frac{I_{\alpha f}(x, y)}{I_{\alpha 1 f}(x, y)}$$





3. Image alignment

- Changing focus distance actually changes the scene due to lens mechanics:

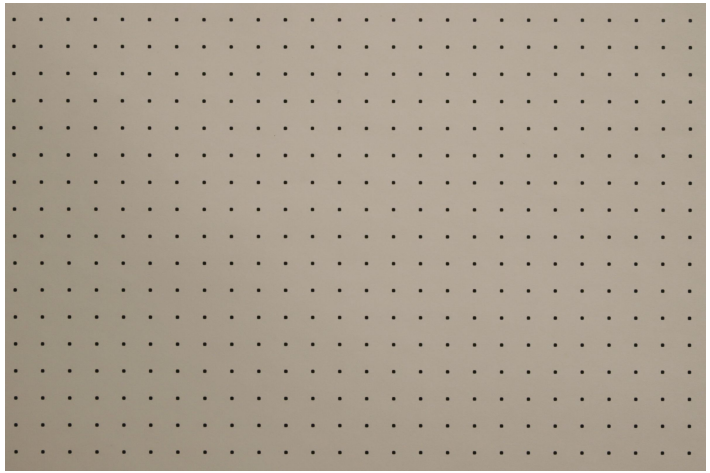


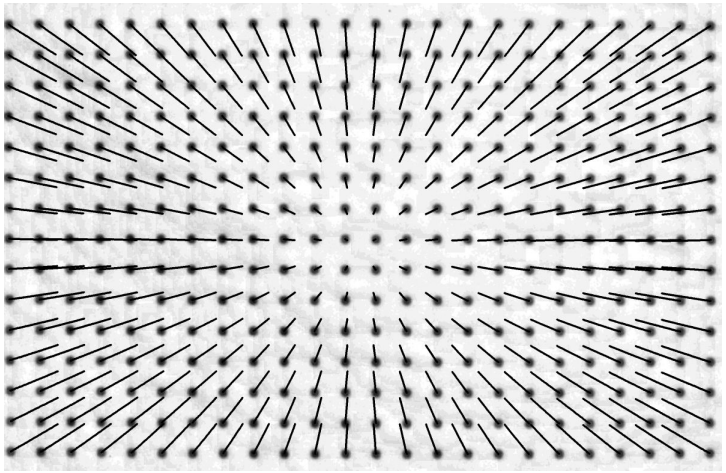
- Modelled as a magnification + quadratic distortion

$$\mathbf{w}_f^{\mathrm{D}}(x,y) = \left[m_f + m_f(f-f_1)(k_0 + k_1 r + k_2 r^2) - 1\right] \cdot \left[(x,y) - (x_c, y_c)\right]$$

3. Image alignment

- Use focal stack of a scene with distinct features to obtain the warp relationships between different focus, and use nonlinear optimization to solve for the parameters



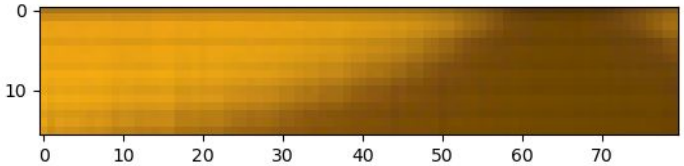


4. AFI

- Example from paper:

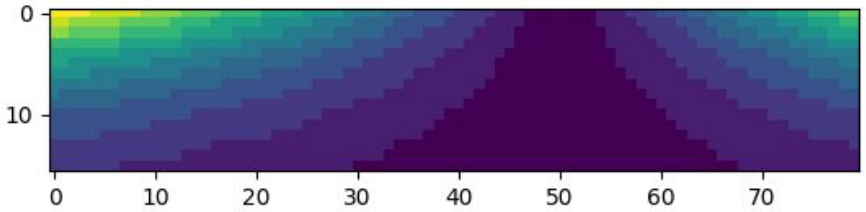


- Example from my data:



5. AFI model fitting

- There is a distinct pattern to AFIs
- Due to different amounts of blurring: similar blurs should have similar colors
- We derive blur = $\frac{\lambda}{2\alpha}\frac{k(f'-f)}{(f-x_0)(k-y_0f'-y_0x_0)}$ where the real-life distance of configuration x is $dist(x) = \frac{k}{x-x_0} + y_0$ due to the thin lens model.
- Quantizing theoretical blur gives us these bands; we seek to minimize the variance within the bands



6. Result from piano dataset

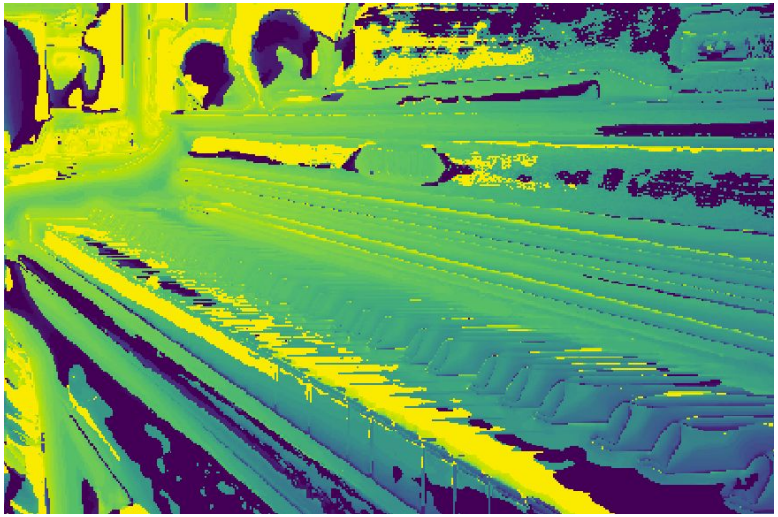


Image smoothing via L0 Gradient Minimization

Ye Wu

Objective

- Globally maintain and possibly enhance the most prominent set of edges by increasing steepness of transition while not affecting the overall acutance.
- A strategy to confine the discrete number of intensity changes among neighboring pixels, which links mathematically to the L0 norm for information sparsity pursuit.

Algorithm

Algorithm 1 L_0 Gradient Minimization

Input: image I, smoothing weight λ, parameters β_0, $\beta_{\max}$, and rate κ

Initialization: $S \leftarrow I$, $\beta \leftarrow \beta_0$, $i \leftarrow 0$

repeat

With $S^{(i)}$, solve for $h_p^{(i)}$ and $v_p^{(i)}$ in Eq. (12).

