



# Inverse-Designed Subwavelength RGB Color Routers

Student: Luke Qi

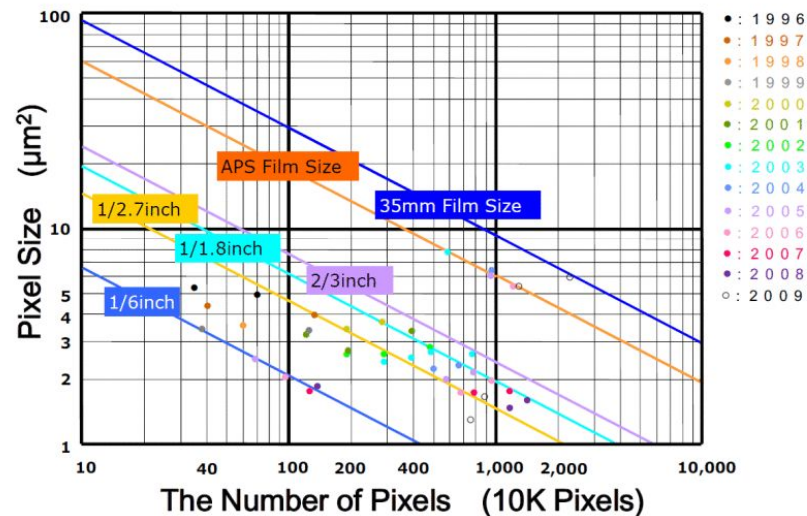
Mentor: Peter Catrysse

Course Instructors: Brian Wandell, Josh Wilson (TA from 2022)

# CMOS image sensors are rapidly developing

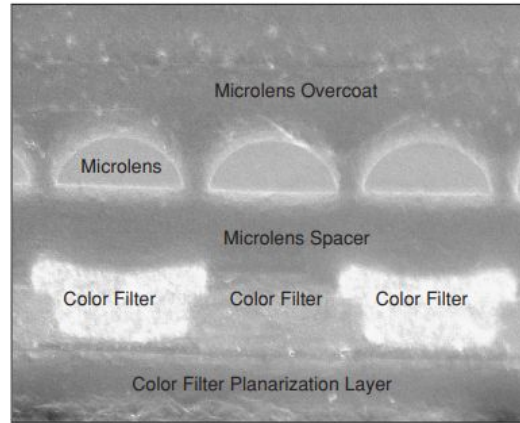
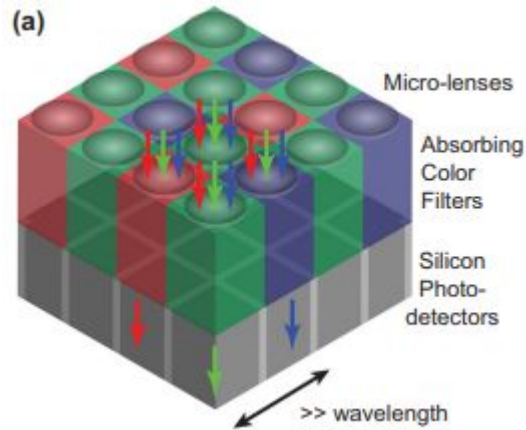


- low power and cost
- scalable fabrication
- integrated with signal processing electronics
- miniaturization
- high speed imaging

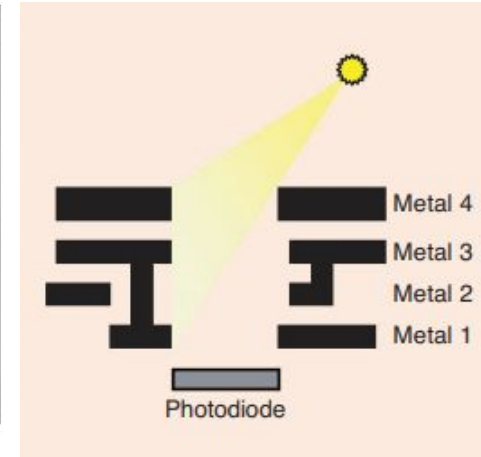


- starting to reach submicron pixel size

# How state-of-the-art CMOS pixels operate



3. A cross-section SEM photograph of an image sensor showing the microlens and CFA deposited on top of the photodetectors.



