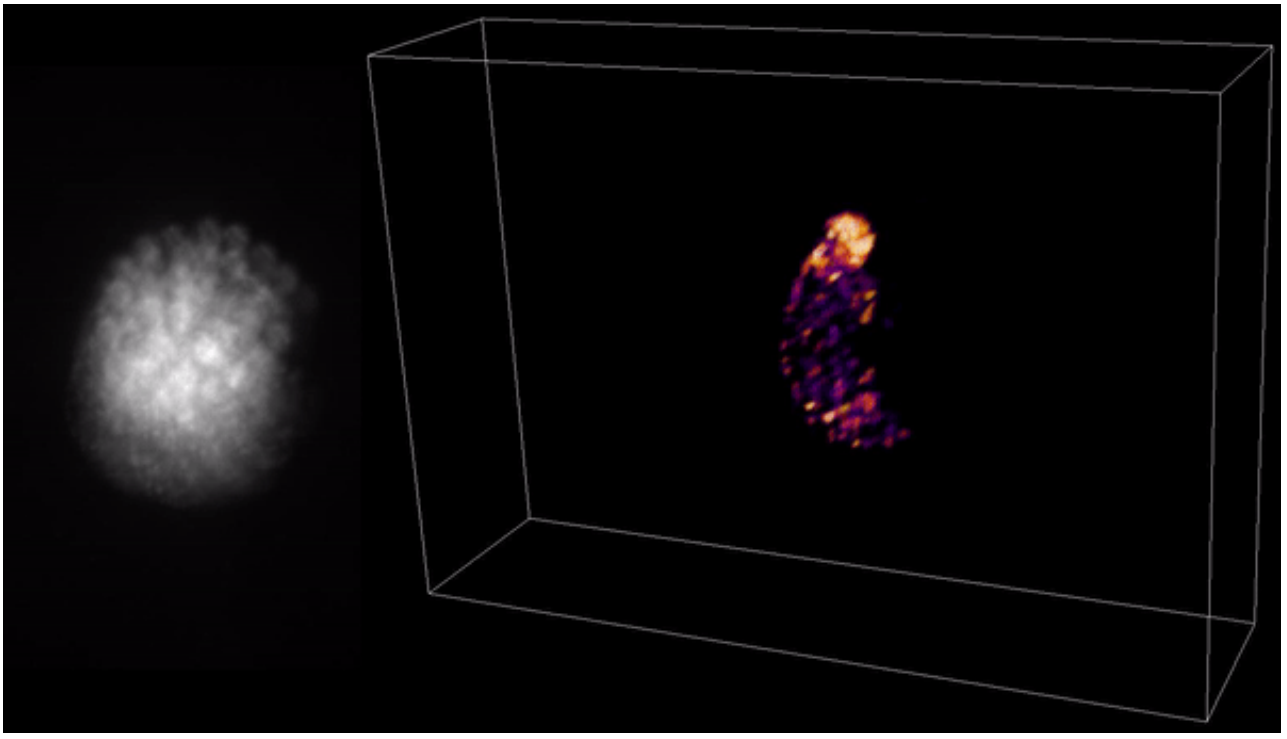
EE 046746 - Technion - Computer Vision

Elias Nehme

Tutorial 14 - Deep Computational Imaging



- Image source

Agenda

- What is Computational Imaging?
- Compressive Imaging
  - Depth Encoding PSF
- Deep "Optics"
  - Computer Vision Pipelines
  - Differentiable Optics
- Applications
- Recommended Videos
- Credits

What is Computational Imaging?

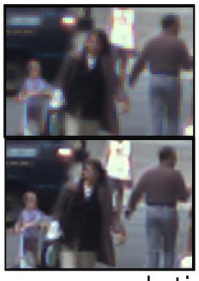
Co-design of optics, sensing, and algorithms!

Computational Imaging

What is Computational Imaging?

HDR Imaging

[Mann,Devebec,Nayar,…]



Super-resolution

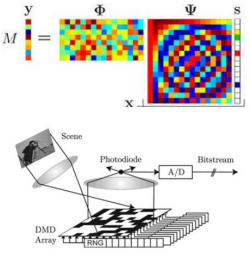
[Baker,Ben-Ezra,…]

EDOF

[Dowski,Nayar,…]

Light Fields

[Levoy,…]



Compressive Imaging

[Baraniuk,…]

Computational Imaging

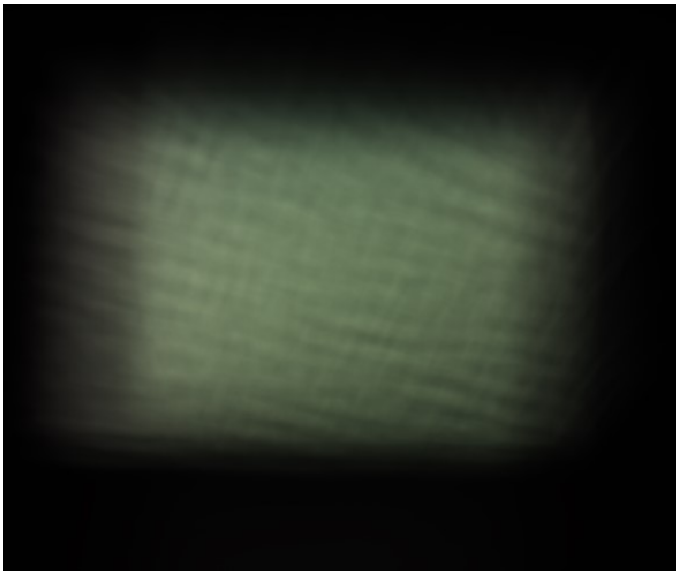
optics
sensing
computation

Compressive Imaging

- Depth encoding PSF

Depth Encoding PSF

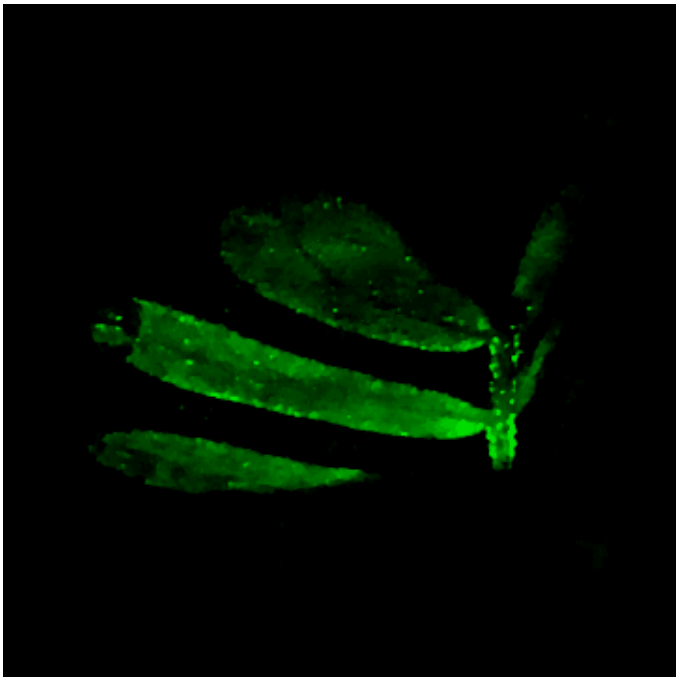
- Measurement is a 2D image:



- Image source - Optica 2018

Depth Encoding PSF

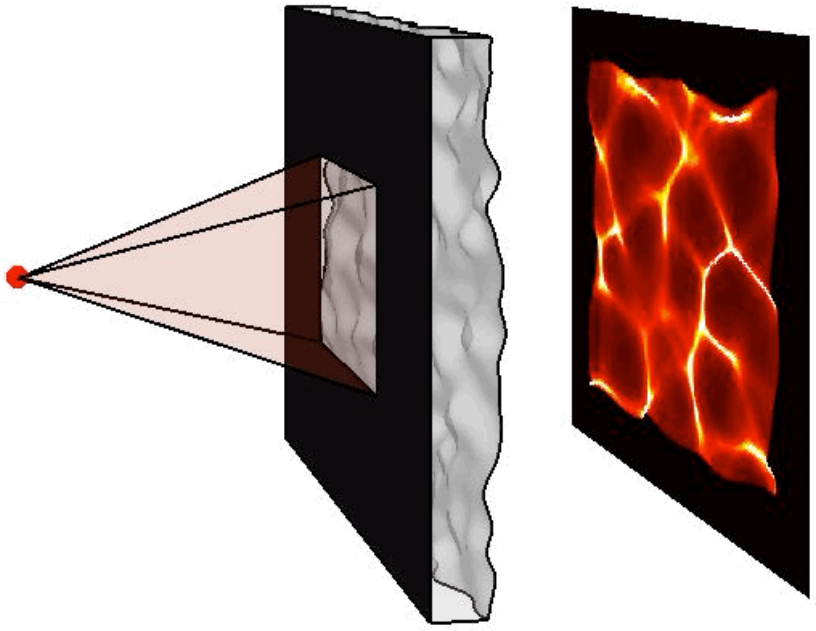
- Recovery is a 3D volume:



- What?? How?!
- Image source - Optica 2018

Depth Encoding PSF

- Depth Encoding Impulse Response/Point Spread Function (PSF)
  - Main idea is to encode depth in the shape generated on the 2D sensor
  - This way, you can recover the depth from looking at the created shape



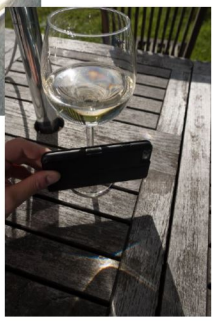
- Image source - Optica 2018

Depth Encoding PSF

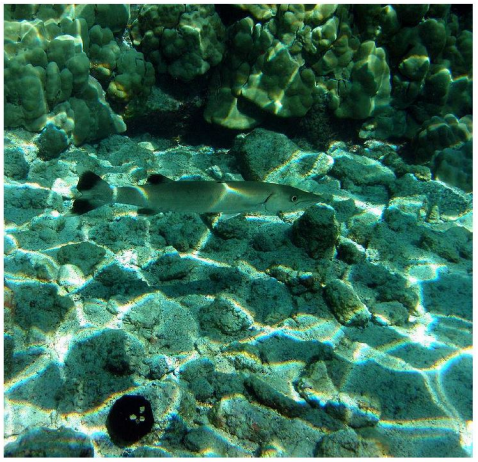
- Where is this inspired from?

Diffuser-induced caustics PSF

Glass of water



Glass of wine



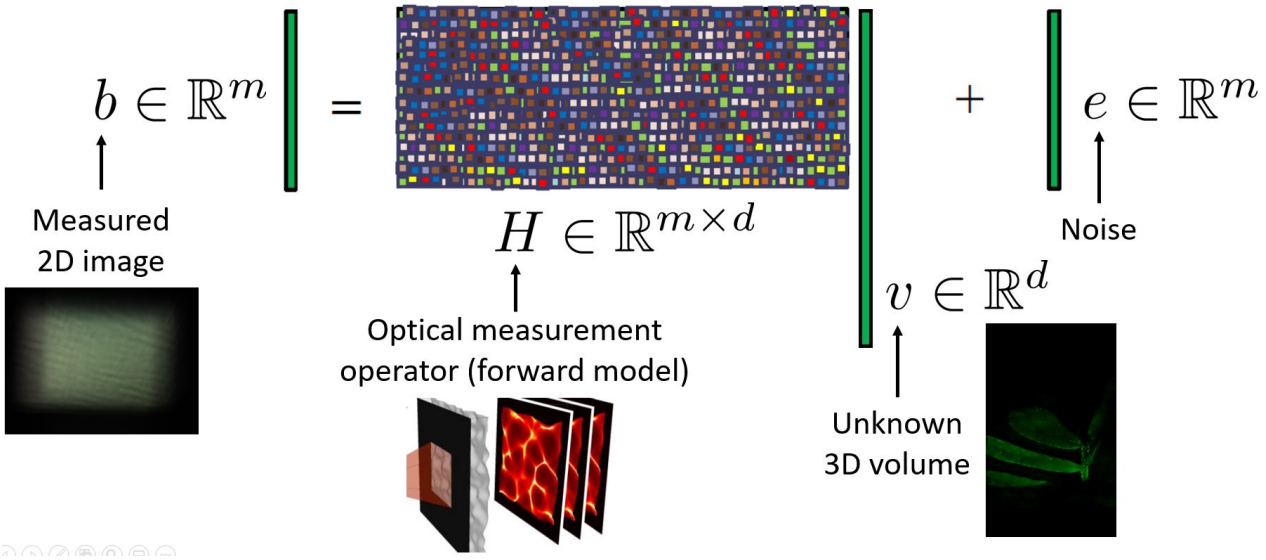
Caustics produced by the surface of water

https://en.wikipedia.org/wiki/Caustic_(optics)

- At sea, the light coming from above the water is going through different amount of water, since the surface of the water isn't still and smooth.
- This means that each point of light that reached the bottom has gained a different amount of phase.
  - It's a random amount phase!