With $h^{(i)}$ and $v^{(i)}$, solver for $S^{(i+1)}$ with Eq. (8).

$\beta \leftarrow \kappa\beta$, $i++$.

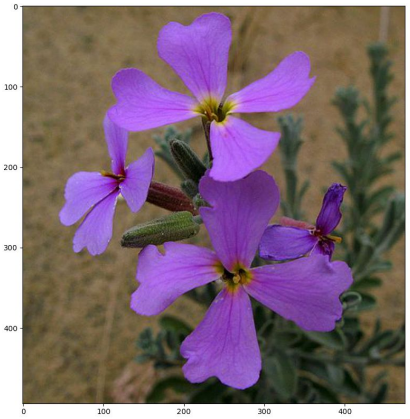
until $\beta \geq \beta_{\max}$

Output: result image S

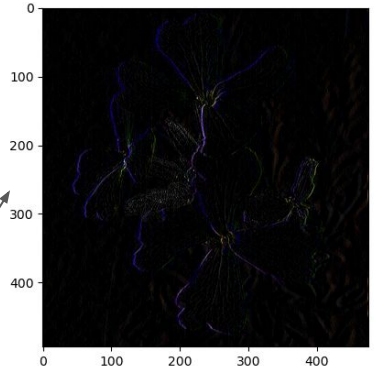
$$S = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(I)+\beta(\mathcal{F}(\partial_x)^*\mathcal{F}(h)+\mathcal{F}(\partial_y)^*\mathcal{F}(v))}{\mathcal{F}(1)+\beta(\mathcal{F}(\partial_x)^*\mathcal{F}(\partial_x)+\mathcal{F}(\partial_y)^*\mathcal{F}(\partial_y))}\right), \quad (8)$$

$$(h_p, v_p) = \begin{cases} (0,0) & (\partial_x S_p)^2 + (\partial_y S_p)^2 \leq \lambda/\beta \\ (\partial_x S_p, \partial_y S_p) & \text{otherwise} \end{cases} \quad (12)$$