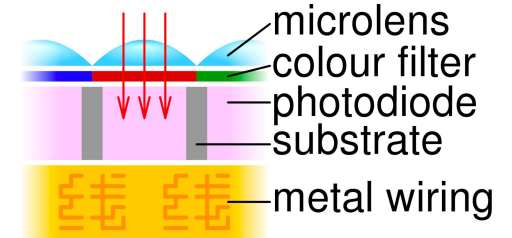
- RGGB Bayer color filter array absorbs/scatters most of the incoming light
- External quantum efficiency (QE) = Internal QE x Optical Efficiency (OE)
  - OE has a large effect on external QE
- OE scales unfavorably as CMOS technology scales
  - Optical crosstalk also increases as pixel size decreases

# Back-Illumination CMOS

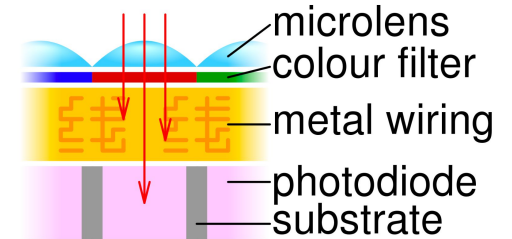
NB. back-illumination has become very popular

- places the interconnects behind the photodiode
- improves OE by 50%
  
- great in low light conditions, where photon collection is very crucial
- great for smaller pixels too
  
- still doesn't solve the problem of microlens as pixel sizes become submicron

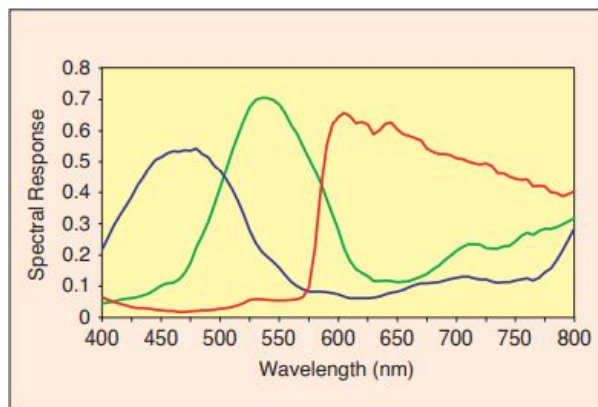
## 1. Back-side illumination



## 2. Front-side illumination



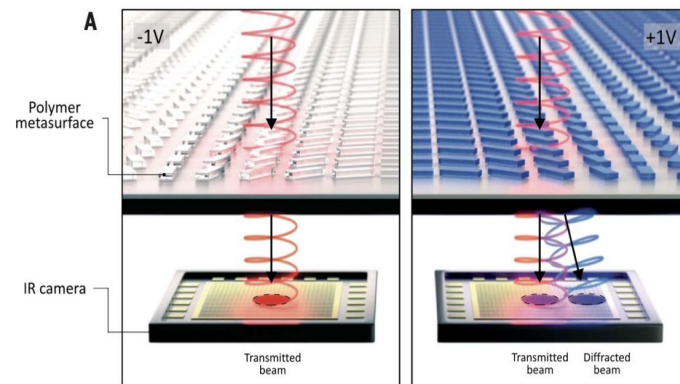
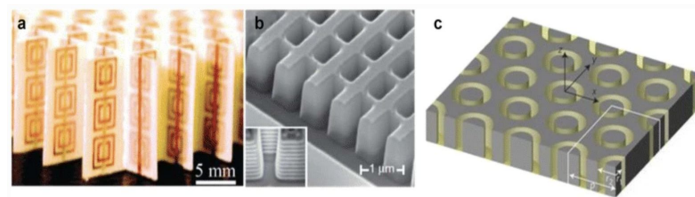
# What truly limits pixel size?