Compressed Sensing

- How to get a 3D reconstruction from a 2d image? formulate an optimization problem!
- Writing down the problem in matrix formulation:

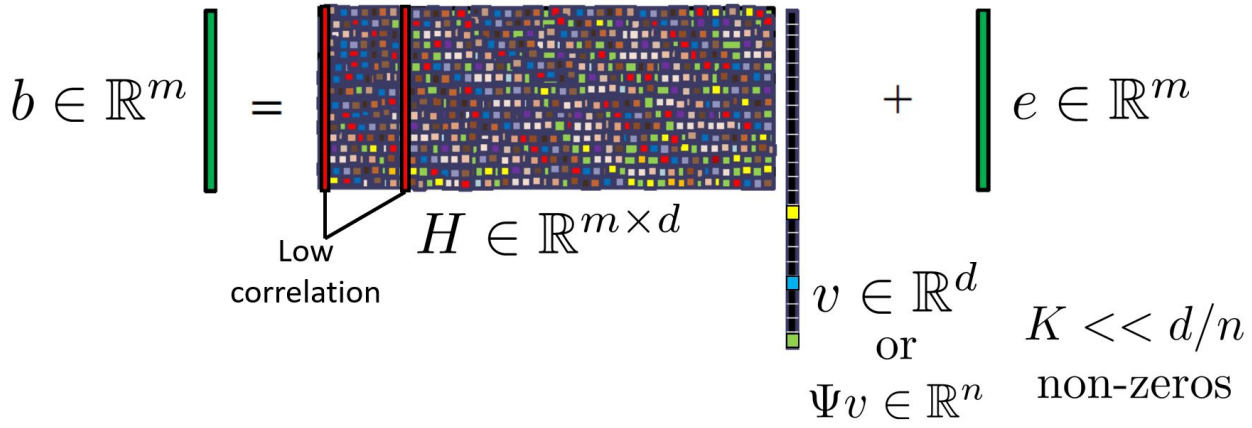


Compressed Sensing

- b - the column stack of the 2D measured image
- H - the convolution system that creates a 2D projection of a given 3D volume.
  - This is built from the PSF
- v - the column stack of all the voxels in the 3D volume.
  - This is the unknown we want to solve for
- e - noise added to each pixel

Compressed Sensing

- Compressive Imaging gives the solution by a "MAP" estimator under certain **conditions**



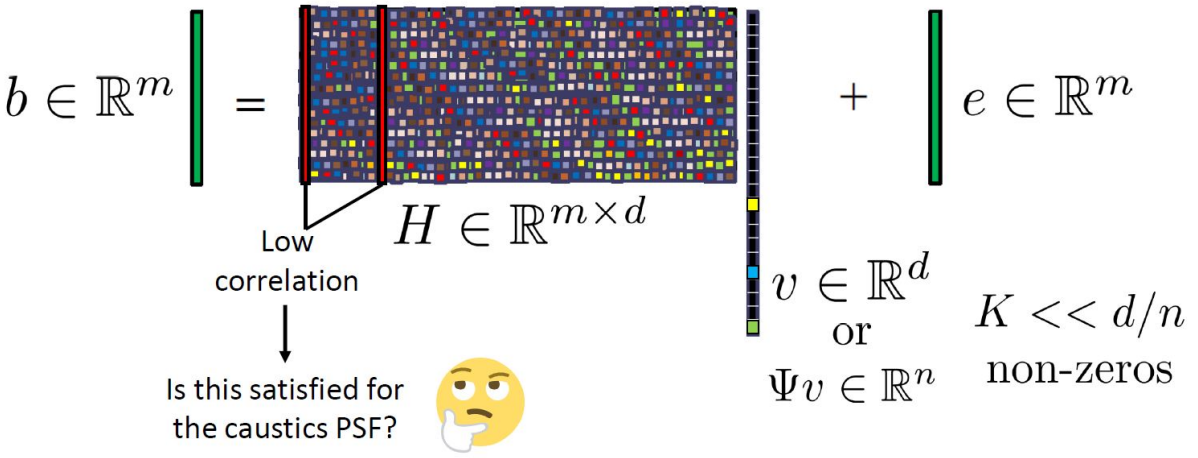
$$\hat{v}_{TV} = argmin_{v \geq 0} \left\{ \frac{1}{2} \|b - Hv\|^2 + \lambda \|\Psi v\|_1 \right\}$$

Compressed Sensing

- Conditions:
  1. The columns of H have low correlation or no correlation at all
  2. **Reularization condition**: v or a transformation of v, Ψv is a sparse signal (A small amount of coefficients K are different from zero)
- Under (1) + (2) the optimization probelm for $\hat{v}_{TV}$ can be solved
  - i.e. reconstructing the 3d volume (v) from a 2d image (b)

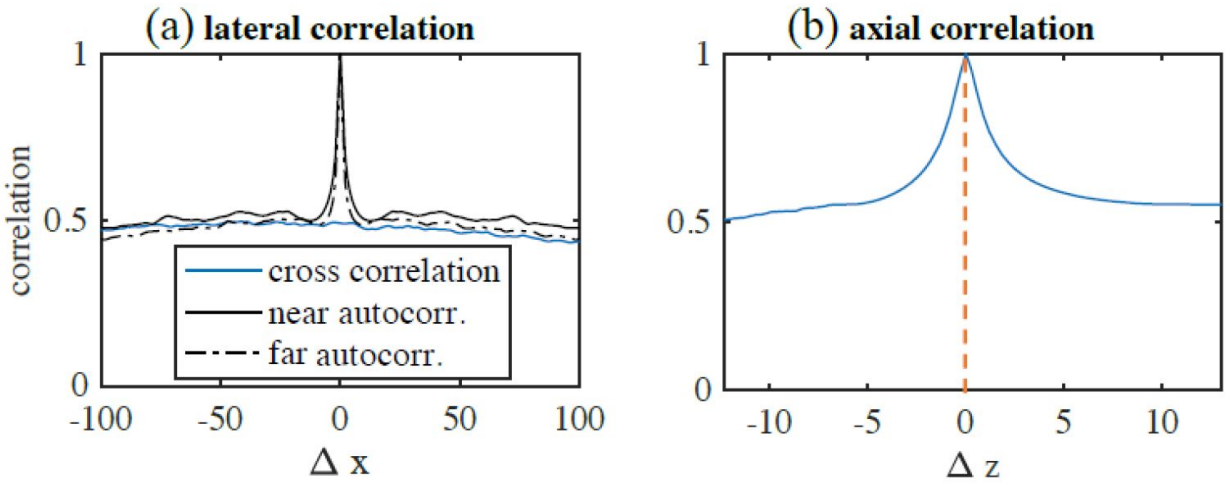
Compressed Sensing: Low Correlation Condition

- Does the low-correlation condition on H hold?



Compressed Sensing: Low Correlation Condition

- Does the low-correlation condition hold?



Caustics at a given depth are unique over shifting

Caustics from two different depths are not similar, even under translation

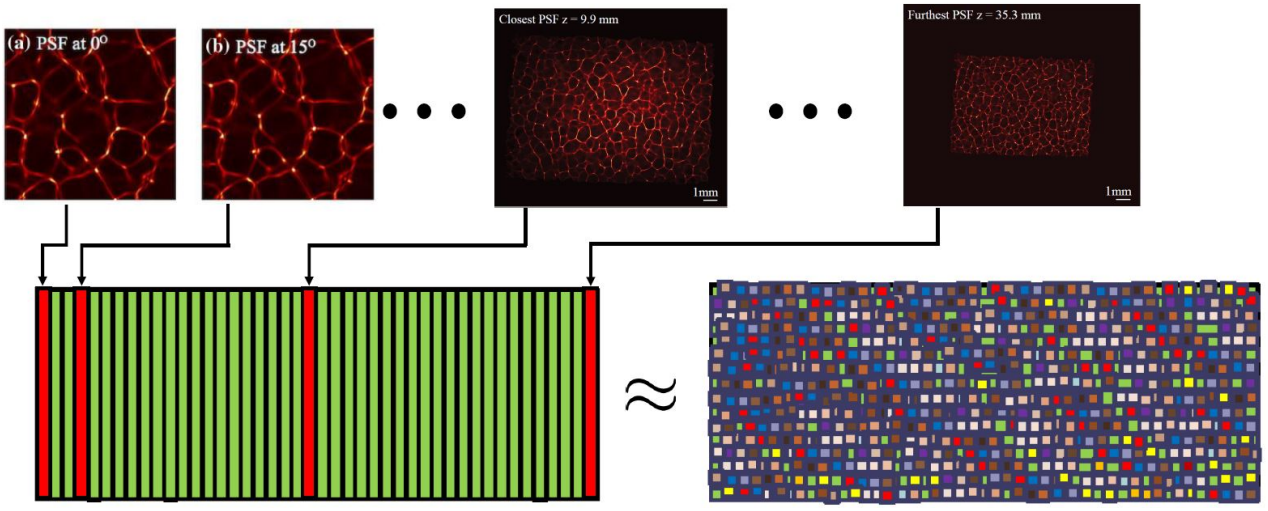
- Images taken from Optica 2018

Compressed Sensing: Low Correlation Condition

- Since the pattern is semi-random, moving it laterally (movement on axes x, y) will cause the autocorrelation to diminish
  - The diminishing rate is very fast
- Axial movement (on axis z, i.e. depth) will also cause the autocorrelation to diminish, since the random areas are widening/shrinking and don't overlap with each other.

Compressed Sensing: Low Correlation Condition

- Does the low-correlation condition hold?



- Images taken from Optica 2018

Compressed Sensing: Low Correlation Condition

- If we take the pattern under axial/lateral shifts, and take each impulse response as a column in H, H can be approximated as a random matrix and so the columns have low correlation.

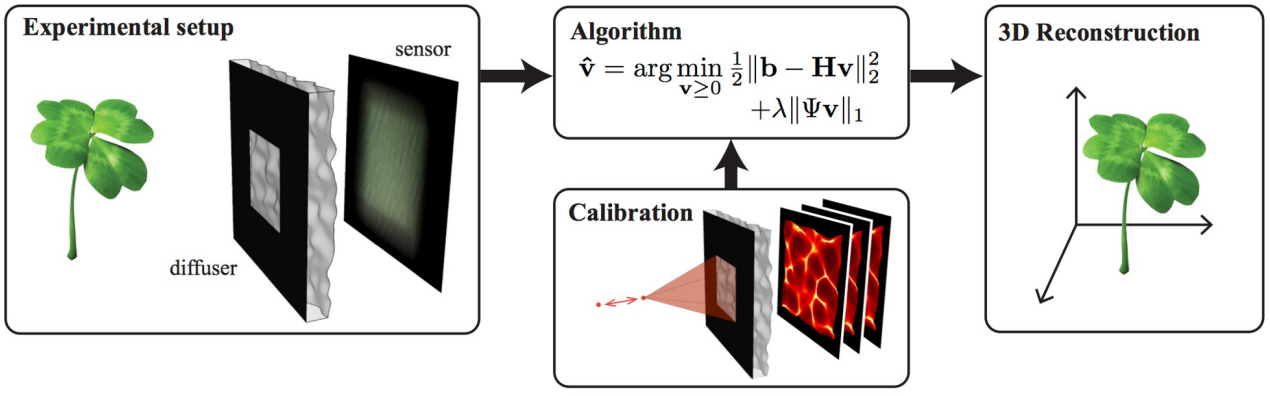
Compressive Sensing: Sparsifying Transform

- Maybe v isn't sparse?
  - Apply Ψ, a "Sparsifying Transform" to it.
  - Now Ψv is surely sparse, and the second condition holds.
- "Sparsifying Transform" - Gradient, Wavelet, Total Variation, Learned Dictionary, etc.

Gradient Magnitude

Compressive Imaging: Pipeline

- Overall framework:



- Image source - Optica 2018

Compressive Imaging: Summary

- Pros:
  - Clear and nice theory developed over ~20 years (Prof. Michael Elad @ CS, Prof. Yonina Eldar @ Weizmann)
  - Promising results in various fields: medical imaging, radar, signal processing, image processing, etc.
- Cons:
  - Hard to generalize to semantic tasks like classification/segmentation
  - Performance is usually limited at low SNR

Deep Optics

- Computer Vision Pipelines
- Differentiable Optics

Computer Vision Pipelines

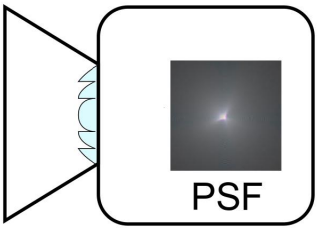
- How do CV pipelines work?

Objective: Solve CV problem on scene

What is this?