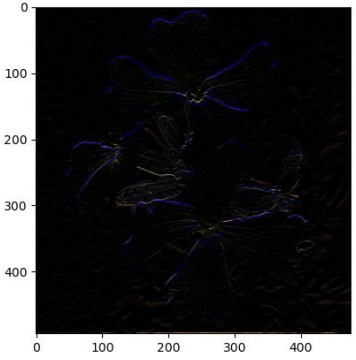
Pipeline



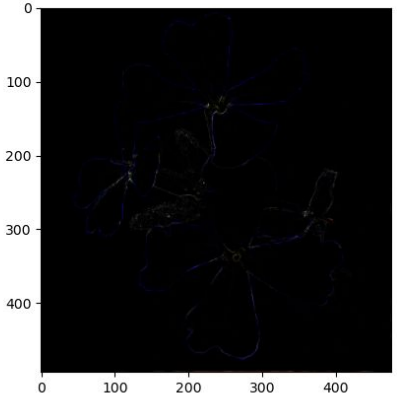
Start



Gradient x

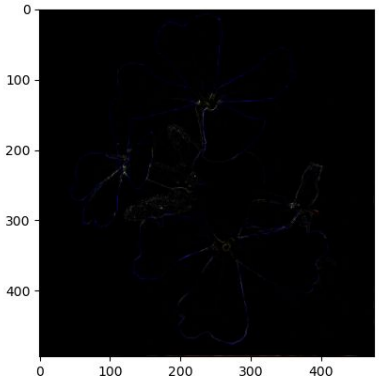


Gradient y

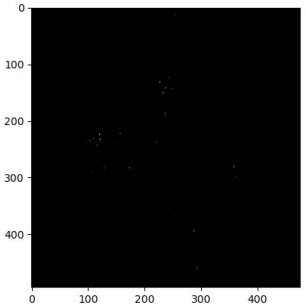


Mix 1

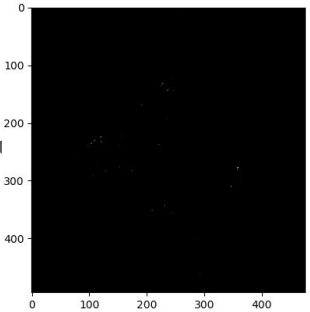
Pipeline



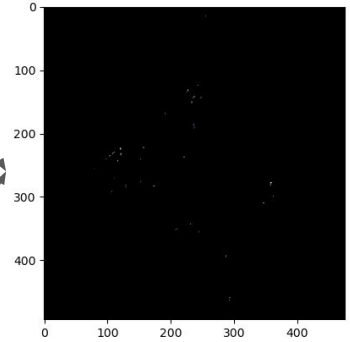
Mix 1 with threshold



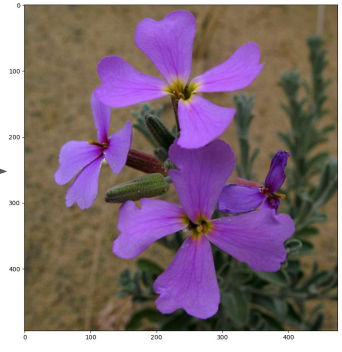
Gradient x new



Gradient y new



Mix 2

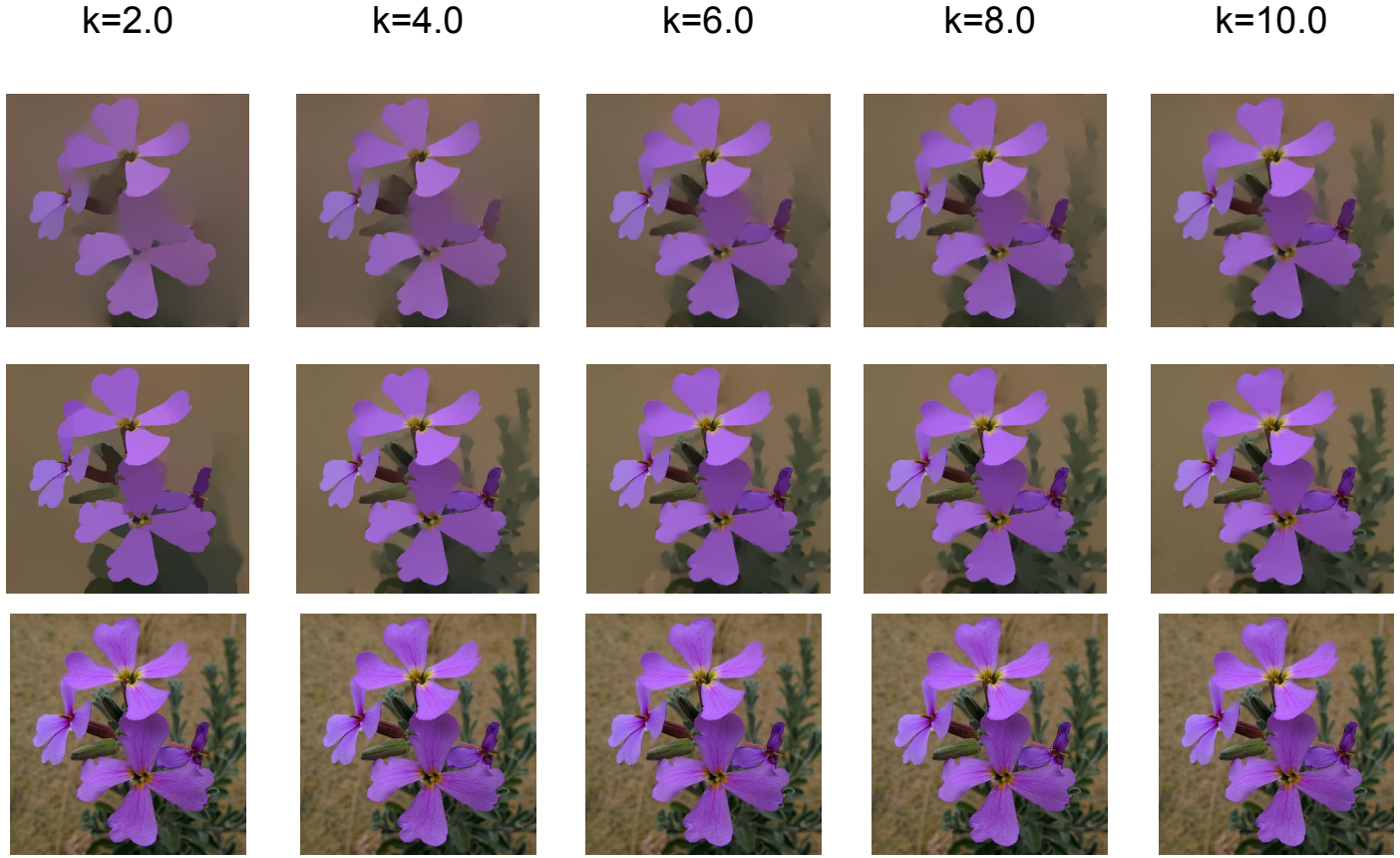


end

Results

	l=0.1	l=0.01	l=0.001	l=0.0001	l=0.00001
k=2.0					
k=3.0					

Results



Comparison

L0 minimization

3.05s





Bilateral filtering

64.13s





Application

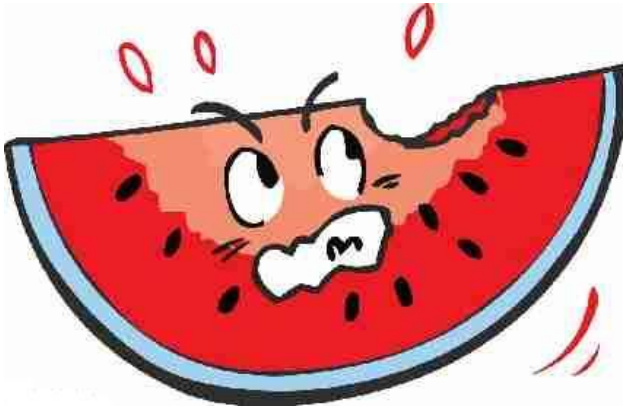
Image abstraction





Application

Clip-art compression artifact removal





Application

Combination of BLF and L0 method

Origin





BLF

L0





BLF+L0

Main References

http://www.cse.cuhk.edu.hk/~leojia/projects/L0smoothing/

https://www.caam.rice.edu/~yzhang/reports/tr0710_rev.pdf

Thank you!
Q&A

Fast Reflection Removal using Hierarchical Bilateral Grids

Zhichao Yin
(zhichaoy@)

Reflection removal



Photo taken through glass



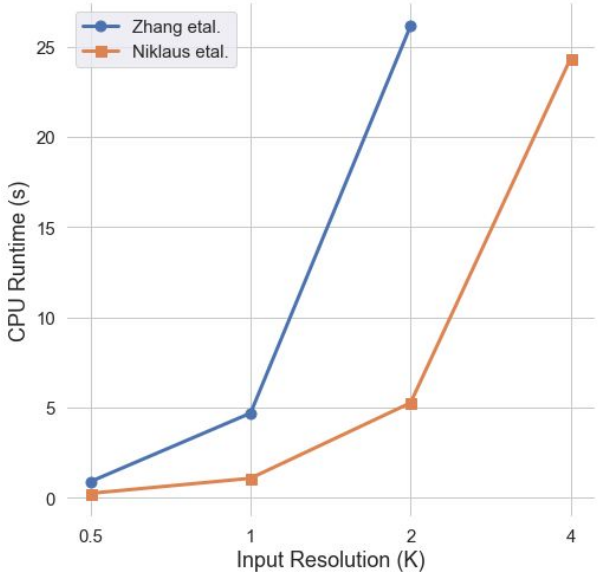
Ground-truth background/transmissive layer

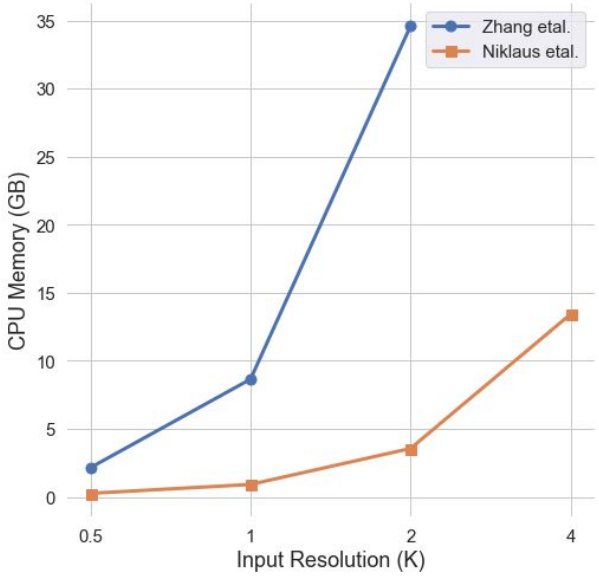
High-res reflection removal

- Q: can existing approaches work well for high-res input images?