6. A spectral response curve for a typical 0.18- $\mu\text{m}$  CMOS image sensor.

- Wavelength of light  $\sim 450$  nm
- Photodiodes scale well below  $1 \mu\text{m}$ 
  - $\text{OE} \propto L^2$
- CFA kernel does not scale well below  $1 \mu\text{m}$ 
  - state-of-the-art  $\sim 20\text{-}25\%$  OE

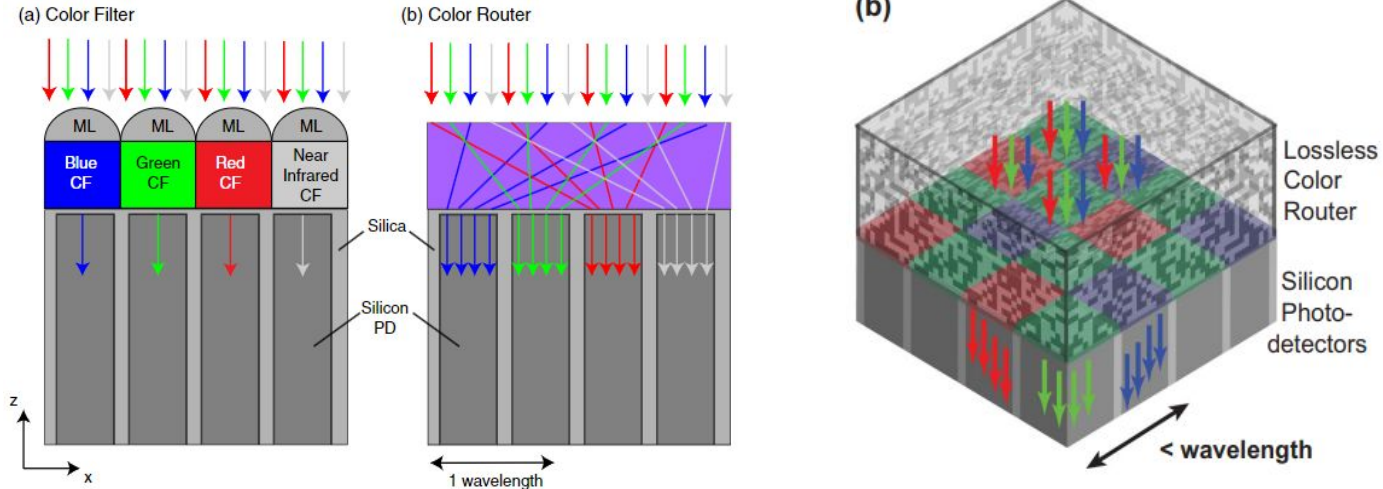
metalenz



- metasurfaces allows for subwavelength control over light

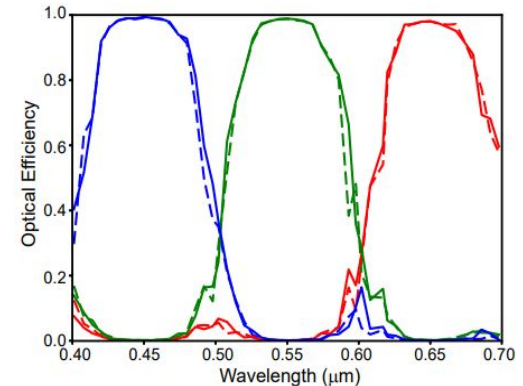
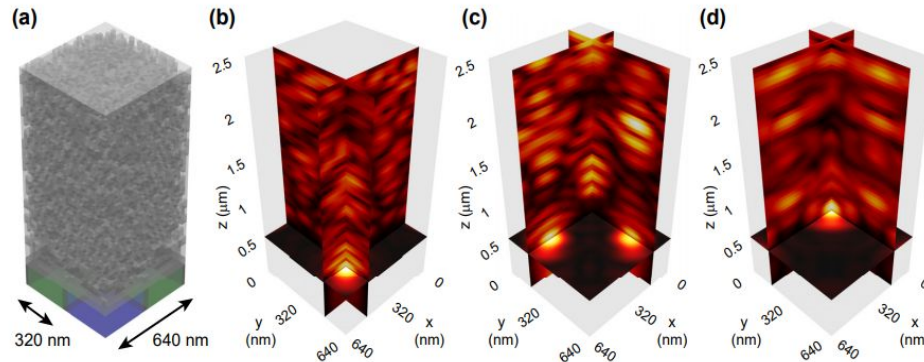
# Color Router: a promising novel device