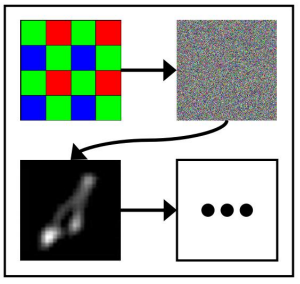
Computer Vision Pipelines

Step 1: Build camera



Optimize optics to minimize aberrations:
Blur/spot size, chromatic aberrations, distortions, …

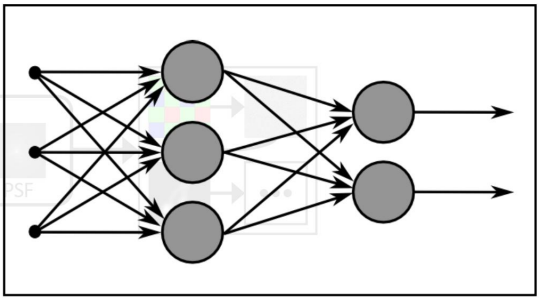
Step 2: Image Signal Processing (ISP)



Maximize PSNR:
Demosaicking, Denoising, Deblurring, …

- Take the bayer pattern (Each pixel only detects a single color), and apply demosaicing to get 3 colors per pixel
- Filter out demosaicing/measurement noise
- Filter out motion blur
- etc...
- We do all this to maximize the PSNR to get a clean and sharp images

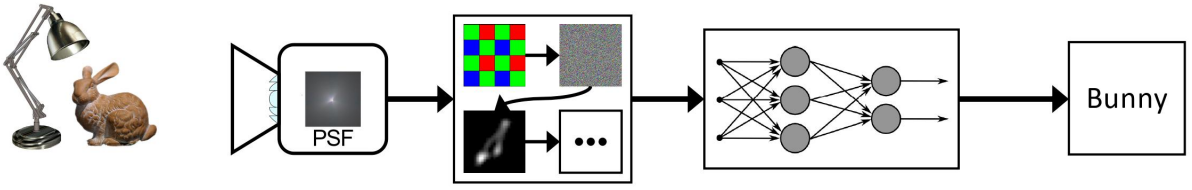
Step 3: CNN for Semantic task



Minimize semantic loss:
classification error, segmentation error, …

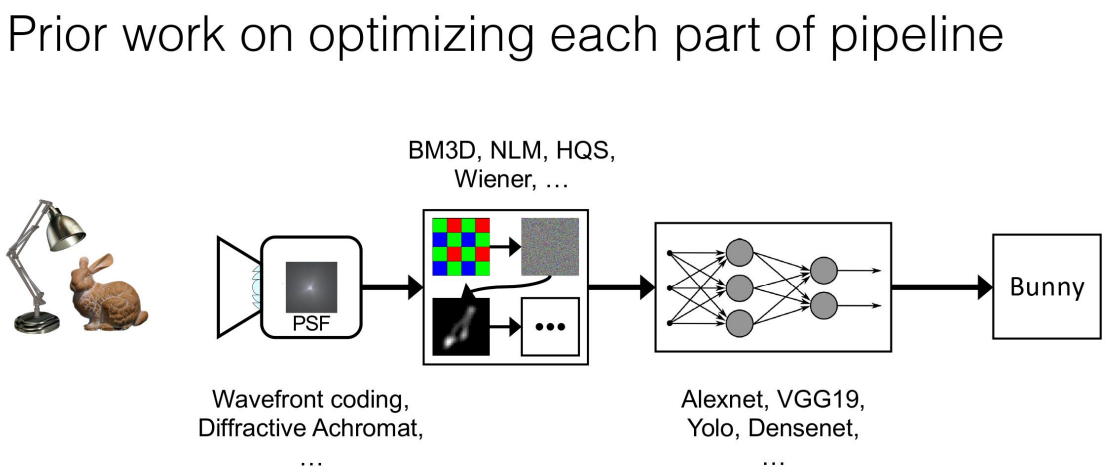
- How do we set the CNN? Training set of images and labels, and train the network.

It works!

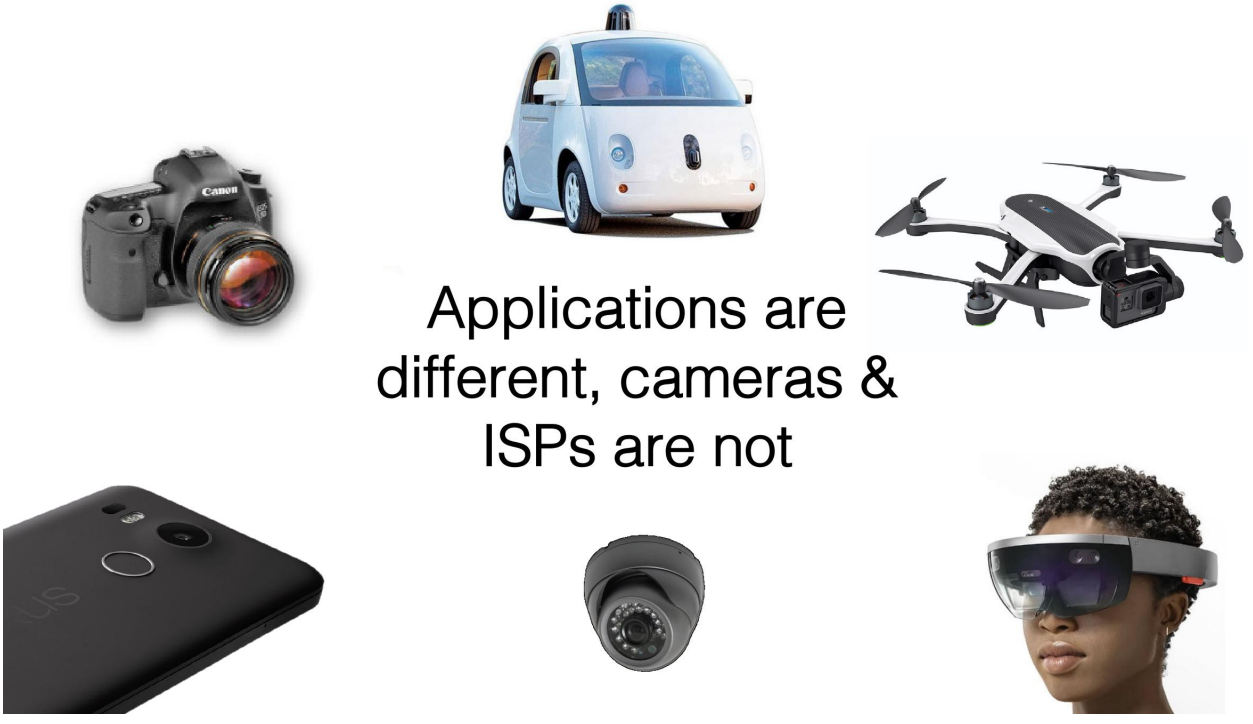


- Each part of this pipeline was designed seperately from the other

- Specifically - The first 2 parts are trying to create a clean and sharp image, without thinking about the network that comes afterwards. And vice versa - the CNN was designed with the knowledge that the dataset is composed of clean images.



Computer Vision Pipelines

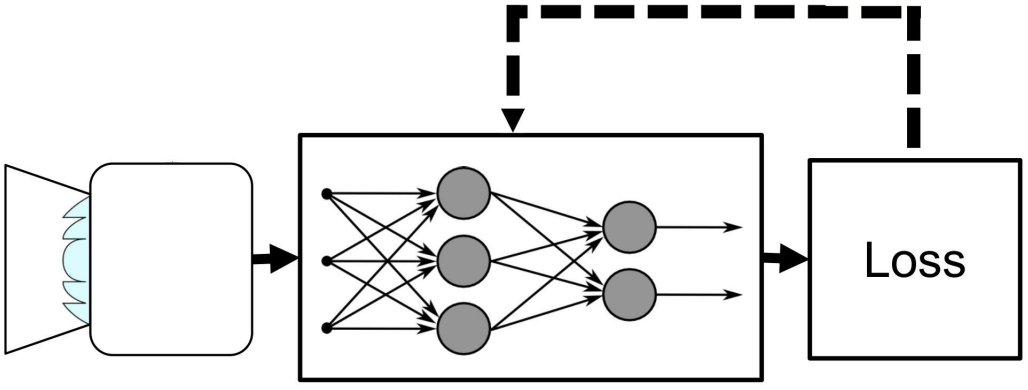


Computer Vision Pipelines

- Why is using the same "vision system" for different task is sub-optimal?
  - Animal vision is adapted to the surrounding and the day-to-day "task"

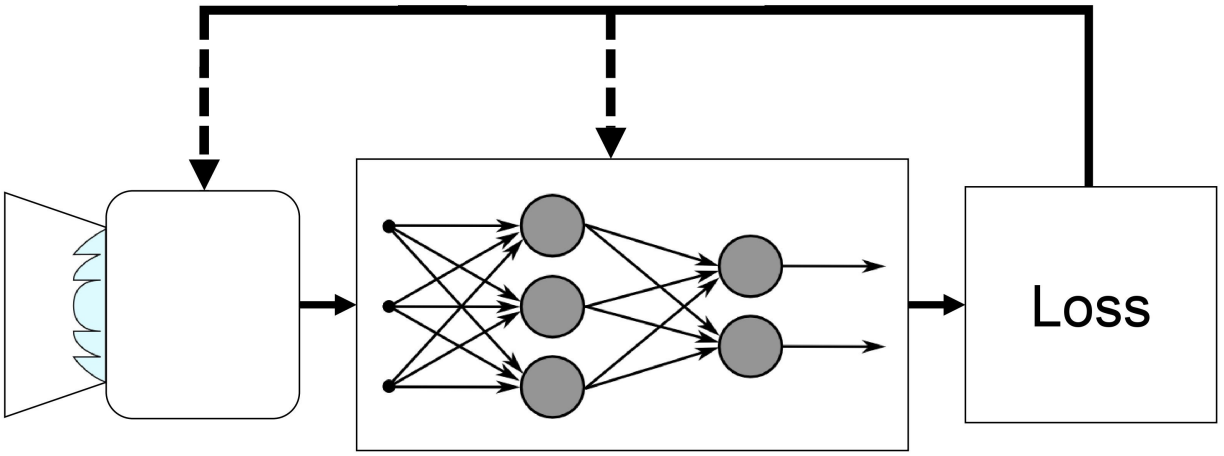
Computer Vision Pipelines

- In "standard" deep image processing, we have the task's loss, and we backpropagate to update the network's weights and improve its performance on the task.
  - This is done based on clean and sharp images that we get from the normal ISP



Computer Vision Pipelines

- Instead, in Deep Computational Imaging, we backpropegate those gradients up to the camera, and ensure the optics and the networks are optimized concurrently to improve the performance on the task.



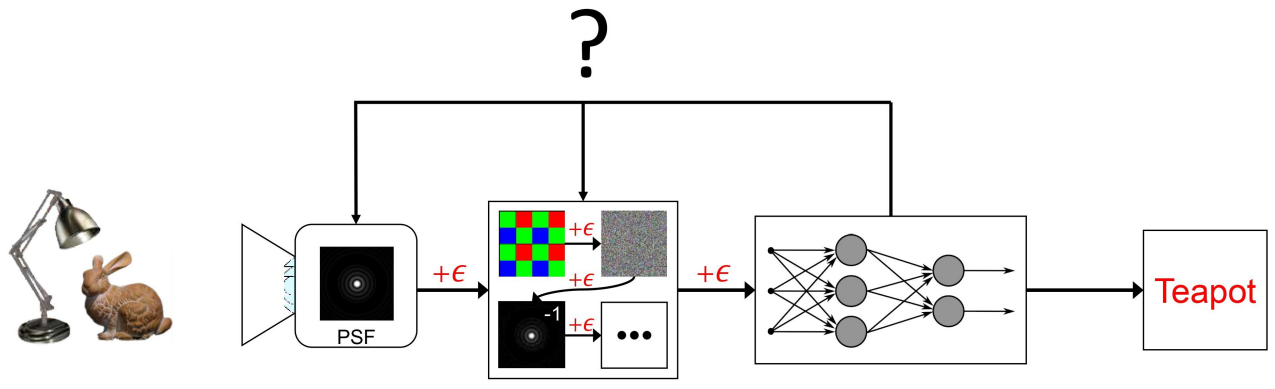
Optimize optics end-to-end with higher-level processing!

Computer Vision Pipelines

- Why is this better than optimizing them seperately?
  - Lets think about a system placed at the garden of a pet-daycare, and should open a different gate for the cats and the dogs.
  - The camera placed there would only ever see cats and dogs.
  - Using Deep Computation Imaging, the camera's optics would amplify the distinguishing features between cats and dogs
    - Filming any other object will result in weird and undistinguishable images. But the camera should anyways never see any other object!

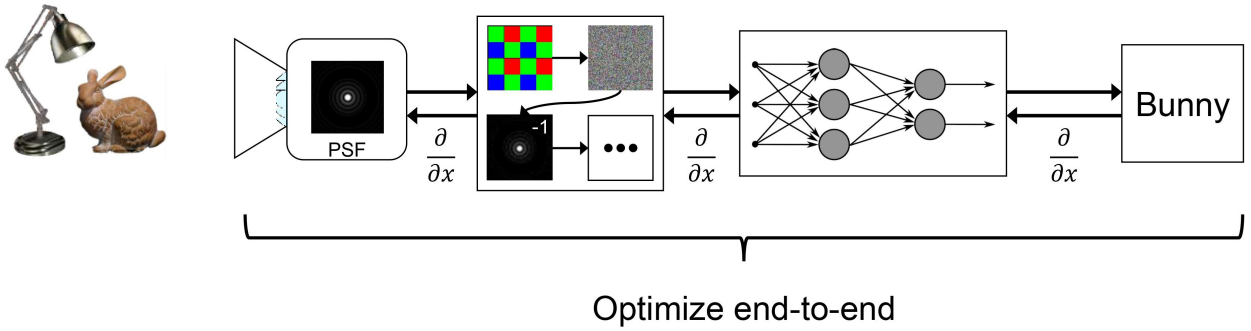
Computer Vision Pipelines

- Image classification with specialized "optics"
- At the start of the training loop, the classification will not be correct, as expected.
  - The gradients will backpropegate through the network to the optics.



