Visual Sensing Using Machine Learning

Vivek Boominathan
Email: vivek.booms@gmail.com
Webpage: vivekboominathan.com

Ashok Veeraraghavan
http://computationalimaging.rice.edu/

Rice University
Tagging computational algorithms with optical systems has expanded visual sensing ability.
Data-driven Visual Sensing

- Visual data lie in a complex high-dimensional space that is hard to model analytically.
- Data-driven approach allows to “learn” a good approximate model from visual data.
- The “learned” model can improve the performance of visual sensing systems.
Data-driven “learned” models can improve the quality of visual sensing systems.
Visual Sensing using Machine Learning

Part II: Joint design with ML

Data-driven

Optical System → Machine Learning

Joint design with data-driven techniques can bring out best in both systems

Output

Improve Quality
Visual Sensing using Machine Learning

Part III: ML with Optical System

Incorporating optical layer(s) into machine learning system can decrease latency and power
Visual Sensing using Machine Learning

Part I: Backend ML

Part II: Joint design with ML

Part III: ML with Optical System

Optical System

Machine Learning System

Optical Layer(s)

Electronic Layer(s)
## Image Priors

<table>
<thead>
<tr>
<th>Hand-crafted priors</th>
<th>Data-driven priors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Images as signals</td>
<td>Captures complex “visual” priors</td>
</tr>
<tr>
<td>Signal statistics priors</td>
<td>Not easily describable as signal statistics</td>
</tr>
<tr>
<td>E.g. Sparse gradients, self-similarity</td>
<td>E.g. Perceptual feature maps, adversarial</td>
</tr>
</tbody>
</table>

Capturing good “Visual” priors lead to high-quality reconstructions
Example 1:

Solving Inverse Compressive Imaging Problems using Deep Pixel-level Prior

Akshat Dave, Anil Kumar Vadathya, Ramana Subramanyam, Rahul Baburajan, and Kaushik Mitra.
“Solving inverse computational imaging problems using deep pixel-level prior.”
Compressive Optical System

Single Pixel Camera (SPC)

Line Sensor Camera (LiSens)

Spatially multiplexing to reduce sensor cost e.g. for non-visible wavelength

Baraniuk et al., 2006
Wang et al., 2015
Ill posed inverse problem

$Y = AX$

Forward model

Measurements $<\ll$ Image size

III posed – A is is non invertible or poor condition number

ILL posed inverse problem

SPC

LiSens

45x
Hand-crafted prior

Original Image

Reconstructed Image (5% Measurements)

TVAL3
Learn the prior

• Using a set of dataset of natural images

• Deep generative models: Quite successful in modelling natural image distribution
Autoregressive Models

- Factorize the prior distribution as

\[ p(X) = p(x_1, x_2, \ldots, x_{n^2}) = \prod_{i=1}^{n^2} p(x_i | x_{<i}) \]

- Tractable expression for prior density
- Provides pixel-level consistencies in reconstructions
- PixelCNN++
Problem Formulation

• Given $Y$ and the forward model, find $X$

• Maximum-A-Posteriori Inference

\[
\hat{X} = \arg\max_X \log p(X|Y)
\]

\[
\hat{X} = \arg\max_X \log p(X) + \log p(Y|X)
\]

\[\text{Learned Autoregressive Model} \quad \text{Forward Model}\]
Iterative Approach

\[
\hat{X} = \arg\max_X \log p(X) + \log p(Y|X)
\]

Repeat until convergence

- **Prior Step**: Gradient ascent over Autoregressive Prior
- **Likelihood Step**: Satisfy forward model
Iterative Approach

\[ \hat{X} = \arg\max_X \log p(X) + \log p(Y|X) \]

Repeat until convergence

Prior Step

- Gradient ascent over Autoregressive Prior

Likelihood Step

- Satisfy forward model
Results

Better reconstructions in terms of metrics and pixel-level consistencies.
Example 2:

Deep Photorealistic Reconstruction of Lensless Images


*Under review*
Thin Optical System

FlatCam

Drastically reducing camera thickness by replacing lens with thin mask

S. Asif et al., *IEEE Transactions on Computational Imaging* (2016)
Ill posed inverse problem

Forward Model:

Capture

\[
Y = \Phi_L = \Phi_R = X
\]

Ill posed – \(\Phi_L\) and \(\Phi_R\) are poorly conditioned
Regularized reconstruction

\[ X = \arg \min_X \left\| \Phi_L X \Phi_R^T - Y \right\|_F^2 + \lambda \left\| X \right\|_F^2 \]
Data-driven reconstruction

Measurement → Deep learning → New

Previous
Naïve approach

Measurement → Hand-crafter prior reconstruction → Perceptual enhancement → Data-driven → Output
End-to-end approach

Based on forward model:

\[ W_1 \times Y \times W_2^T \]

Learned

Fully trainable deep network

Model inversion

Perceptual enhancement

Measurement

Output

Naïve approach
Results

Raw Captures

Tikhonov regularization

Data-driven End-to-End
Part I: Summary

Data-driven method captures good “Visual” priors leading to high-quality reconstructions
Visual Sensing using Machine Learning

Part I: Backend ML

Part II: Joint design with ML

Part III: ML with Optical System
Optical system design

Independent-crafted design

- Motivated by signal processing theory
- Reconstruction algorithm developed separately and may not compensate for drawbacks in design
- Doesn’t achieve best optimal combination

Data-driven design

- Design and reconstruction algorithm tied tightly
- Design is optimized to bring the best out of reconstruction algorithm
- Jointly produce optimal system

Joint (optical + algorithm) design lead to high-quality reconstructions

Part II: Joint design with ML
Example 1:

PhaseCam3D - Learning Phase Masks For Passive Single-view Depth Estimation

Yicheng Wu, Vivek Boominathan, Huaijin Chen, Aswin Sankaranarayanan, and Ashok Veeraraghavan.

“PhaseCam3D—Learning Phase Masks for Passive Single View Depth Estimation.”

*IEEE International Conference on Computational Photography (ICCP), 2019*
PhaseCam3D sensor

Part II: Joint design with ML

Scene → Phase-mask based camera → Optical System → Depth reconstruction algorithm → Digital network → Depth map → End-to-end neural network
Defocus of general lens

Defocused image

- Identical PSF at both sides of the focal plane.
- Impossible to tell the depth based on the blur size.
Independent-crafted designs

Veeraraghavan et al., 2007

Levin et al., 2007

Far <-> Focus <-> Near
PhaseCam3D sensor

- Scene
- Optical System
  - Lens
  - Phase mask
  - Sensor
- Digital network
- Depth map

- Differential optical model
- Digital network
- End-to-end learning
Model for the optical system

PSF formulation

\[ PSF = |F\{P\}|^2 \]
\[ P = \exp[j\phi^M_\lambda(h) + j\phi^{DF}_\lambda(z)] \]

Phase mask  Defocus

Image formulation

\[ I_{sensor}(h) = \sum_z I_{obj}(z) \otimes PSF_\lambda(h,z) + N(0, \sigma^2) \]

Noise

Differentiable model
Model for the digital network

- Pixel-wise prediction
- Skip connection

**U-Net**

- conv 3x3, ReLU, BN
- copy and concatenate
- max pool 2x2
- upsampling 2x2
- conv 1x1, sigmoid

Coded image → conv 3x3, ReLU, BN → copy and concatenate → max pool 2x2 → upsampling 2x2 → conv 1x1, sigmoid → Estimated depth
Train the network

• RGBD dataset
  – Experimental: boundary mismatch, missing depth
  – Synthetic: precise texture and depth

• Loss functions
  – Root mean square (RMS) loss
  – Gradient loss

• Mask parameters
  – Coefficients of Zernike polynomial basis

FlyingThings3D

Zernike polynomials

RGB
Disparity
PhaseCam3D: an end-to-end learning approach
Optimal simulation results