A color router is a device that *routes all incident light based on its color content, i.e., different parts of the visible spectrum, directly (without additional propagation) and without loss to the silicon photodetector of the intended color channel.*



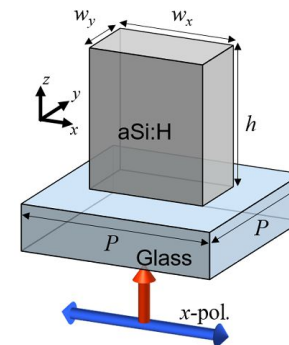
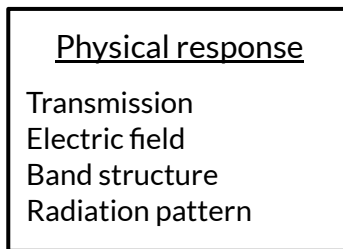
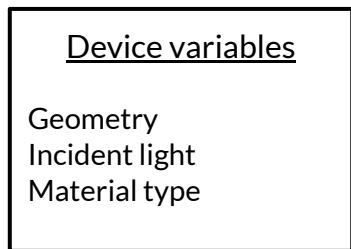
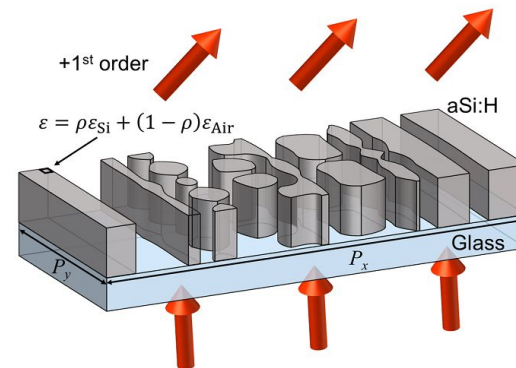
# Near unity efficiency is possible in principle

- performance is not limited by physics
- replaces all other parts of the stack:
  - microlens, color filter, anti-reflection coating
- TiO<sub>2</sub> ( $\epsilon = 7$ ), SiO<sub>2</sub> ( $\epsilon = 2.1$ )
- OE remains high (>98%) for different polarizations and  $\pm 10$ -16 deg angular extent
  - improves OE by 2.5-5x



# Can we fabricate some prototypes with current tools?

- Lithography can pattern feature sizes down to  $\sim 10$  nm
- Requires fabrication friendly designs
- Utilize **inverse-designed photonics**
- Open-sourced torcwa package
  - forward simulator: RCWA
  - backwards gradients: adjoint variable method
  - ADAM optimizer