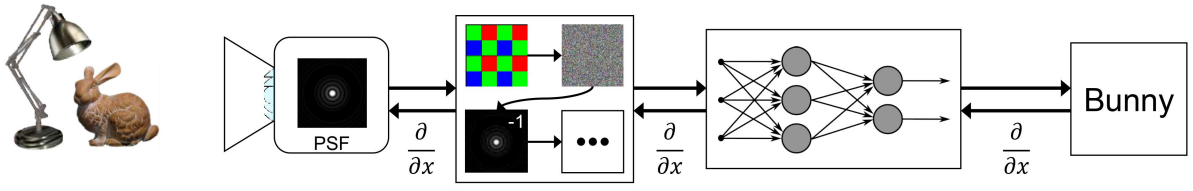
- Main idea: optimize the optics and the algorithm jointly to excel in the final task
  - We'll get new cameras dedicated to the task

Vision: The Deep Computational Camera



Computer Vision Pipelines

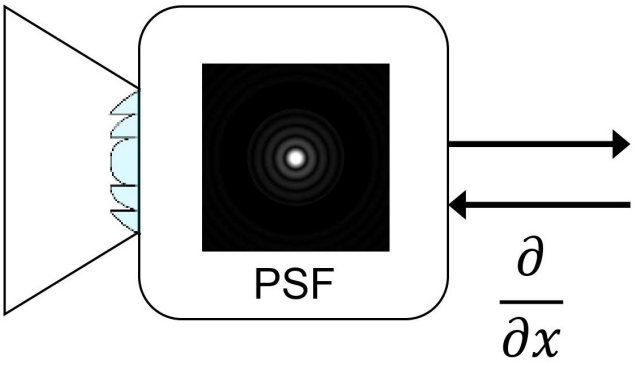
Vision: The Deep Computational Camera



- Performance & robustness gains
- Domain-specific hardware may reduce footprint, cost, power…
- New design space: The “BunnyCam”

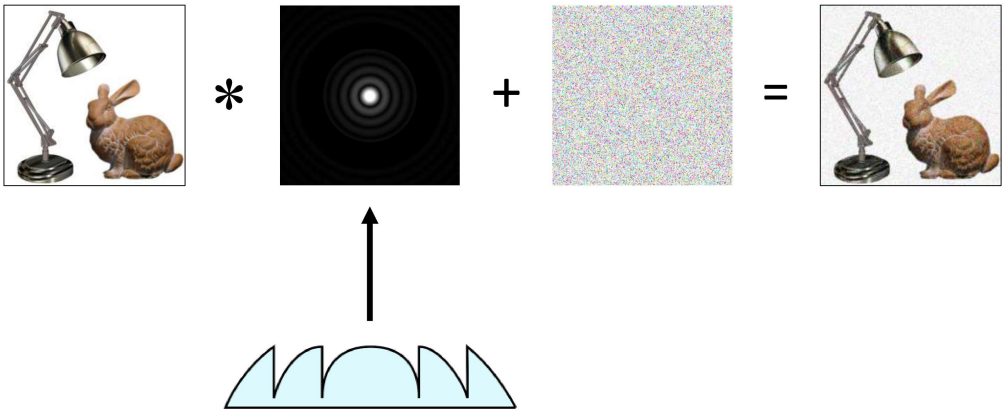
Differentiable Optics

- So far we drew an arrow in the diagram and said "the gradients will backpropagate"
  - How do we actually do it?
  - We model the camera as a **differentiable optical model**



Differentiable Optics

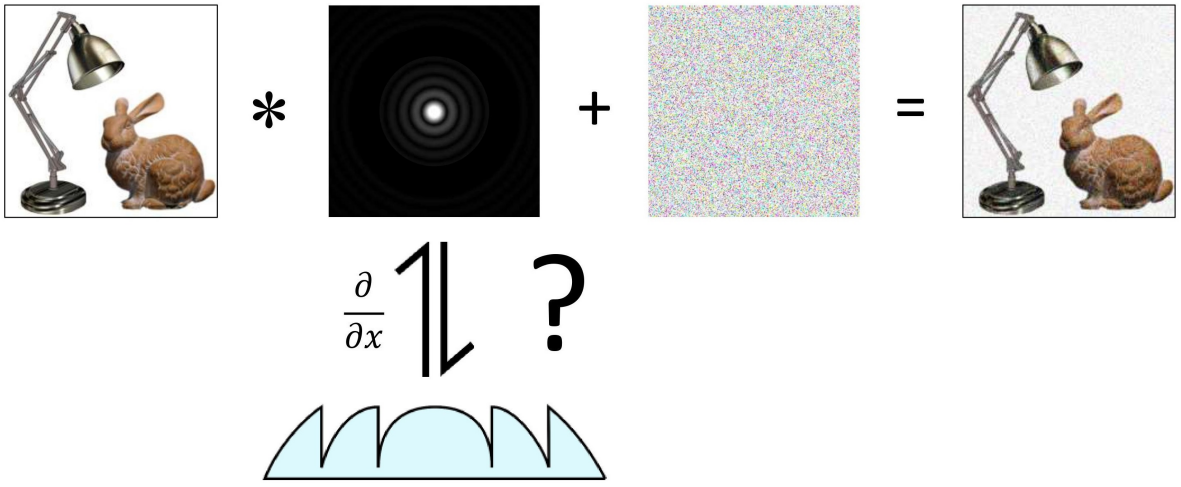
- Image formation model
  - A phase mask controls the impulse response (PSF) of a 3D point in the scene
  - The 3D scene is convolved (3D convolution) with this PSF
  - Noise is added to the output
  - This is projected onto a 2D sensor to create a 2D image



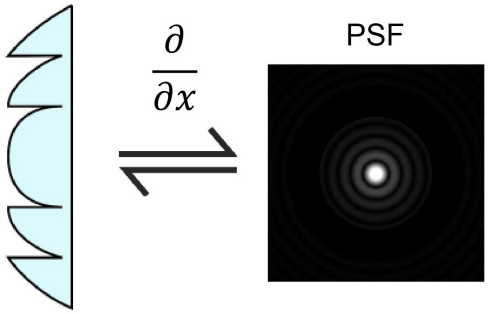
Differentiable Optics

- Now we know how to backpropegate up to the PSF (since we know a 3D convolution is differentiable, and an addition of noise is still differentiable)
- But how do we map the optical element to the PSF?

- How do we backpropegate through a physical lens?



Differentiable Optics

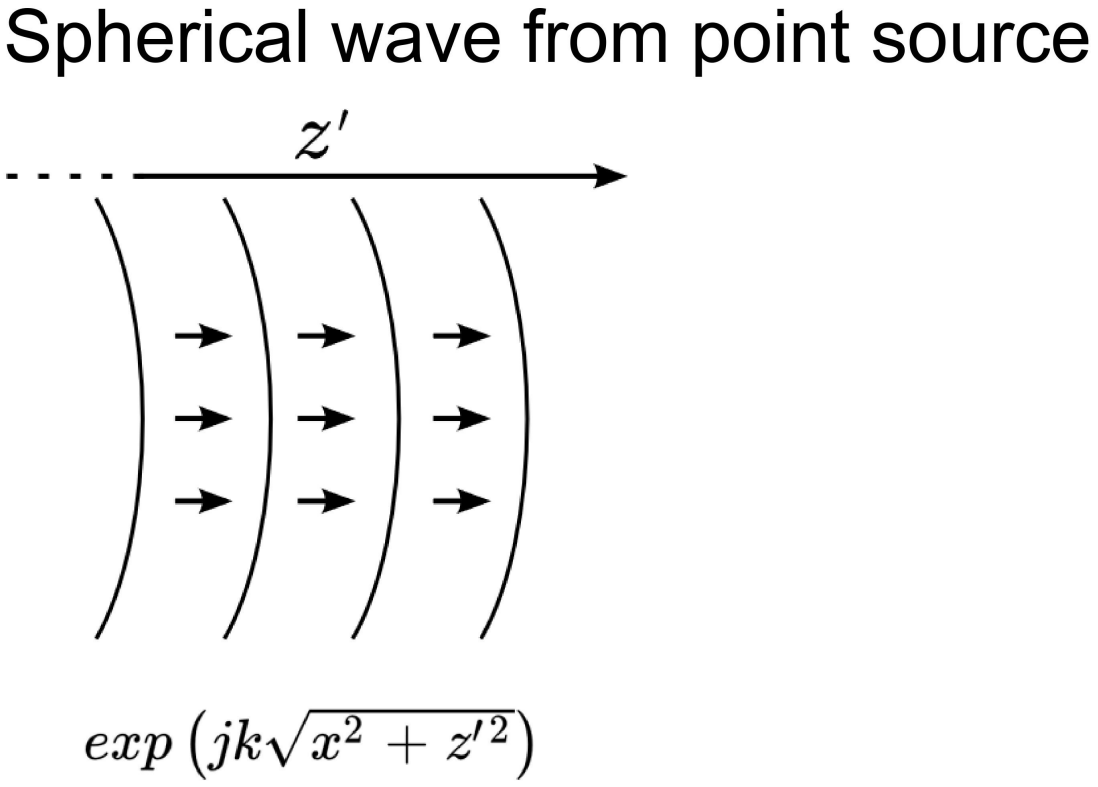


Wave Optics PSF simulator

Differentiable Optics

- Model the light coming from a point in the scene until it reaches the camera

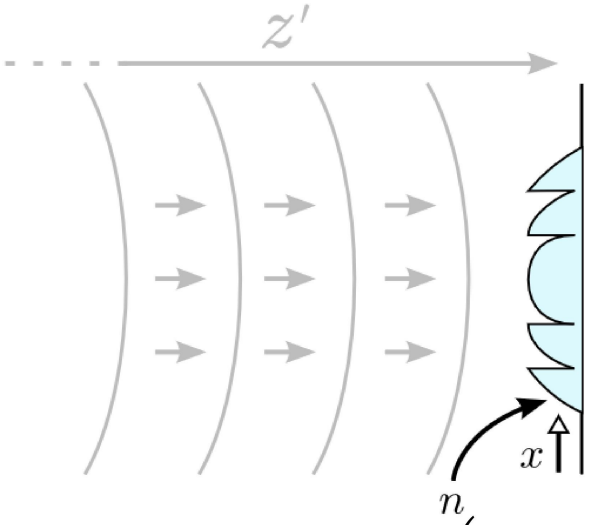
- The light gains a quadratic phase as long as it travels through space



Differentiable Optics

- When the light hits the phase mask, it gains a differnt amount of phase depending on where it hits the phase mask, since every point in the phase mask has a different depth.
  - This phase mask will define an impulse response, based on the refractive index of the phase mask and the depth at each point

Phaseshift by optical element

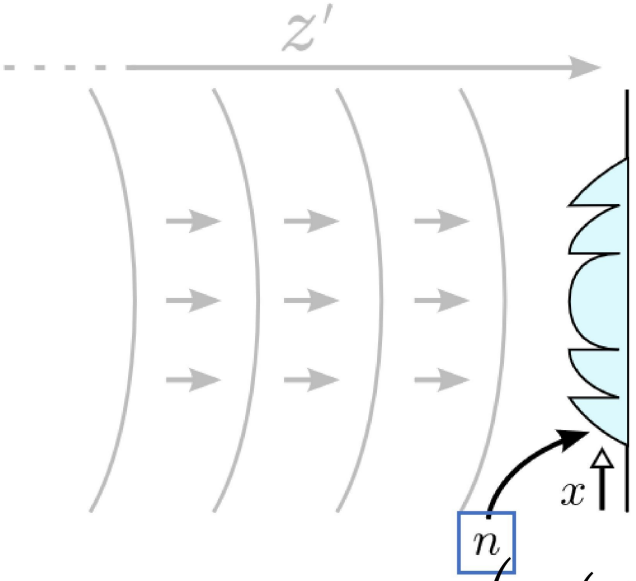


$$U(x) = \exp\left(jk\left(\sqrt{x^2 + z'^2} + (n-1)\Phi(\mathrm{x})\right)\right)$$

Differentiable Optics

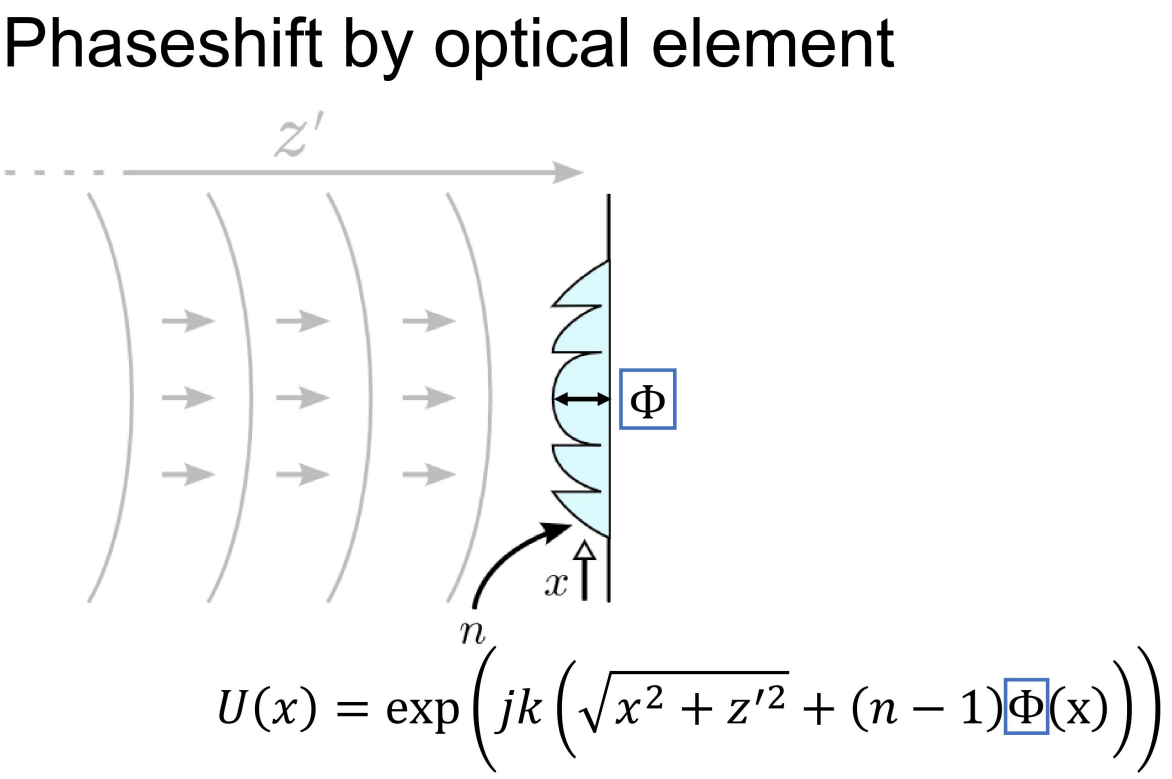
- The refractive index dictates a multlipicative scalar on the gained phase

Phaseshift by optical element



$$U(x) = \exp\left(jk\left(\sqrt{x^2 + z'^2} + (n-1)\Phi(\mathrm{x})\right)\right)$$

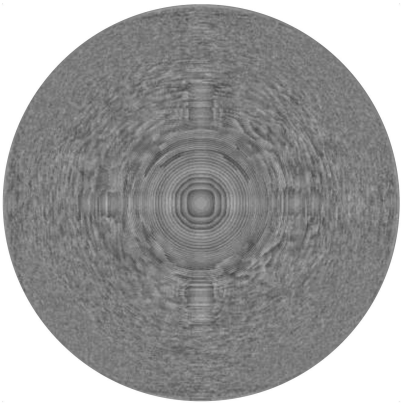
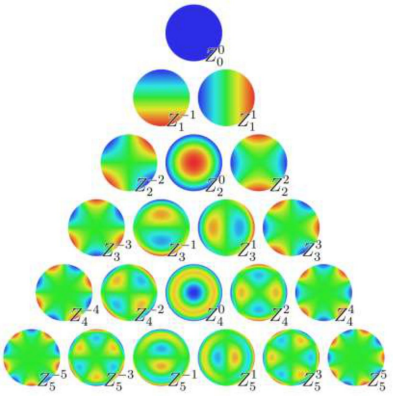
- The different depth at each point, modeled as the **height profile** $\Phi(x)$ of the phase mask, will of course also affect the phase gained by the light, and the autocorrelation of the light when it reaches the sensor



Differentiable Optics

- How to model the height profile along a 2D space?
  - We can model it per-pixel, i.e. each point in the phase mask
  - Or we can create a spanning basis of the profile.