- Challenges:
  - ❑ slow runtime
  - ❑ memory consumption
  - ❑ degraded quality (CNNs are not scale equivariant)

High-res challenges – runtime & memory

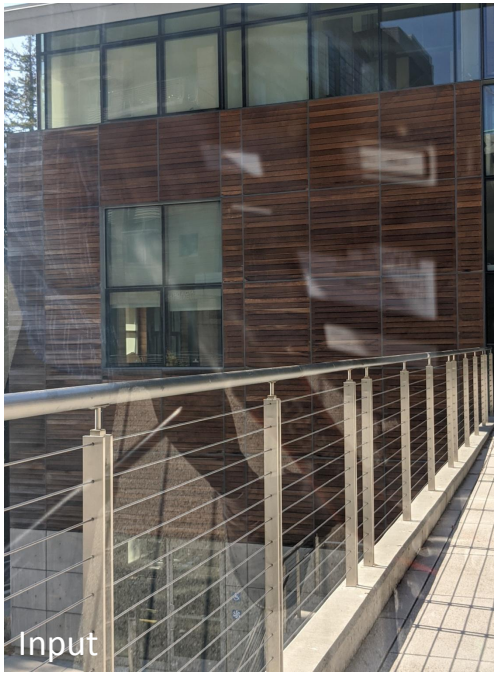


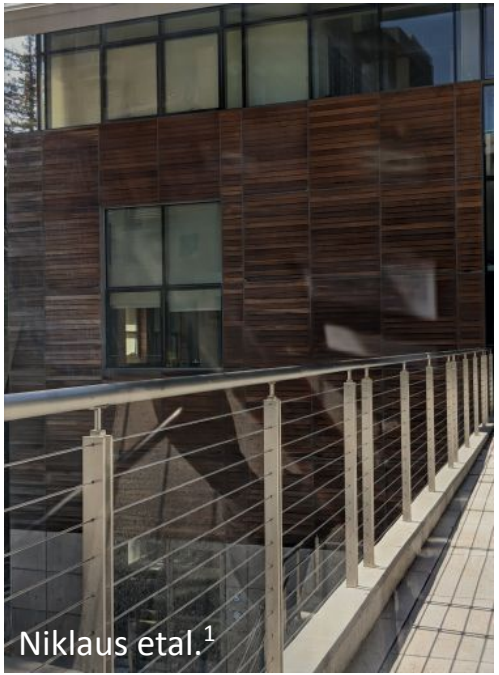


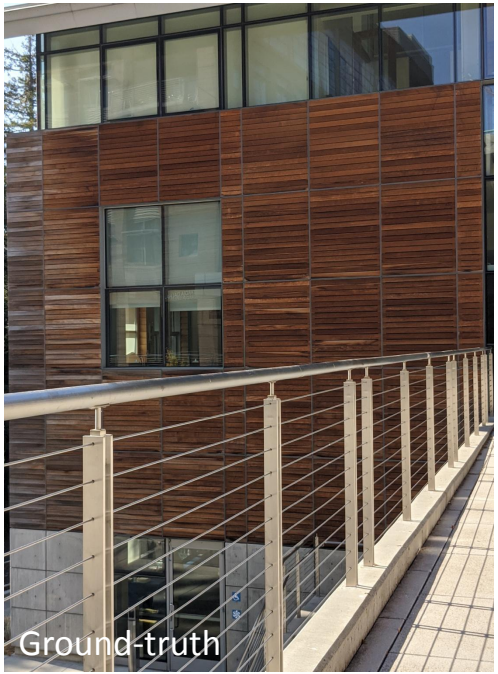
CPU runtime & memory vs. input resolution on real-world test set (208 images)

1. Zhang et al. "Single Image Reflection Separation with Perceptual Losses." CVPR, 2018
2. Niklaus et al. "Learned Dual-view Reflection Removal." In Submission, 2020