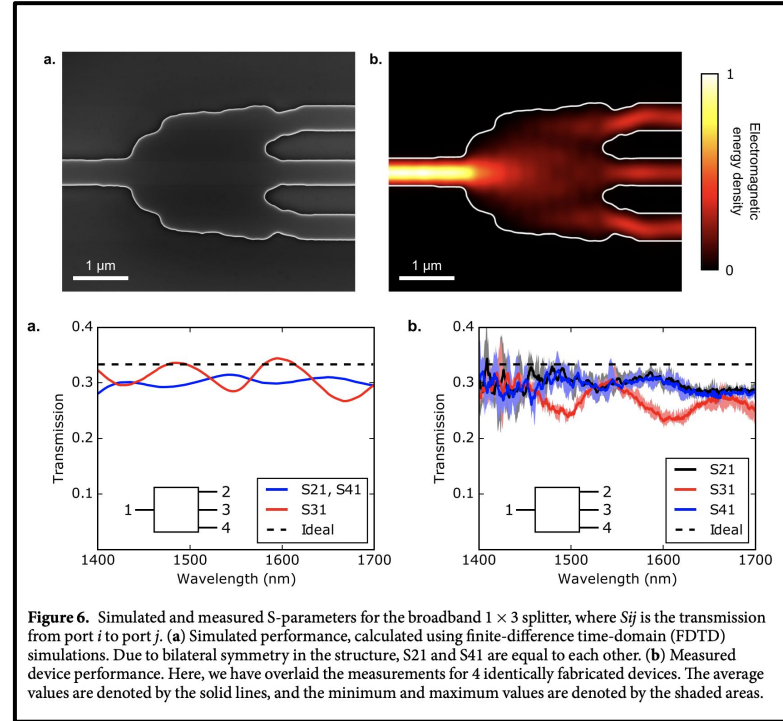
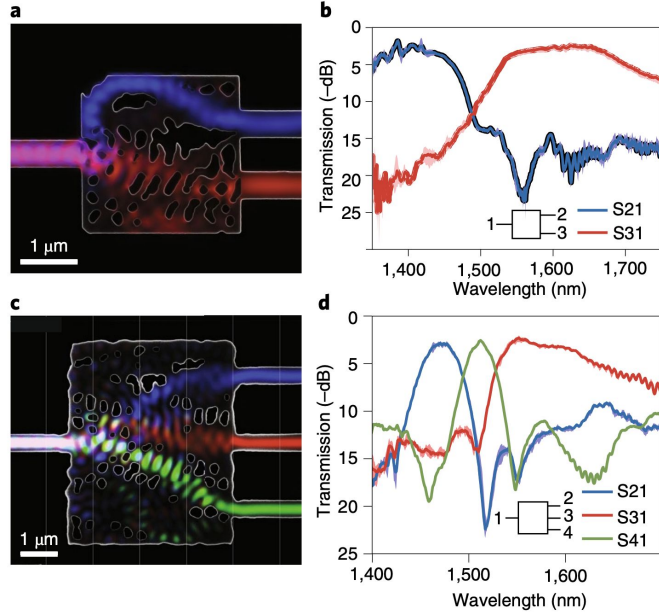
# Inverse Design is experimentally robust.

36 hrs, a single server,  
three NVidia GTX Titan  
graphics cards.

Electron-beam  
lithography

60 hrs, a single  
computer, Intel Core  
i7-5820K processor,  
64GB RAM, three  
Nvidia Titan Z graphics  
cards.

Electron-beam  
lithography



**Figure 6.** Simulated and measured S-parameters for the broadband  $1 \times 3$  splitter, where  $S_{ij}$  is the transmission from port  $i$  to port  $j$ . (a) Simulated performance, calculated using finite-difference time-domain (FDTD) simulations. Due to bilateral symmetry in the structure, S21 and S41 are equal to each other. (b) Measured device performance. Here, we have overlaid the measurements for 4 identically fabricated devices. The average values are denoted by the solid lines, and the minimum and maximum values are denoted by the shaded areas.