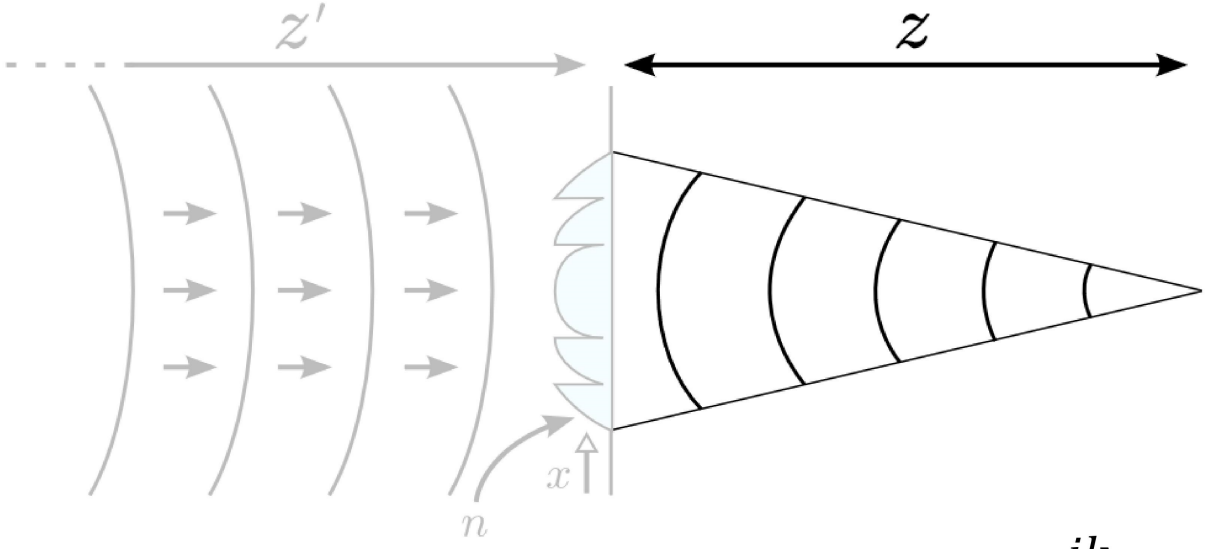
- Generally: we have efficient ways to model it.

Height Map parameterization (diffractive)	Zernike basis parameterization (refractive)
	
$\Phi[x] = [[a_{11}, a_{12}, \dots], \dots]$	$\Phi[x] = \sum Z_i^j[x] \cdot a_{ij}$

Differentiable Optics

- After the light has passed through the phase mask, it continues to propegate through empty space (from the phase mask to the sensor)
- This propagation is approximated by scalar optics with Fresnel's approximation

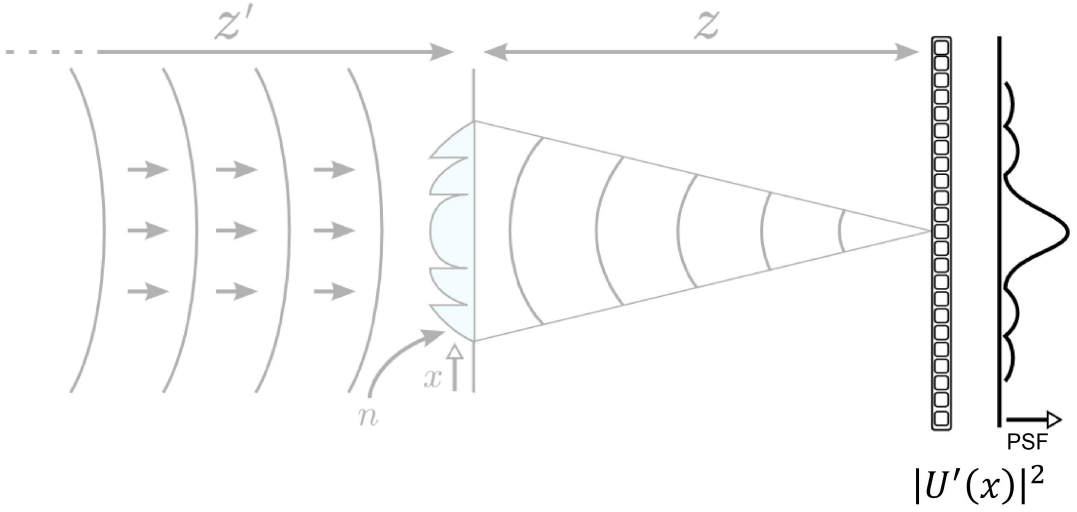
Fresnel propagation to sensor



$$U'(x) = U(x) * \exp(\frac{jk}{2z}x^2)$$

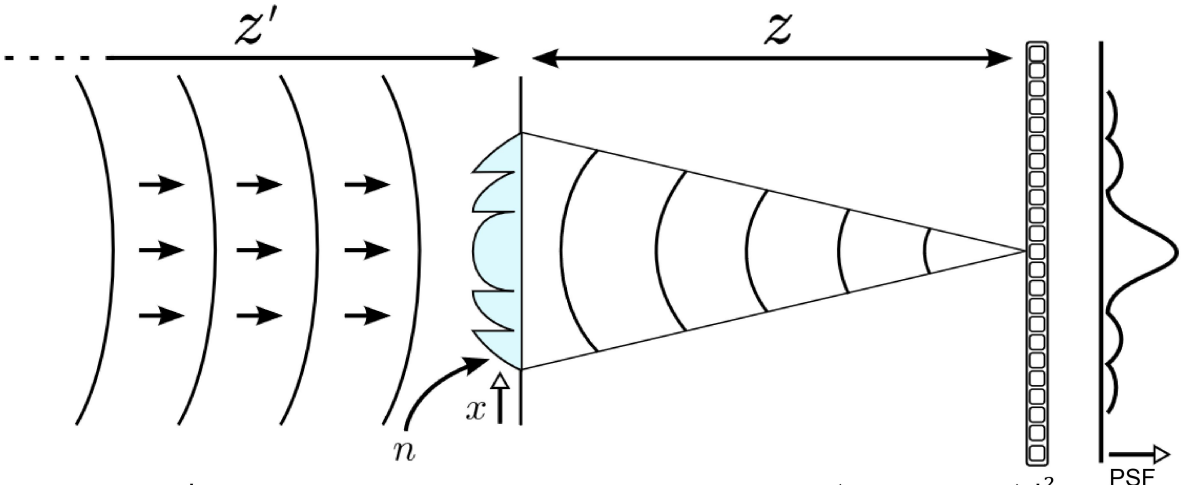
- What we measure at the sensors is the intensity of the light that reached the sensors.
  - The intensity is the squared norm of the field

Intensity measurement at sensor



Differentiable Optics

Calculating the PSF



$$\rho_{z',\lambda} = \left| \exp\left(jk \left(\sqrt{x^2 + y^2 + z'^2} + (n-1)\phi(x,y) \right) \right) * \exp\left(j \frac{k}{2z} (x^2 + y^2) \right) \right|^2$$

Differentiable Optics

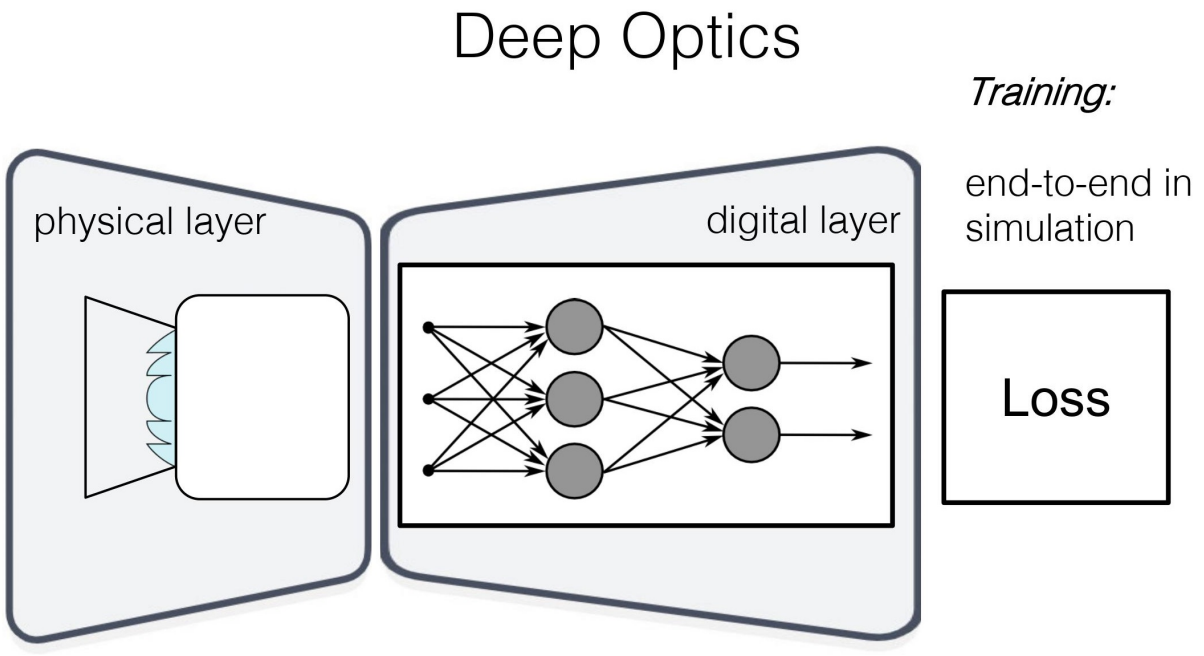
- Finally: this modeling process gave us an impulse response dependant on the:
  - unknowns:
    - z' - the depth of the point in the scene
    - $k = \frac{2\pi}{\lambda}$ - the wavelength of the incoming light from that point
  - known:
    - z - the distance of the sensor from the phase mask
    - $\Phi(x, y)$ - the phase mask we can change to dictate diffent phases gained based on depth
- All of this is differentiable!

Differentiable Optics

- Now it's easy to train this with a network!
  - We have an differentiable optic simulator
  - We have a differentiable NN
  - We can train end-to-end using simulations of 3D scenes.

Differentiable Optics

- Let us summerize with a classification task!

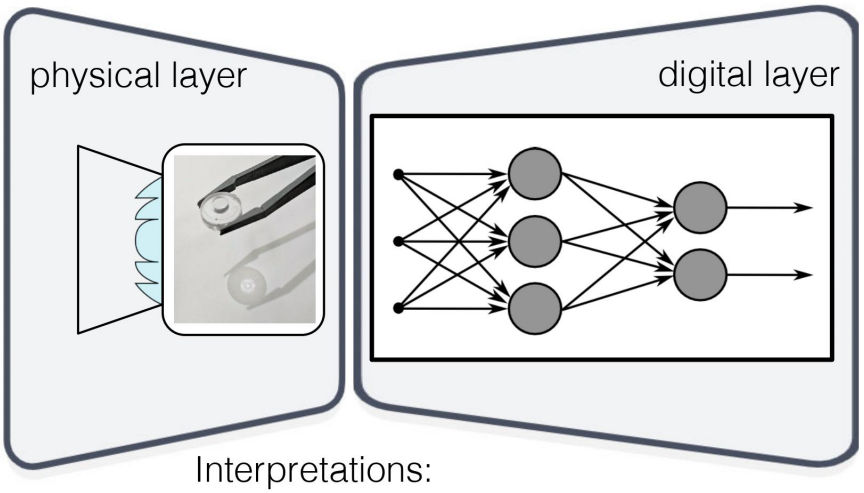


Differentiable Optics

- At test time, we throw away the simulator, create the phase mask, insert it into a real camera, and take pictures of a real 3D scene!
  - Now we can finetune the NN with real physics.