High-res challenges – degraded quality



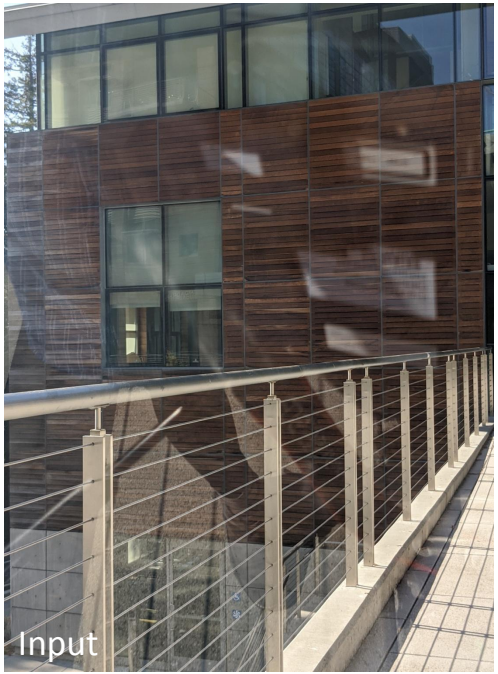


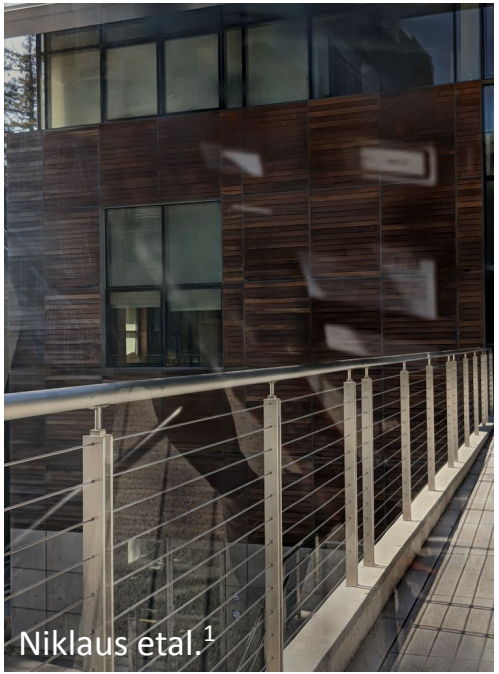


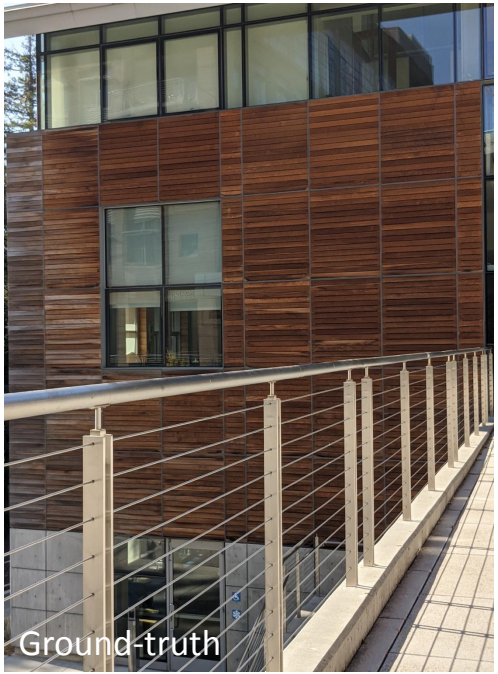
Result upsampled from prediction with 512-res input

1. Niklaus et al. ”Learned Dual-view Reflection Removal.” In Submission, 2020

High-res challenges – degraded quality



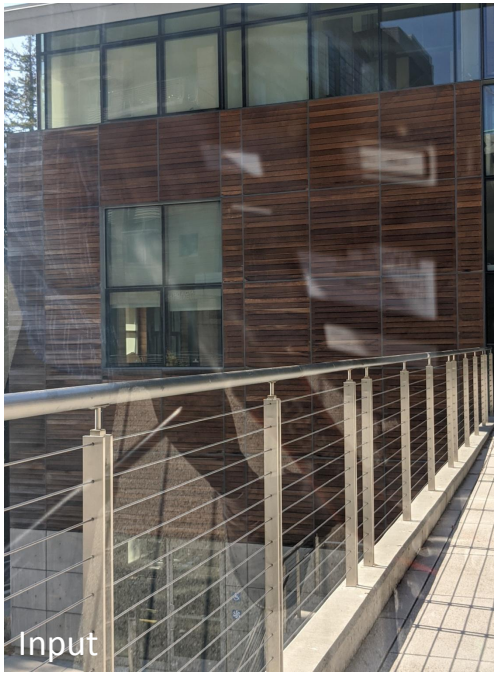


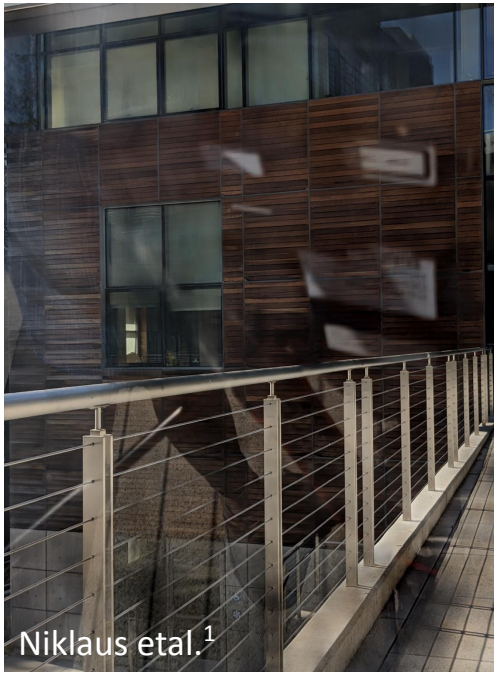


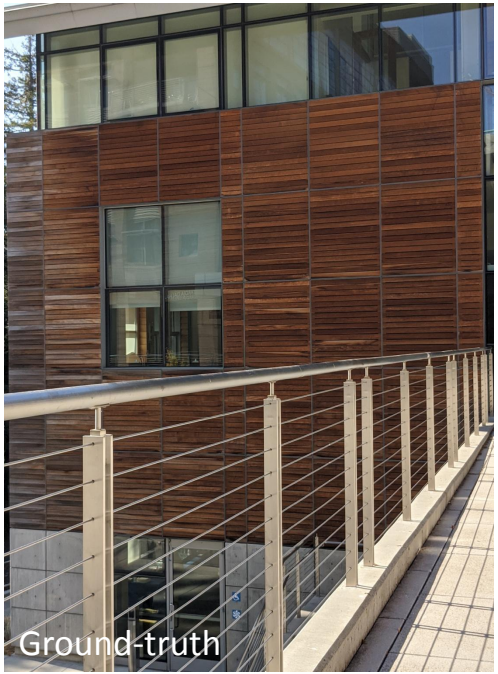
Result upsampled from prediction with 1024-res input

1. Niklaus et al. ”Learned Dual-view Reflection Removal.” In Submission, 2020

High-res challenges – degraded quality







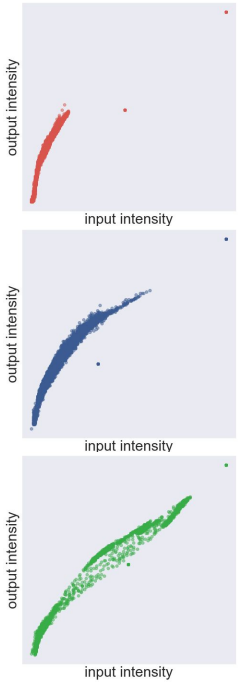
Result upsampled from prediction with 2048-res input