# Initial Results with torcwa

The screenshot displays the GitHub interface for the repository `kch3782/torcwa`. The main content area shows a file tree with the following items:

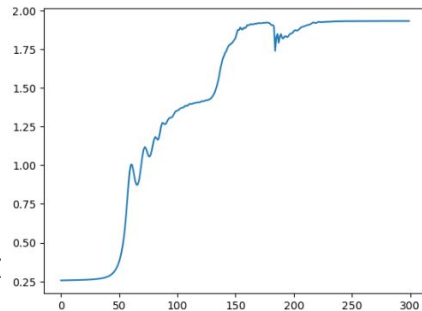
File/Folder	Description	Last Update
docs	User guide upload	last year
example	version 0.1.4 update	last year
image	Initial commit	last year
torcwa	version 0.1.4.2 update	9 months ago
.gitattributes	Initial commit	last year
.gitignore	Initial commit	last year
LICENSE	S-parameter normalization corrected	last year
README.md	version 0.1.4.2 update	9 months ago
requirements.txt	Initial commit	last year
setup.py	version 0.1.4.2 update	9 months ago

The right sidebar provides additional context:

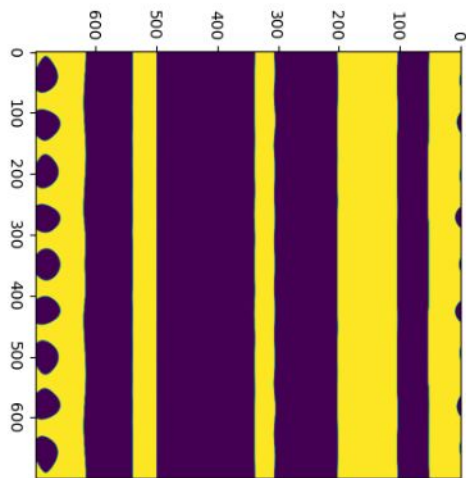
- About:** GPU-accelerated RCWA with automatic differentiation. Tags: `pytorch`, `rigorous-coupled-wave`, `electromagnetic-simulation`, `optics-simulation`.
- Readme:** View license
- Activity:** 58 stars, 5 watching, 11 forks
- Releases:** Report repository

# Diffraction of 450nm, 550nm, 650nm

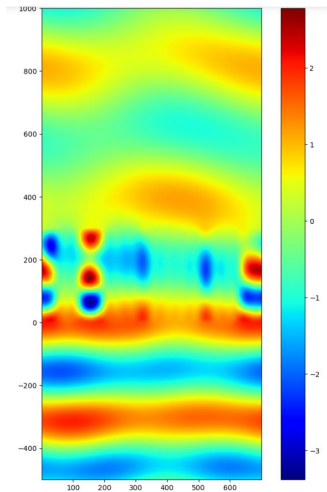
- 15s per FoM calculation (4500s total)
  - 450nm at 0,0 diffraction, 550nm at 1,0, 650nm at -1,0 (Example6\_data\_450nm,550nm,650nm\_)
- 700 nm by 700 nm device



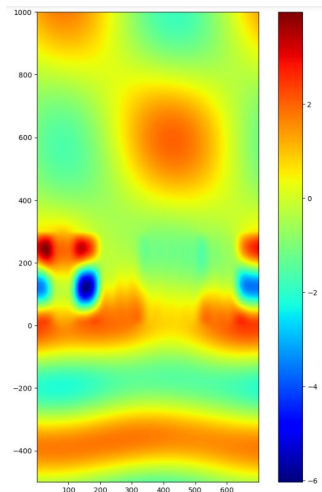
Yellow corresponds to aSiH  
Blue corresponds to air



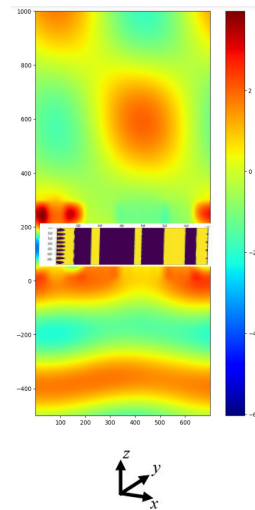
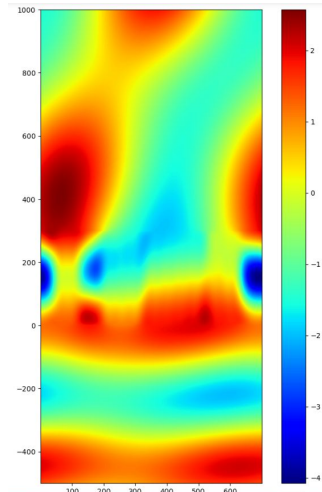
450nm XZ slice