Deep Optics

Inference:



fabricate lens or other physical components, run network

Interpretations:

- Optical encoder, electronic decoder system
- Hybrid optical-electronic neural network

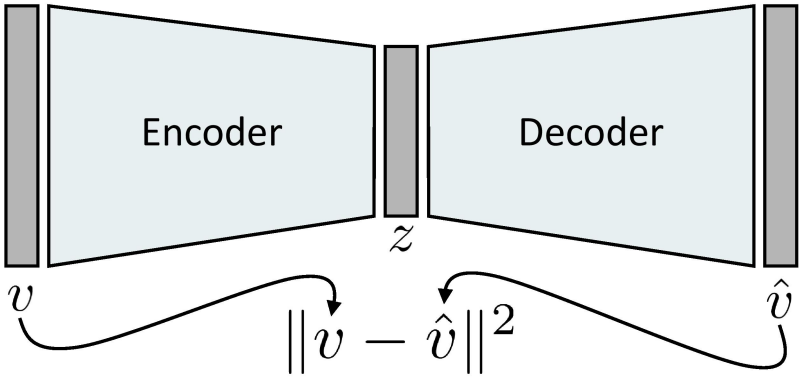
Differentiable Optics

- You can think of the system as an encoder-decoder system:
  - Where a 3D scene is inserted on one side, encoded into a 2D image with the camera
  - The 2D image is the "latent space"
  - Then the latent space is decoded back to a 3D scene via a NN.

Differentiable Optics

Autoencoders: Background

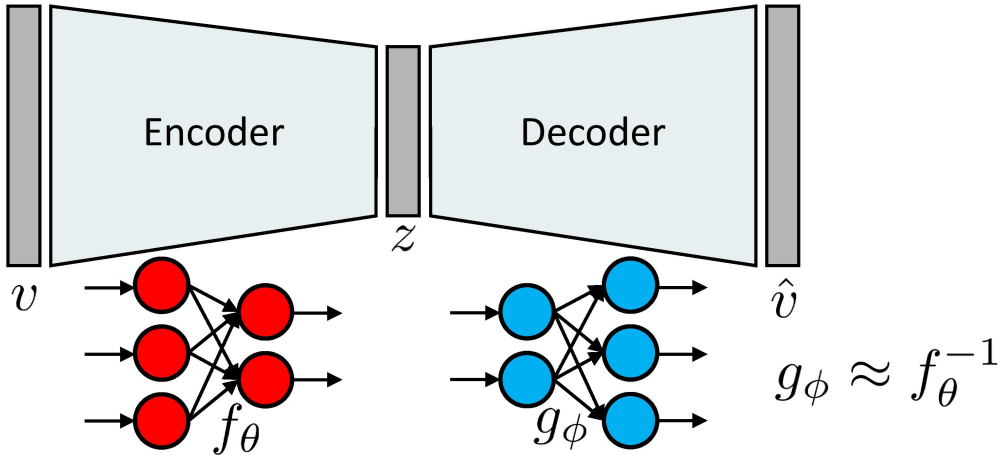
- Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data



- Train such that features can be used to reconstruct original data

Autoencoders: Background

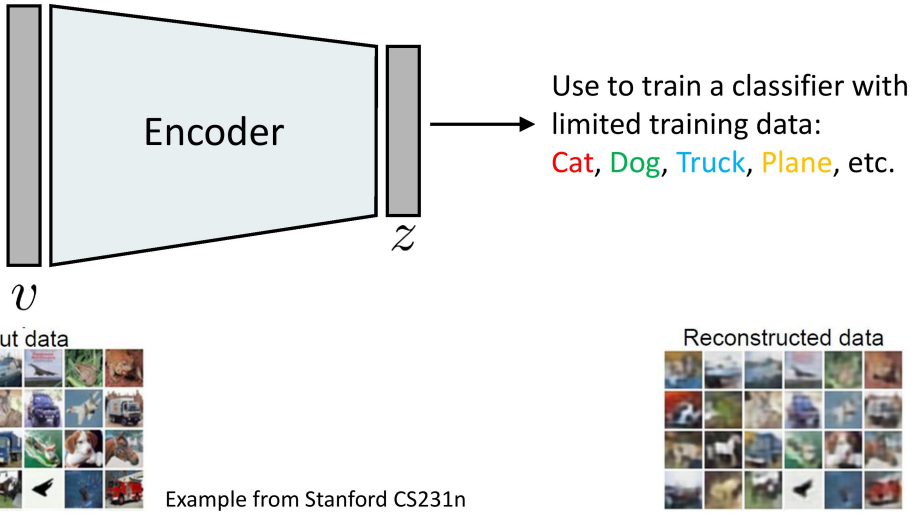
- Unsupervised approach for learning a lower-dimensional feature representation from unlabeled training data



Differentiable Optics

Autoencoders: Background

- Learned lower-dimensional representation can be used for classification

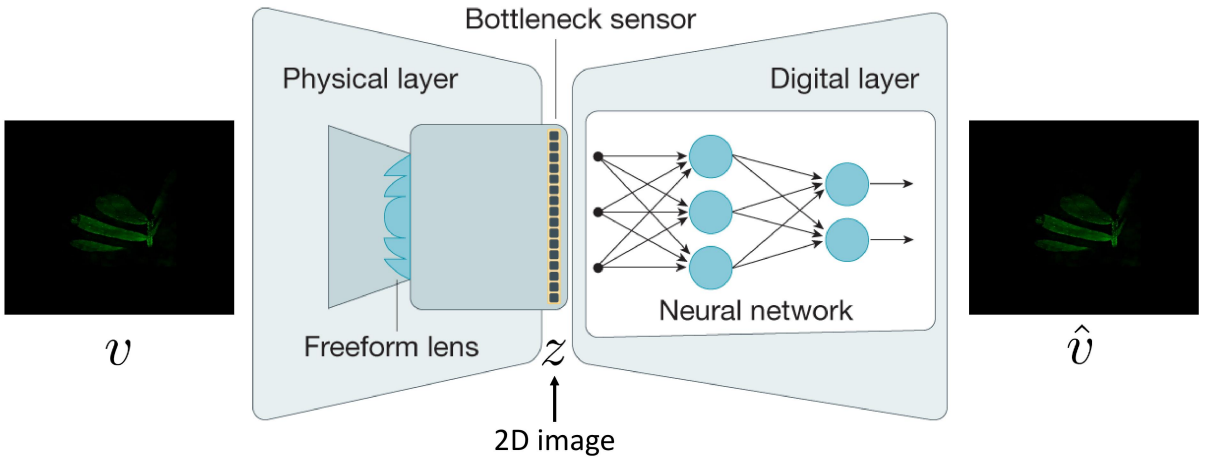


Example from Stanford CS231n

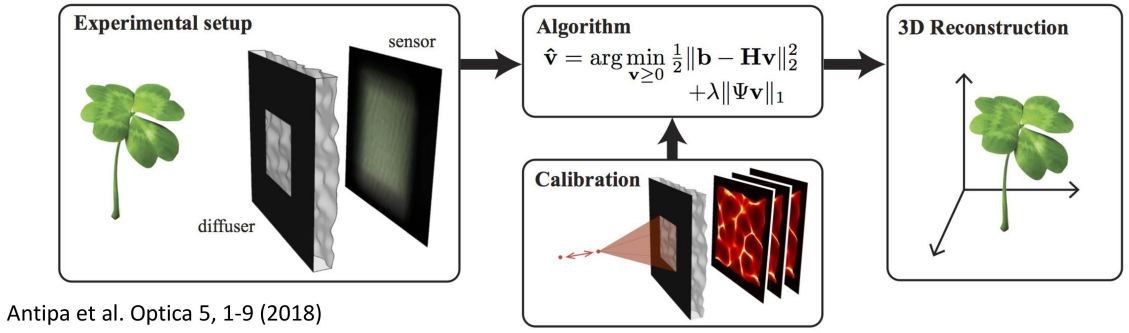
Differentiable Optics

Autoencoder interpretation: Physical encoder – Electronic decoder

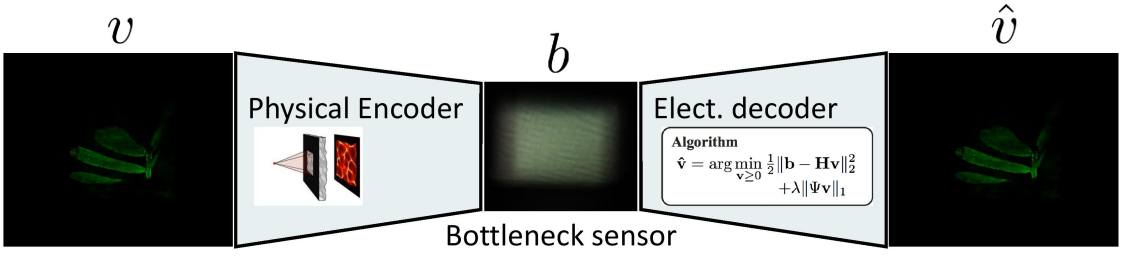
- For example, we can optimize the “Freeform” lens for depth imaging



Analogy to Compressed Sensing



Antipa et al. Optica 5, 1-9 (2018)

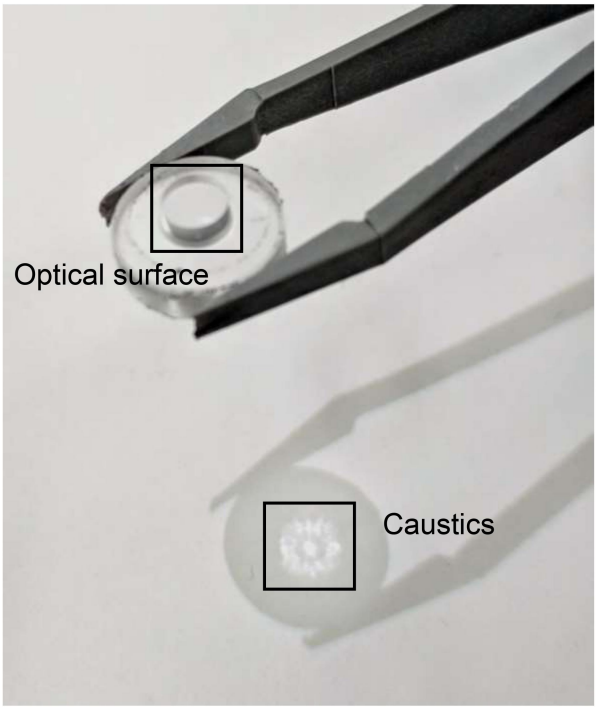


- Compressed Sensing Logic: recovery algorithm is based on a sparsity regularization → optics need to result in an incoherent matrix.
- Deep Optics Logic: NN is a superior algorithm to MAP estimation with a sparisty regularization → optics should be designed to allow NNs to further excel!

Applications

- Extended Depth of Field
- Monocular Depth Estimation / Depth from Defocus
- High Dynamic Range Imaging
- Video Compressive Sensing
- Computational Microscopy (Will not show examples)

Application 1: Extended Depth of Field (EDOF)



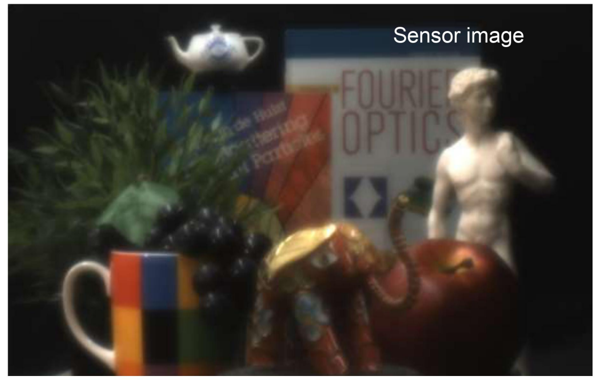
Refractive Achromatic EDOF element

- Polymethyl methacrylate (PMMA)
- 5 mm aperture size
- Sensor distance $35.5mm$
- F-number 7.1
- One active optical surface
- Feature size $3.69\mu m$

- Image source - ACM SIGGRAPH 2018

Application 1: Extended Depth of Field (EDOF)

Test scene



Regular bi-convex lens Optimized lens

Elephant (0.5m) → Book (2.0m)

- Image source - ACM SIGGRAPH 2018
- Both look blurry!

Application 1: Extended Depth of Field (EDOF)

- Remember the NN the deep computational camera must pass through!