1. Niklaus et al. ”Learned Dual-view Reflection Removal.” In Submission, 2020

HDRNet[1] works well for tone mapping





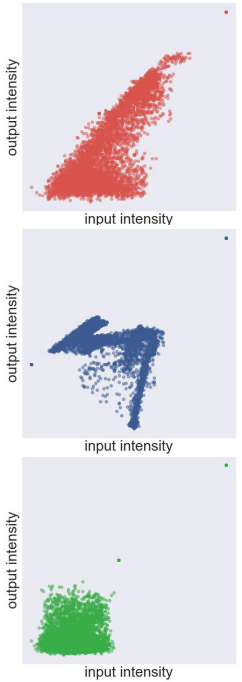


1. Gharbi et al. "Deep Bilateral Learning for Real-Time Image Enhancement." SIGGRAPH, 2017

But HDRNet[1] cannot work for dereflection



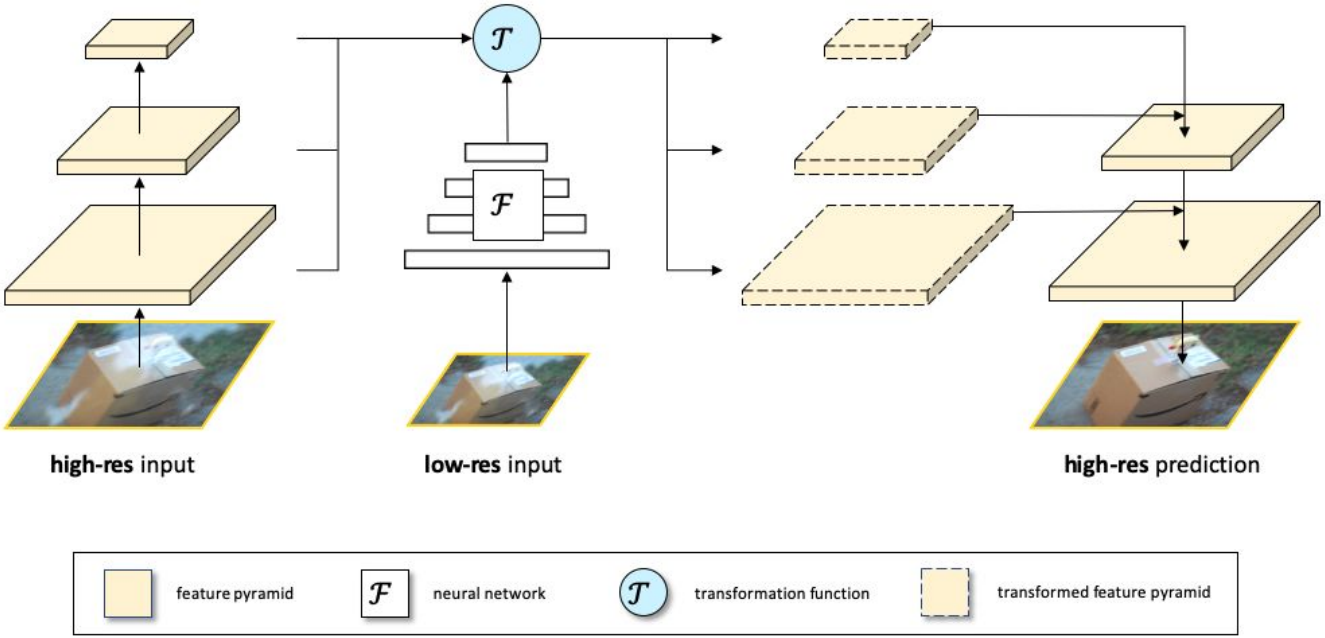




Addition of reflection parts makes the transformation highly non-linear in **color space**.

1. Gharbi et al. "Deep Bilateral Learning for Real-Time Image Enhancement." SIGGRAPH, 2017

Our method – overview



High-res Comparison



Input, Resolution: 2048 x 1536



GT, Resolution: 2048 x 1536

High-res Comparison



Input, Resolution: 2048 x 1536



Niklaus etal.

1. Niklaus et al. "Learned Dual-view Reflection Removal." In Submission, 2020

High-res Comparison



Input, Resolution: 2048 x 1536



Ours

High-res Comparison



Input, Resolution: 2048 x 1536



Zhang etal.

1. Zhang et al. "Single Image Reflection Separation with Perceptual Losses." CVPR, 2018

High-res Comparison



Input, Resolution: 2048 x 1536



Ours

High-res
Comparison



High-res
Comparison



High-res Comparison



1. Niklaus et al. ”Learned Dual-view Reflection Removal.” In Submission, 2020

High-res
Comparison



High-res Comparison



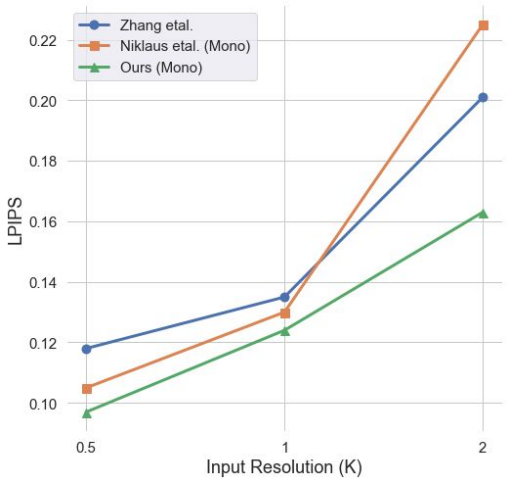
1. Zhang et al. "Single Image Reflection Separation with Perceptual Losses." CVPR, 2018