Height map

PSFs

Sharp image

Coded image

True disparity

Estimated disparity
### Comparison with independent-crafted designs

<table>
<thead>
<tr>
<th>Ground truth</th>
<th>Levin et al.</th>
<th>Veeraraghavan et al.</th>
<th>PhaseCam3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coded images</td>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
</tr>
<tr>
<td>Disparity map</td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td>Avg. RMS loss</td>
<td>0.052</td>
<td>0.054</td>
<td>0.028</td>
</tr>
</tbody>
</table>

**Legend:**
- Chart depicts comparison of coded images and disparity maps for different methods.
- Levin et al. and Veeraraghavan et al. represent independent-crafted designs.
- PhaseCam3D is the proposed method.
- The chart includes a color bar for visual comparison, ranging from 0 to 1.
Fabricate the learned phase mask

Two-photon lithography 3D printer

Fabricated phase mask

Photonic Professional GT, Nanoscribe GmbH

Fabricated phase mask 2.835 mm
Experimental results

Indoor scenes

Outdoor scenes

Coded images

Disparity map
Experimental results

Indoor scenes

Outdoor scenes

Coded images

Depth map

meters

meters
Accuracy evaluation: compare with Kinect

Estimated depth by PhaseCam3D

Estimated depth by Kinect

Coded Images

Error:

$\sigma_{\text{PhaseCam3D}} = 1.25\text{cm}$
Example 2:

End-to-end Optimization of Optics and Image Processing for Achromatic EDOF


“End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging.”

Joint design of optics and Image reconstruction

Optical system:
Parameterized Lensing Element

\[ \star \]

Image Dataset

Optics Model

Sensor Model

Computational Image Reconstruction

Domain-Specific Loss

\[
\begin{align*}
S(I) & \xrightarrow{\text{Convolve with PSF } p_{\lambda}} y \\
\frac{\partial S(I)}{\partial y} & \xrightarrow{\text{Noise } \eta} \min_x ||y - G(x)||_2^2 + y ||x||_2^2 \\
\frac{\partial c}{\partial x} & \xrightarrow{\text{Loss } L}
\end{align*}
\]

Sitzmann et al., TOG, 2018
Results – Achromatic EDOF

- Achromatic EDOF

- Depth: 1.0m
  - Fresnel Lens
  - Cubic Phase Lens
  - Multi-focal Lens
  - Diffractive Achromat
  - End-to-end Optimized

- Depth: 0.5m
  - Fresnel Lens
  - Cubic Phase Lens
  - Multi-focal Lens
  - Diffractive Achromat
  - End-to-end Optimized

Sitzmann et al., TOG, 2018

Dowskey & Cathey, 95

Peng et al., 2016
Example 3:

Learning Sensor Multiplexing Design through Back-propagation

Ayan Chakrabarti.

“Learning sensor multiplexing design through back-propagation.”

Joint design of color multiplexing and demosaicking

Optical system:
Sensor color filter array

Optical Measurement

Sensor CFA Pattern (PxP Repeated)
Measures one of C channels per pixel

Computational Reconstruction

Reconstruction Network

Loss Gradients

Input as intensities in C possible channels

Full Color RGB Image

Ayan Chakrabarti, NeurIPS, 2016
Results – Color demultiplexing

Ground Truth

Independent-crafted

Bayer

CFZ

Learned

ICCP 14

Ayan Chakrabarti, NeurIPS, 2016
Part II: Summary

Joint design with data-driven techniques can bring out best in both systems.