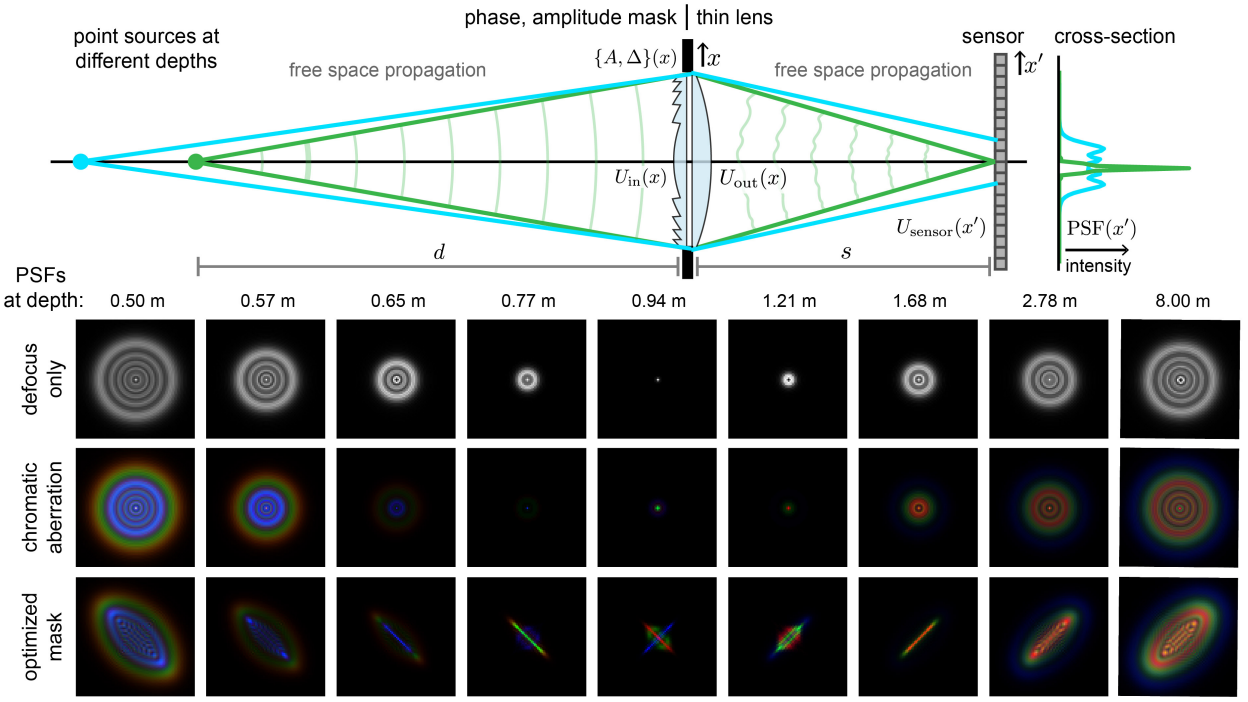
Test scene

Regular bi-convex lens Optimized lens

Elephant (0.5m) → Book (2.0m)

- Image source - ACM SIGGRAPH 2018

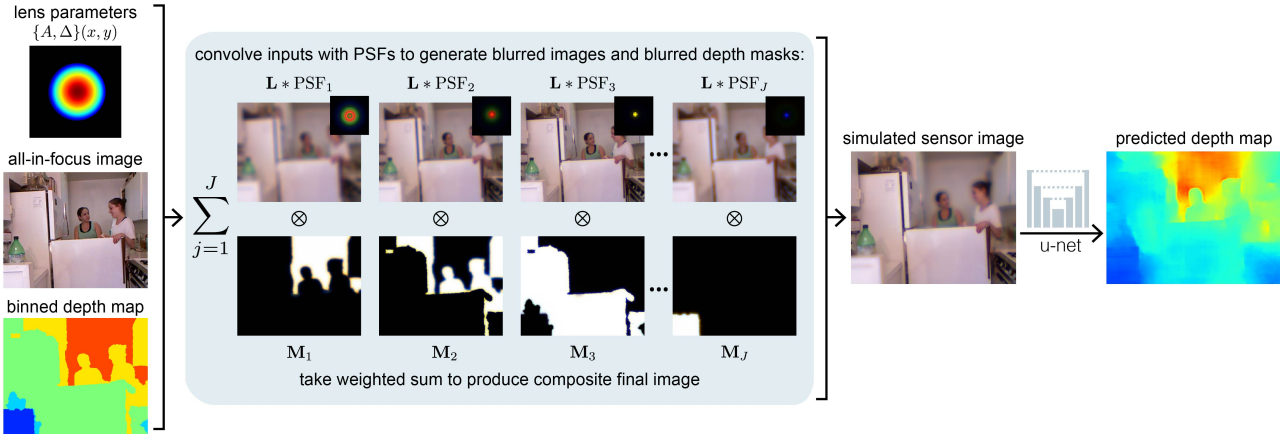
Application 2: Monocular Depth Estimation



- Image source - ICCV 2019

Application 2: Monocular Depth Estimation

- The PSF is depth dependant.
  - At training we have a depth map as well.
  - Here the input is an image, and the depth map gives us the mask needed to simulate the depth affected PSF, to created the simulated 2D image.



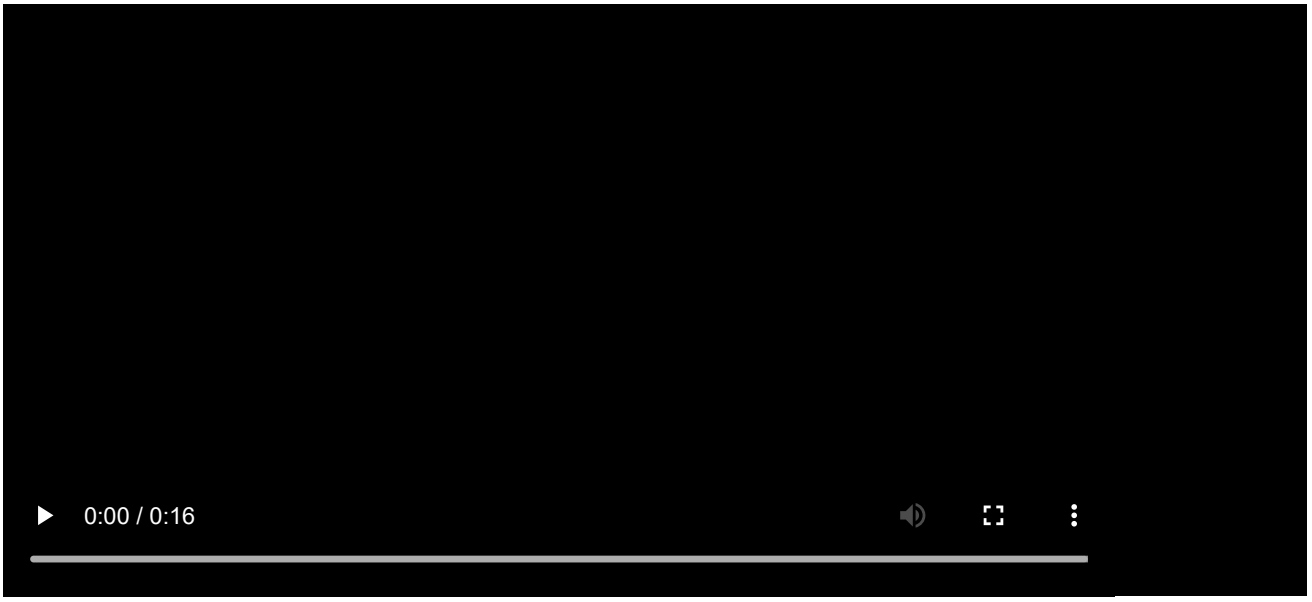
- Image source - ICCV 2019

In [12]:
```
from IPython.display import Video
Video("./assets/input_vid_dfd.mp4")
```

Out[12]:

In [10]:
```
Video("./assets/output_vid_dfd.mp4")
```

Out[10]:

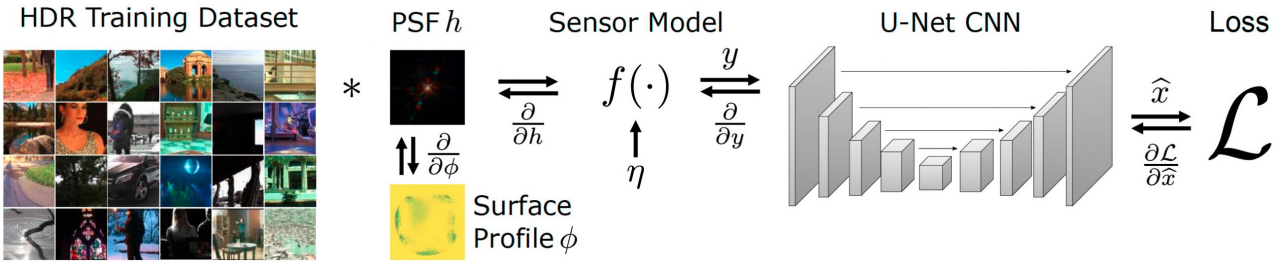


In [13]:
```
Video("./assets/bb_3d.mp4")
```

Out[13]:

Application 3: High Dynamic Range Imaging

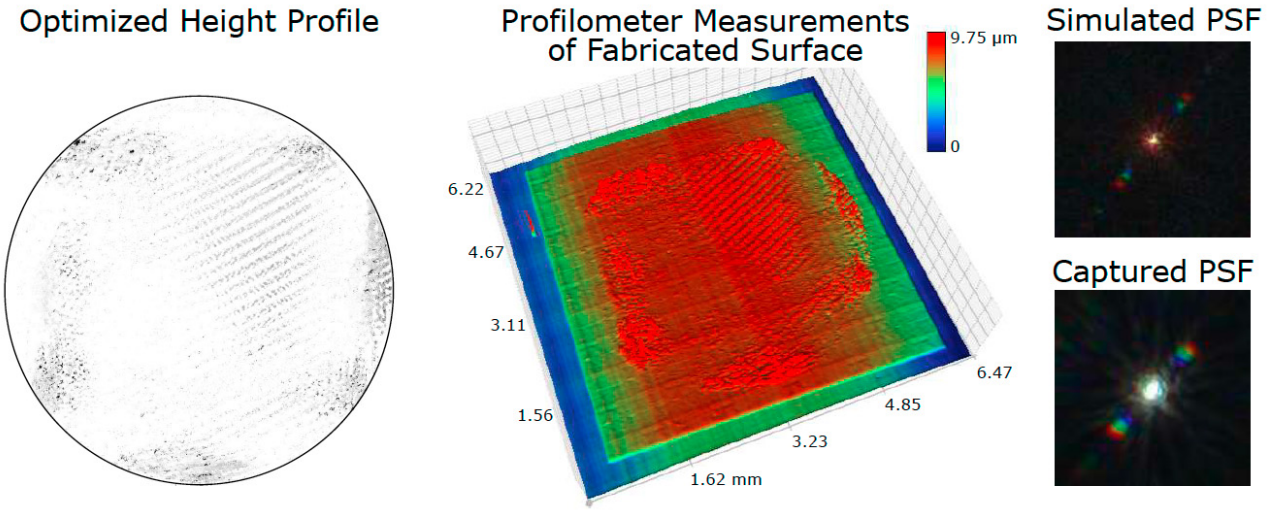
- Usually the sensor model is cutting off the dynamic range, so when trying to photograph high-dynamic range we get saturation in different parts of the scene.
- In HDR, we take 3 different photos with different dynamic ranges, and combine them to get a single high-dynamic range photo.
- Can we optimize a phase mask such that the camera will create a single image from which we can get an HDR image?



- Image source - CVPR 2020

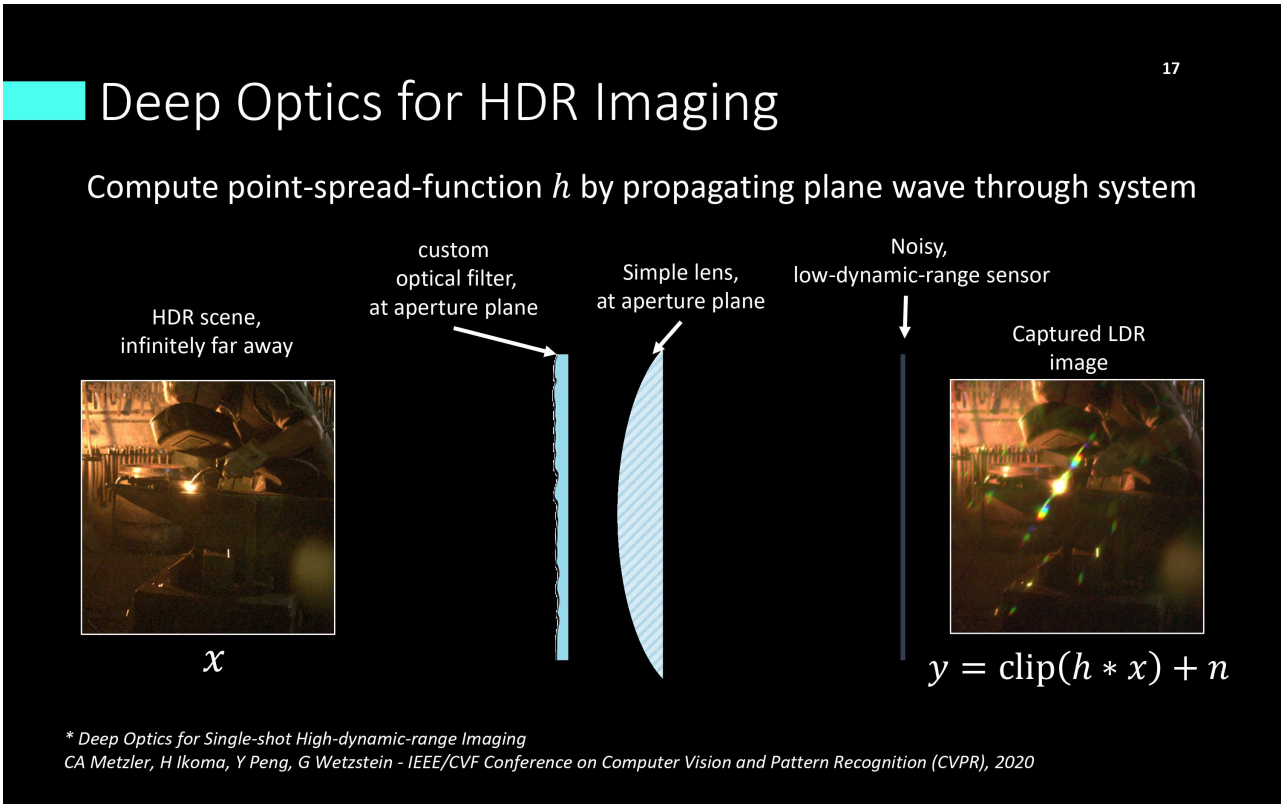
Application 3: High Dynamic Range Imaging

- If the optics is creating copies of the scene at lowered intensity.
  - Each point in the image is captured as 3 points, where the side lobes are at a lower intensity than the middle lobe.
  - This means if the middle point is in saturation, there are still 2 copies of that point that aren't in saturation.