High-res
Comparison

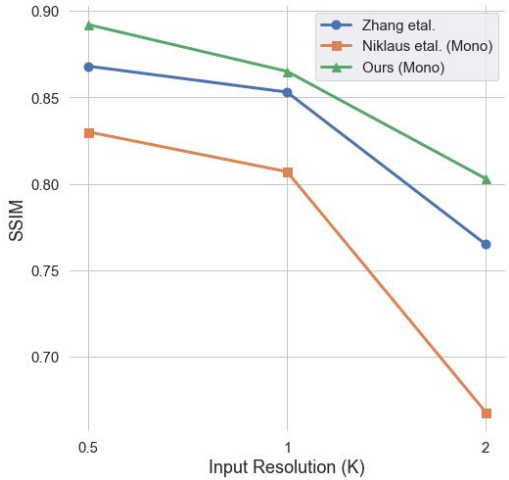


Quantitative Evaluation

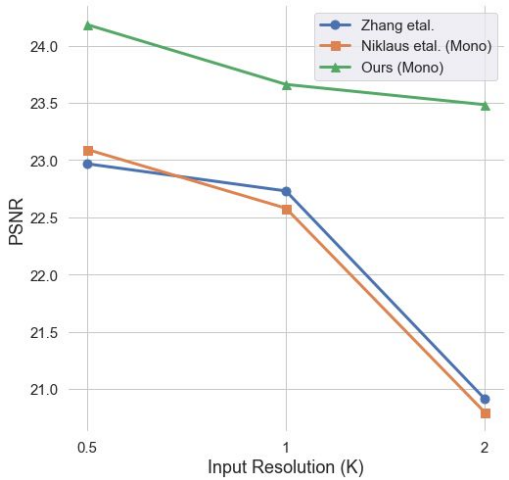
LPIPS ↓



SSIM ↑



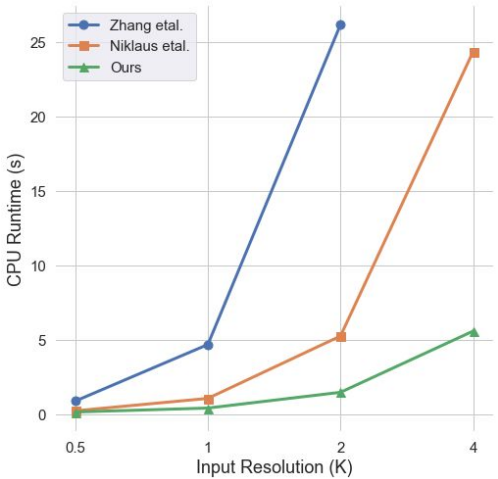
PSNR ↑



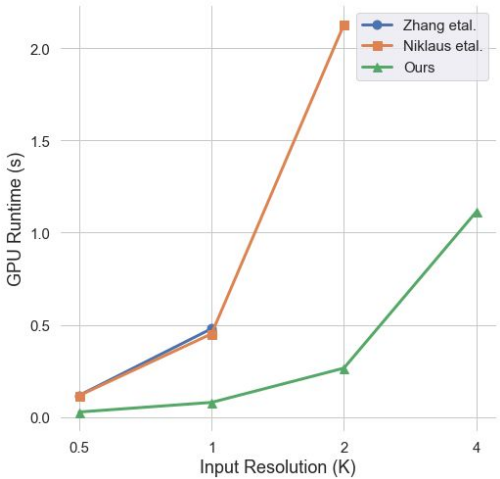
1. Zhang et al. "Single Image Reflection Separation with Perceptual Losses." CVPR, 2018
2. Niklaus et al. "Learned Dual-view Reflection Removal." In Submission, 2020

Runtime Comparison

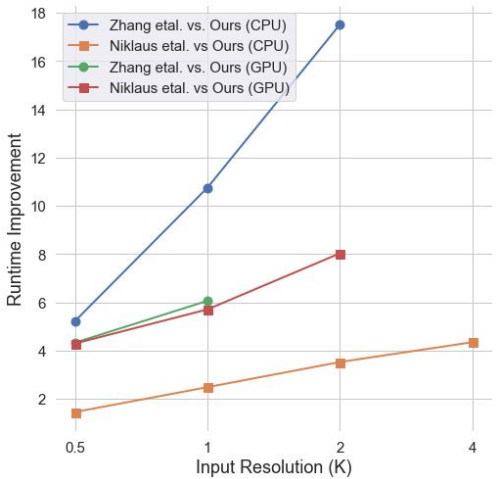
CPU



GPU



Relative Improvement



1. Zhang et al. "Single Image Reflection Separation with Perceptual Losses." CVPR, 2018
2. Niklaus et al. "Learned Dual-view Reflection Removal." In Submission, 2020

Lensless Imaging with DiffuserCam

Emily Zheng

Exploring Lensless Imaging

- Cameras with lenses are most commonly used to capture images
- Lensless imaging systems
  - Can be more lightweight
  - Can be constructed easily without the precision of a normal lens using everyday materials
- **DiffuserCam:** replace lens with a diffuser (Scotch tape)
  - From DiffuserCam Paper

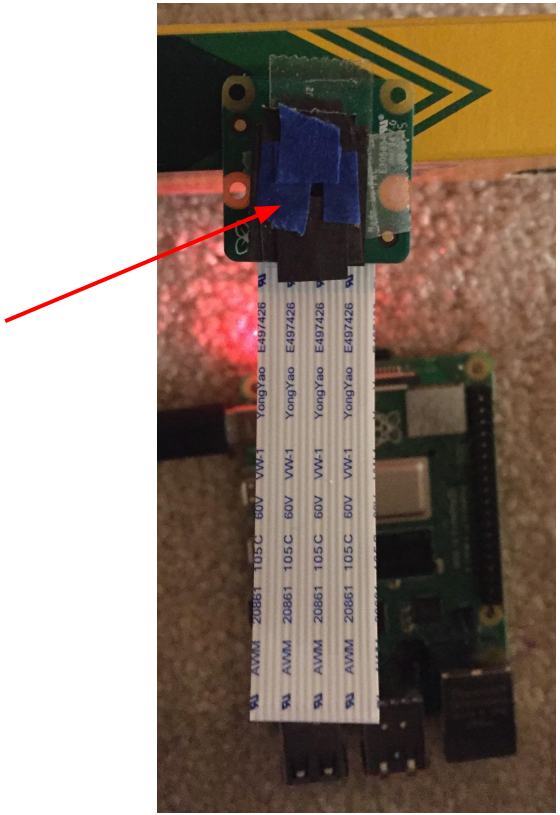
DiffuserCam Setup + Calibration

Setup

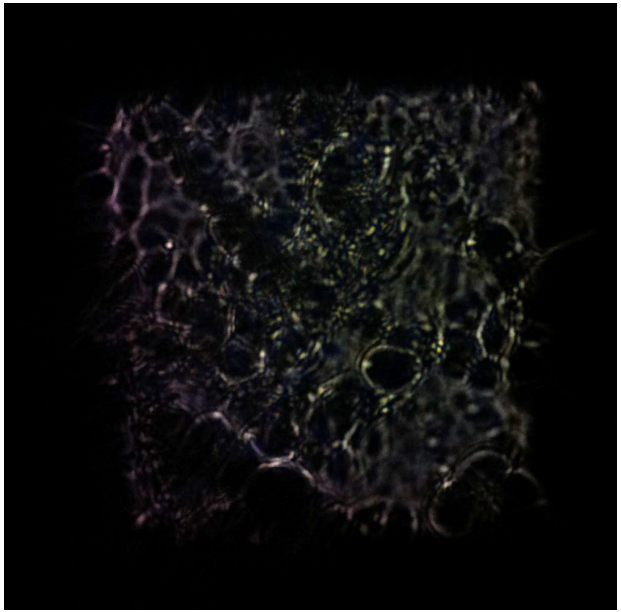
- RaspberryPi PiCamera with lens removed (only sensor left)
- Place a diffuser at focal length of PiCamera (double sided Scotch tape)
- Create aperture using black paper / tape