Part II: Joint design with ML

Data-driven

Optical System

Machine Learning

Output

Improve Quality

Joint design with data-driven techniques can bring out best in both systems.
Visual Sensing using Machine Learning

Part I: Backend ML

Part II: Joint design with ML

Part III: ML with Optical System

Machine Learning System

Optical Layer(s) → Electronic Layer(s)

Machine Learning System

Optical Layer(s) → Electronic Layer(s)

Machine Learning System

Optical Layer(s) → Electronic Layer(s)
ML Optical layer

Traditional Vision
- Camera captures sharp image
- ML algorithm extracts features to perform inference
- All computation is electronic, consuming energy

ML with optical layer
- Optical layer directly extracts features
- Electronic layer uses the features for inference
- Reduced power consumption

Optical layer can extract features directly that can be used for inference
Example 1:

ASP Vision: Optically Computing the First Layer of CNNs using Angle Sensitive Pixels

ASP camera as first layer of DNN

ASP Vision: Sensor + Deep Learning Co-Design

Scene → Optically computed → Reduced CNN

ASP camera has gabor-like filters that show up as kernels in many CNNs, e.g., AlexNet

“Elephant”
Angle Sensitive Pixels (ASPs)

- ASPs are CMOS image sensors that have two diffraction gratings over photodiodes.

- Different oriented/spaced gratings produce different oriented/frequency gabor-like point-spread-functions.

[Image of ASPs and diffraction gratings]

[Wang and Molnar, JSSC 2012]
[M. Hirsch et al., ICCP 2014]
ASP sensors save energy with edge-only digitization

- 90% savings in image sensing
- 90% savings in bandwidth

<table>
<thead>
<tr>
<th></th>
<th>Sony (ISSCC 2015)</th>
<th>ASP Image Sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>5256 x 3934 (20M)</td>
<td>384 x 384 (effective ASP tile resolution: 96 x 64)</td>
</tr>
<tr>
<td>Energy consumption</td>
<td>Total power: 428 mW No breakdown of power reported</td>
<td>Total Power: 1.8 mW Pixel Array: 300 μW Amplifiers: 900 μW Timing/Addressing: 500 μW ADCs: 100 μW</td>
</tr>
<tr>
<td></td>
<td>90.2% Energy saving</td>
<td></td>
</tr>
<tr>
<td>Transmission bandwidth</td>
<td>Transmitting the entire image 1.2 Mbits/frame @ 384x384x8bits</td>
<td>Transmitting only edges 120 Kbits/frame @ 384x384x8bits</td>
</tr>
<tr>
<td></td>
<td>10:1 Compression ratio</td>
<td></td>
</tr>
<tr>
<td>Capabilities</td>
<td>2D image and video capture</td>
<td>2D images and video, edge filtered images, light field information</td>
</tr>
</tbody>
</table>

[A. Wang, S. Sivaramakrishnan, and A. Molnar, CICC 2012]
Results - ASP Vision performs comparably to CNNs on datasets

Performance on Visual Recognition Tasks

- **PF-83 - VGG-M** (Face Identification):
  - ASP Vision: 66.80%
  - Baseline - 12 filters: 69.78%
  - Baseline - Original # of filters: 65.67%

- **CIFAR-100 - NiN** (Object Classification):
  - ASP Vision: 50.90%
  - Baseline - 12 filters: 55.60%
  - Baseline - Original # of filters: 57.50%

- **CIFAR-10 - NiN** (Object Classification):
  - ASP Vision: 81.80%
  - Baseline - 12 filters: 84.90%
  - Baseline - Original # of filters: 86.40%

- **MNIST - LeNet** (Digit Recognition):
  - ASP Vision: 99.04%
  - Baseline - 12 filters: 99.14%
  - Baseline - Original # of filters: 99.12%
How many FLOPs can we save by skipping the first layer?