550nm XZ slice



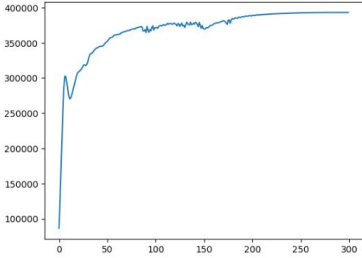
650nm XZ slice



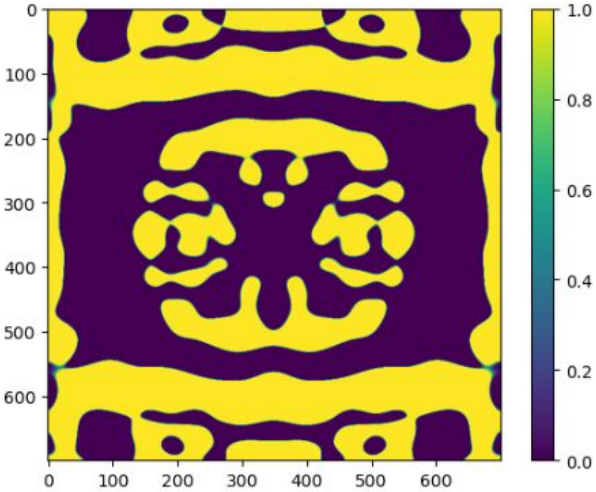
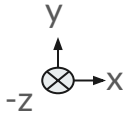
# Flux (Poynting vector) above the structure



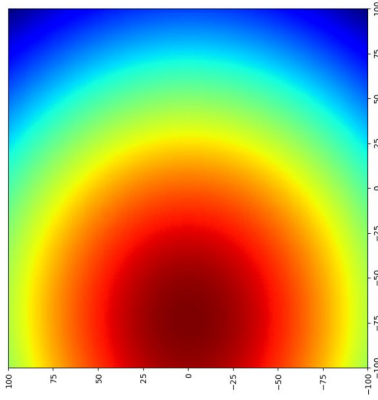
- Optimized for 100nm by 100nm XY box, 100nm above the surface
  - all three wavelengths (450nm, 550nm, 650nm)
  - 18s per FoM calculation (5630s total)



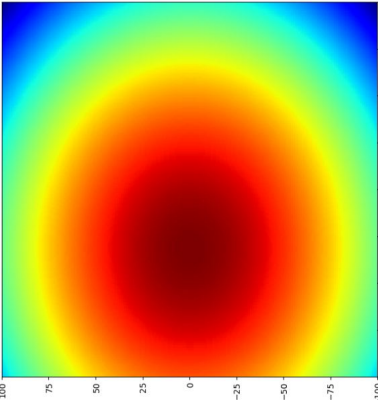
Yellow corresponds to aSiH  
Blue corresponds to air