Calibration

- Take a picture of a point source through pinhole to get psf of the diffuser



Example PSF and Image Capture

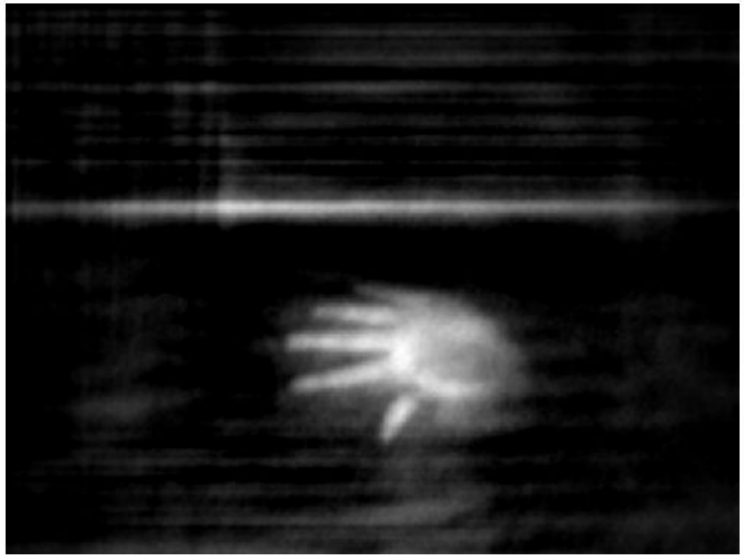


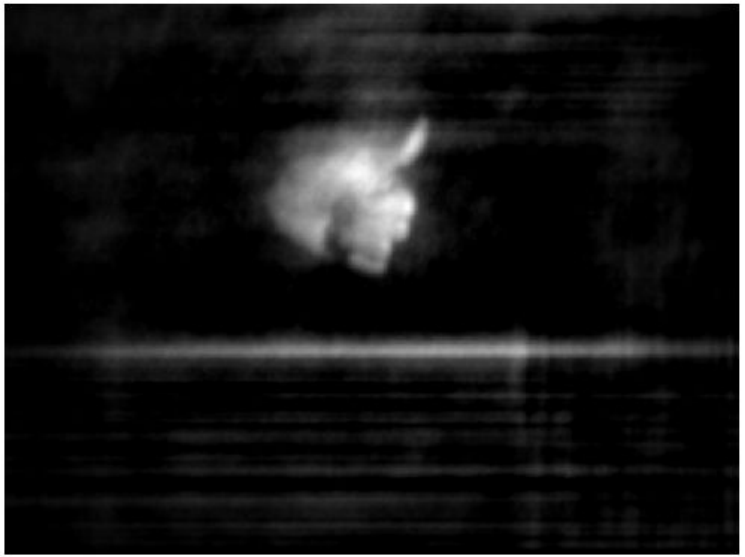


Reconstruction with Deconvolution

- Alternating direction method of multipliers (ADMM)
- Optimization problem
  - Assume image of scene captured is result of some convolution of original scene with blur of diffuser
  - Blur from diffuser is known from the psf
  - Solve for original scene image using ADMM
- Can reconstruct original scene with a single diffused image

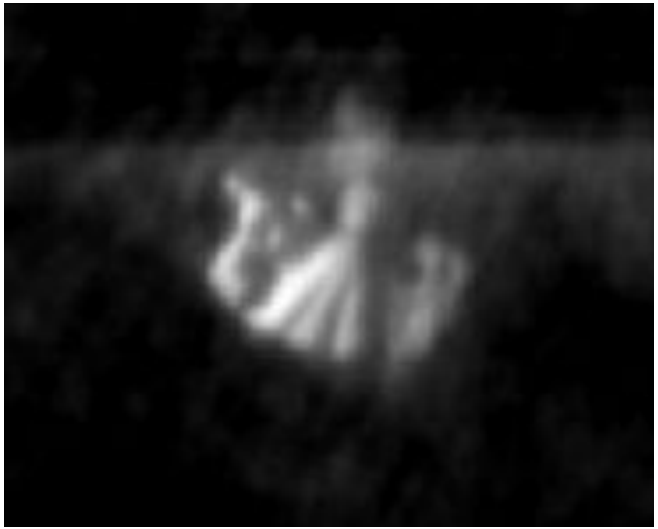
Results





Results







Remaining course logistics

- Optional homework assignment 7 is due on 12/18 at 23:59 ET.
  - Gradescope submission site available.

- Final project reports are due on Sunday 12/20 at 23:59 ET.
  - Gradescope submission site available.

- One upcoming computational photography talk:
  - Tuesday 12/22, 1 – 2 pm, Grace Kuo (lensless cameras, diffuserCam, AR/VR displays).

- Returning borrowed equipment:
https://docs.google.com/spreadsheets/d/1CVg7nUbI701pvZFPX3BR0uzKB76Y6PEl3tXmq1UF4AU/edit#gid=0

Class evaluation*s* – please take them!

- CMU's Faculty Course Evaluations (FCE): https://cmu.smartevals.com/
- TA evaluation: https://www.ugrad.cs.cmu.edu/ta/F20/feedback/
- 15-463/663/862 end-of-semester survey: https://docs.google.com/forms/d/e/1FAIpQLSew6XHzagr0KqWPFCmdoVmnHXQR5vfi4A6nEG0jMg6O1lBixA/viewform
- Please take all three of them, super helpful for developing future offerings of the class.
- Thanks in advance!