<table>
<thead>
<tr>
<th></th>
<th>VGG-M</th>
<th>NiN</th>
<th>LeNet</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Conv. Layers</td>
<td>8</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>Input Image Size</td>
<td>$224 \times 224 \times 3$</td>
<td>$32 \times 32 \times 3$</td>
<td>$28 \times 28 \times 1$</td>
</tr>
<tr>
<td># of First Layer Filters</td>
<td>96 (Original)</td>
<td>12 (Prototype)</td>
<td>192 (Original)</td>
</tr>
<tr>
<td>First Layer Conv. Kernel</td>
<td>$7 \times 7 \times 96$</td>
<td>$7 \times 7 \times 12$</td>
<td>$5 \times 5 \times 192$</td>
</tr>
<tr>
<td>FLOPS of First layer</td>
<td>708.0M</td>
<td>88.5 M</td>
<td>14.75M</td>
</tr>
<tr>
<td>Total FLOPS</td>
<td>6.02G</td>
<td>3.83 G</td>
<td>200.3M</td>
</tr>
<tr>
<td>First Layer FLOPS Saving</td>
<td><strong>11.76%</strong></td>
<td><strong>2.3%</strong></td>
<td><strong>7.36%</strong></td>
</tr>
</tbody>
</table>
Example 2:

**Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification**

Julie Chang, Vincent Sitzmann, Xiong Dun, Wolfgang Heidrich, and Gordon Wetzstein.

“Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification.”

Hybrid optical-electronic CNN

Phase mask in Fourier plane to generate **learned** convolution filters

Chang et al., Scientific Reports, 2018
4F Optical system

(a) 4f system

- Input plane (DMD)
- Lens
- Fourier plane (phase mask)
- Lens
- Output plane (camera sensor)

(b) Phase mask fabrication
- Optimized mask template
- Mask template:
  - Zoom 1
  - 4x objective
- Fabricated mask

(c) Captured PSF and sensor image
- Captured PSF
- Captured input image
- Captured sensor image
## Results – CIFAR 10 classification

<table>
<thead>
<tr>
<th>Method</th>
<th>Test accuracy mean ± std. dev.</th>
<th>Learned parameters optical/electronic</th>
<th>FLOPs optical/electronic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Simulation:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fully connected (FC) only</td>
<td>29.8 ± 0.5%</td>
<td>−/10,250</td>
<td>−/20,480</td>
</tr>
<tr>
<td>digital conv &gt; ReLU &gt; FC, unconstrained</td>
<td>51.9 ± 1.3%</td>
<td>−/82,586</td>
<td>−/1,490,954</td>
</tr>
<tr>
<td>digital conv &gt; ReLU &gt; FC, nonnegative</td>
<td>36.3 ± 0.5%</td>
<td>−/82,586</td>
<td>−/1,490,954</td>
</tr>
<tr>
<td>digital conv &gt; ReLU &gt; FC, pseudonegative</td>
<td>51.8 ± 0.6%</td>
<td>−/83,234</td>
<td>−/2,818,058</td>
</tr>
<tr>
<td>optical conv &gt; ReLU &gt; FC, pseudonegative</td>
<td>51.0 ± 1.4%</td>
<td>104,976/81,938</td>
<td>3,779,136/180,234</td>
</tr>
<tr>
<td><strong>Physical experiment:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>optical conv &gt; ReLU &gt; FC, pseudonegative</td>
<td>44.4%</td>
<td>−/81,938</td>
<td>−/180,234</td>
</tr>
</tbody>
</table>

2x classification accuracy for same power

Only 12% of FLOPs for same accuracy

Chang et. al., Scientific Reports, 2018
Example 3:

**All-optical machine learning using diffractive deep neural networks**

Xing Lin, Yair Rivenson, Nezih T. Yardimci, Muhammed Veli, Yi Luo, Mona Jarrahi, and Aydogan Ozcan.

“All-optical machine learning using diffractive deep neural networks.”

*Science* (2018)
All Diffractive Optical System

Diffractive layers interacting with input coherent field

Lin et. al., Science, 2018
Learned diffractive masks

Phase mask profiles

Lin et. al., Science, 2018
Results – Classification accuracy

**MNIST classification**

**Fashion-MNIST classification**

Lin et al., Science, 2018
Incorporating optical layer(s) into machine learning system can decrease latency and power.
Summary: Visual Sensing using Machine Learning

(I) Backend ML

Optical System → Machine Learning

(II) Joint design with ML

Optical System ↔ Machine Learning

(III) ML with Optical System

Machine Learning System

Optical Layer(s) → Electronic Layer(s)