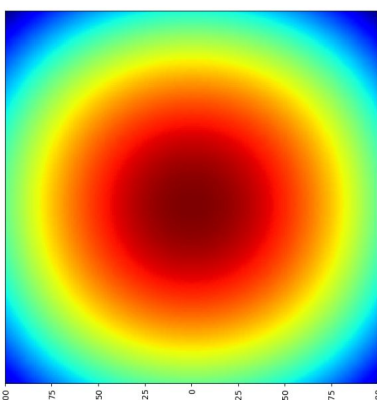
450nm XY slice



550nm XY slice

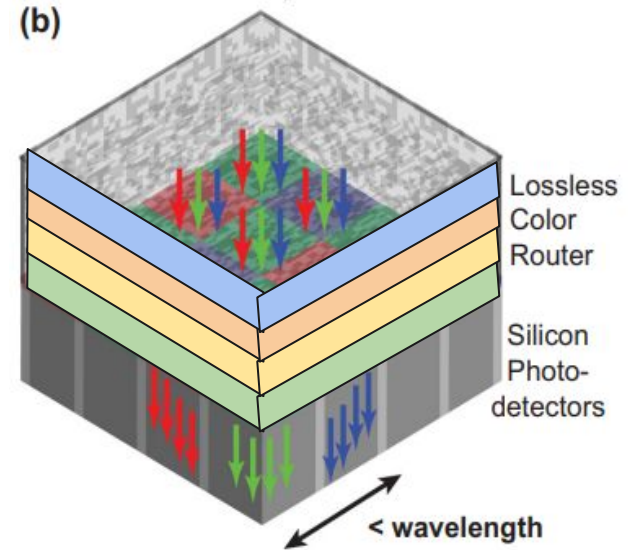


650nm XY slice



## Next Steps

- improve optimization cycle
  - utilize CUDA to accelerate forward solving
  - try other optimization algorithms
- add other materials besides aSiH
  - SiO<sub>2</sub> and TiO<sub>2</sub> is popular
- add multiple layers
- **try to fabricate some designs!**



---

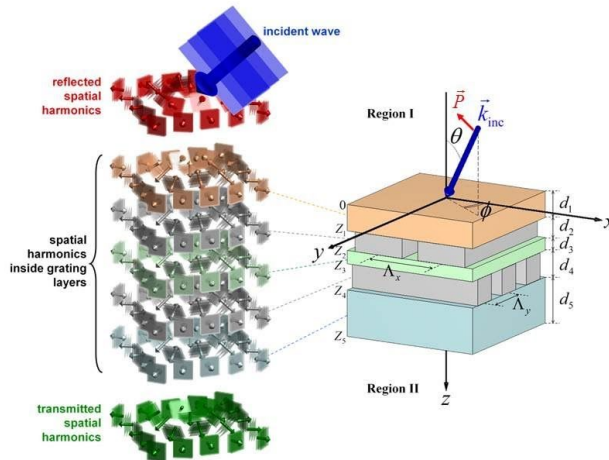
# Additional slides

# RCWA

- commonly used for periodic dielectric surfaces
- devices and fields represented in Fourier space
- solve for transfer matrices between layers

## Geometry of RCWA

CEM



- `torcwa.rcwa_geo` class defines the geometry and makes a grid
- `torcwa.rcwa` solves S matrix using Rigorous Coupled Wave Analysis
- objective function takes in a pattern and returns a scalar
  - torch scalar value, which has a backwards function to calculate gradients
- initialize hyper parameters for optimization, use PyTorch
  - needs a blur kernel for fabrication feasibility
  - uses ADAM optimizer

# Adjoint State Method

- efficient method for calculating gradients of functions
- used commonly in EM and also neural networks (backpropagation)

**Adjoint Simulation**

Consider solving the first two terms:

$$\frac{dJ}{dp} = 2\mathcal{R} \left\{ -\frac{\partial J}{\partial e} A^{-1} \frac{dA}{dp} e \right\}$$

↓

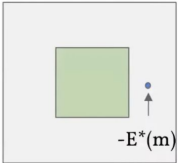
$$e_{adj}^T = -\frac{\partial J}{\partial e} A^{-1}$$

↓

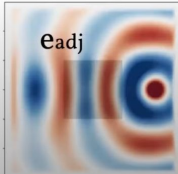
Transpose, multiply by A.      Note: A is symmetric

$$A e_{adj} = -\frac{\partial J}{\partial e} \quad \text{"adjoint source"}$$

We call this the "adjoint" simulation the source at m with amplitude  $E^*(m)$



Solve the same way as original.



**Tidy3D**

# Single Layer Color Routing



- 100nm by 100nm XY box, 100nm above the surface
  - all three wavelengths (450nm, 550nm, 650nm)
- 17s per FoM calculation
  - 5079s total