- Image source - CVPR 2020

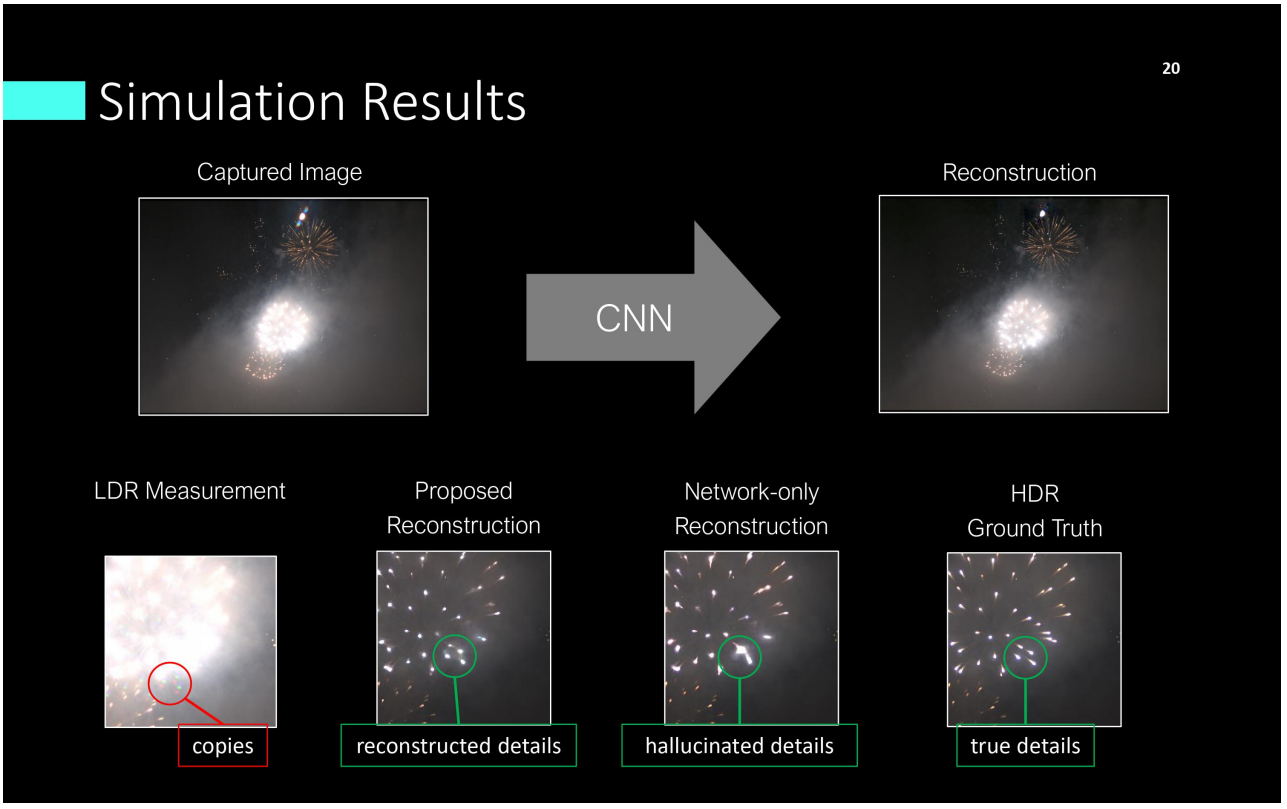
Application 3: High Dynamic Range Imaging



- Image source - CVPR 2020

Application 3: High Dynamic Range Imaging

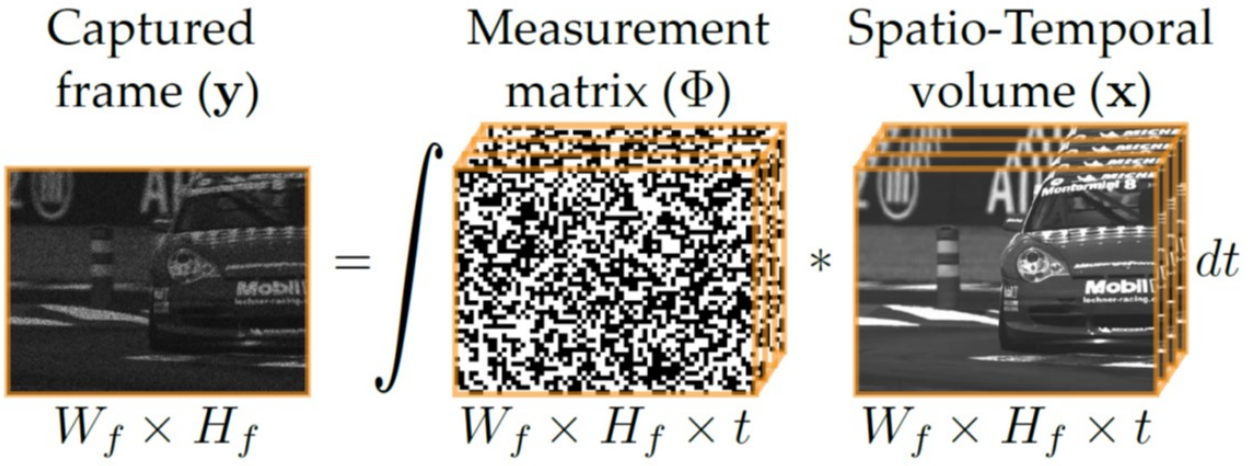
- After inserting the captured image to a NN we get an HDR reconstruction
  - Notice that the normal NN trained on saturated images and HDR groud truth can't reconstruct well, since the data is missing from a normal image - and therefore it's hard for the network to "understand" what's going on there



- Image source - CVPR 2020

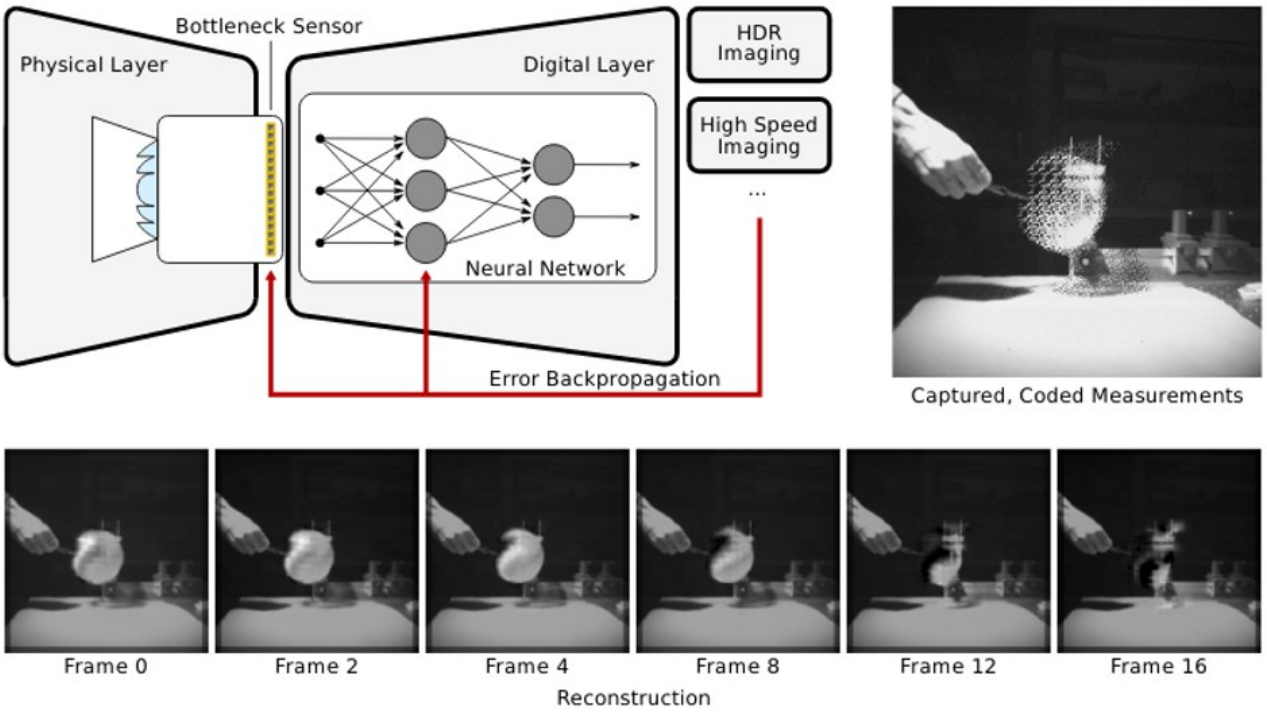
Application 4: Video Compressive Sensing

- Try to get from a single frame the sub-frames
- Challenge is to learn the binary masks Φ to recover video x from snapshot y
  - Φ dictates which pixels are captured at each moment (sub-frame)



- Image source - DSP Magazine 2020

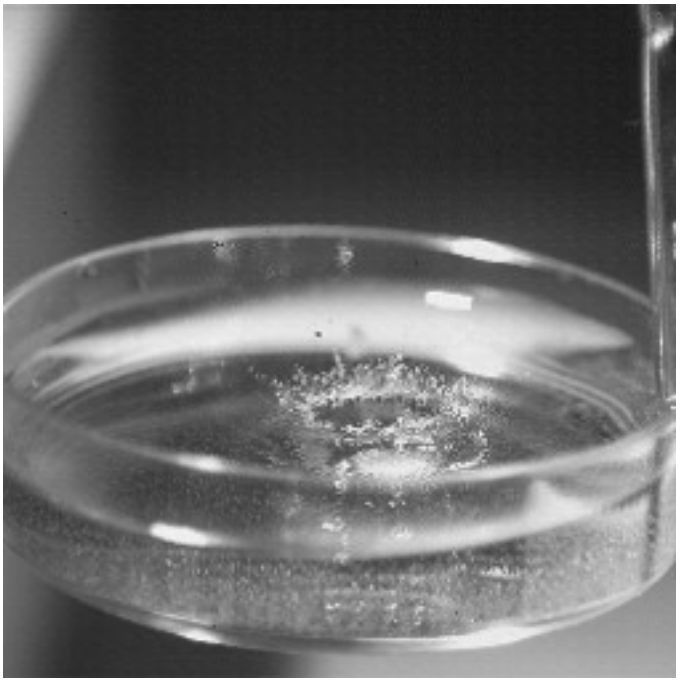
Application 4: Video Compressive Sensing



- Image source - ICCP 2020

Application 4: Video Compressive Sensing

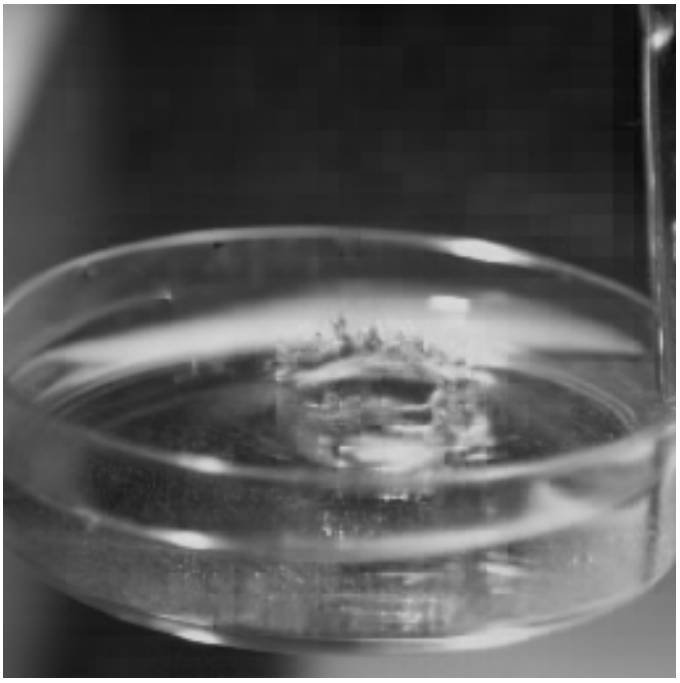
- captured image



- Image source - ICCP 2020

Application 4: Video Compressive Sensing

- 64 frames recovered from 4 measurements (16 frames/capture)



- Image source - ICCP 2020

Available resources online

- https://github.com/computational-imaging/opticalCNN
- https://github.com/vsitzmann/deepoptics
- https://github.com/computational-imaging/DeepOpticsHDR
- https://github.com/EliasNehme/DeepSTORM3D
- https://github.com/computational-imaging/DepthFromDefocusWithLearnedOptics
- etc.

Recommended Videos

Warning!

- These videos do not replace the lectures and tutorials.
- Please use these to get a better understanding of the material, and not as an alternative to the written material.

Video By Subject

- End-to-end Optimization of Optics and Image Processing - Vincent Sitzmann
- Neural Sensors - J.N.P Martel
- High Dynamic Range Imaging - Christopher Metzler
- 3D Single Molecule Localization Microscopy - Elias Nehme
- Towards Neural Imaging & Signal Processing - Gordon Wetzstein

Credits

- End-to-end optimization of optics and image processing - Vincent Sitzmann
- ACM SIGGRAPH 2020 Courses - Yifan (Evan) Peng, Ashok Veeraraghavan, Wolfgang Heidrich, Gordon Wetzstein
- Stanford EE367/CS448I (Computational Imaging and Display) - Gordon Wetzstein
- IEEE SPACE Webinar - IEEE Computational Imaging TC
- Research papers:
    - End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging
    - Deep Optics for Monocular Depth Estimation and 3D Object Detection
    - Deep Optics for Single-shot High-dynamic-range Imaging
    - Neural Sensors: Learning Pixel Exposures for HDR Imaging and Video Compressive Sensing With Programmable Sensors
    - Convolutional neural networks that teach microscopes how to image
    - DeepSTORM3D: dense 3D localization microscopy and PSF design by deep learning
    - Depth From Defocus With Learned Optics
    - etc.
- Icons from Icon8.com - https://icons8.com