Image Systems Simulation

Prof. Brian Wandell
Stein Family Professor

Stanford's Center for Cognitive and Neurobiological Imaging
Founding Director

Stanford Neurosciences Institute
Deputy Director

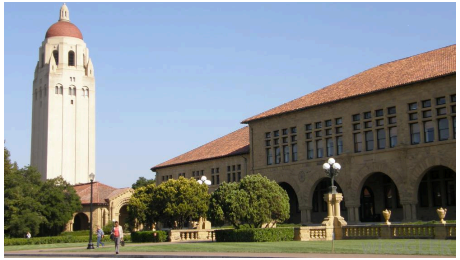
Dr. Joyce Farrell
Electrical Engineering

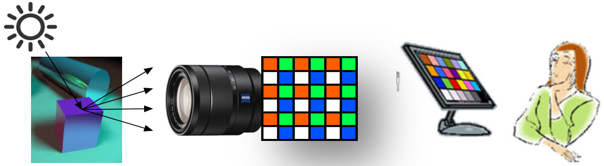
Stanford's Center for Image Systems Engineering
Executive Director

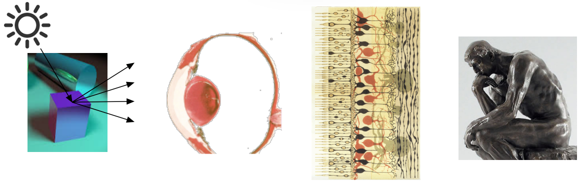
Max-Planck Center for Visual Computing and Communications
Executive Director

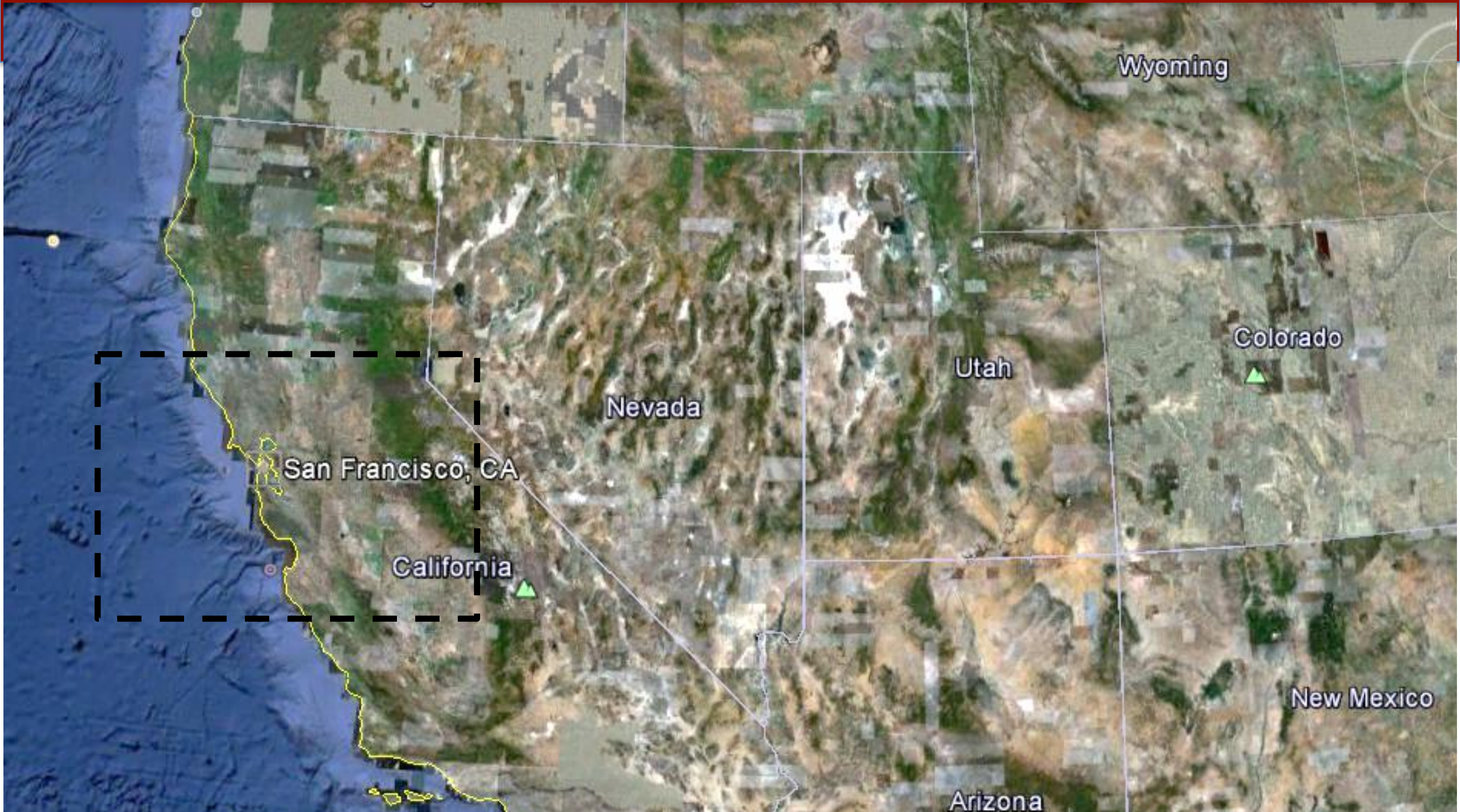
Outline

- About Stanford
- Image systems simulation
- ISS Examples
  - ISET – consumer photography
  - ISETBIO – visual neuroscience
  - CISET – computational imaging
  - RenderToolbox3, PBRT, docker, git
- Discussion about vehicles and new imaging methods
  - Camera arrays
  - RGB-D sensors
  - Light field sensors









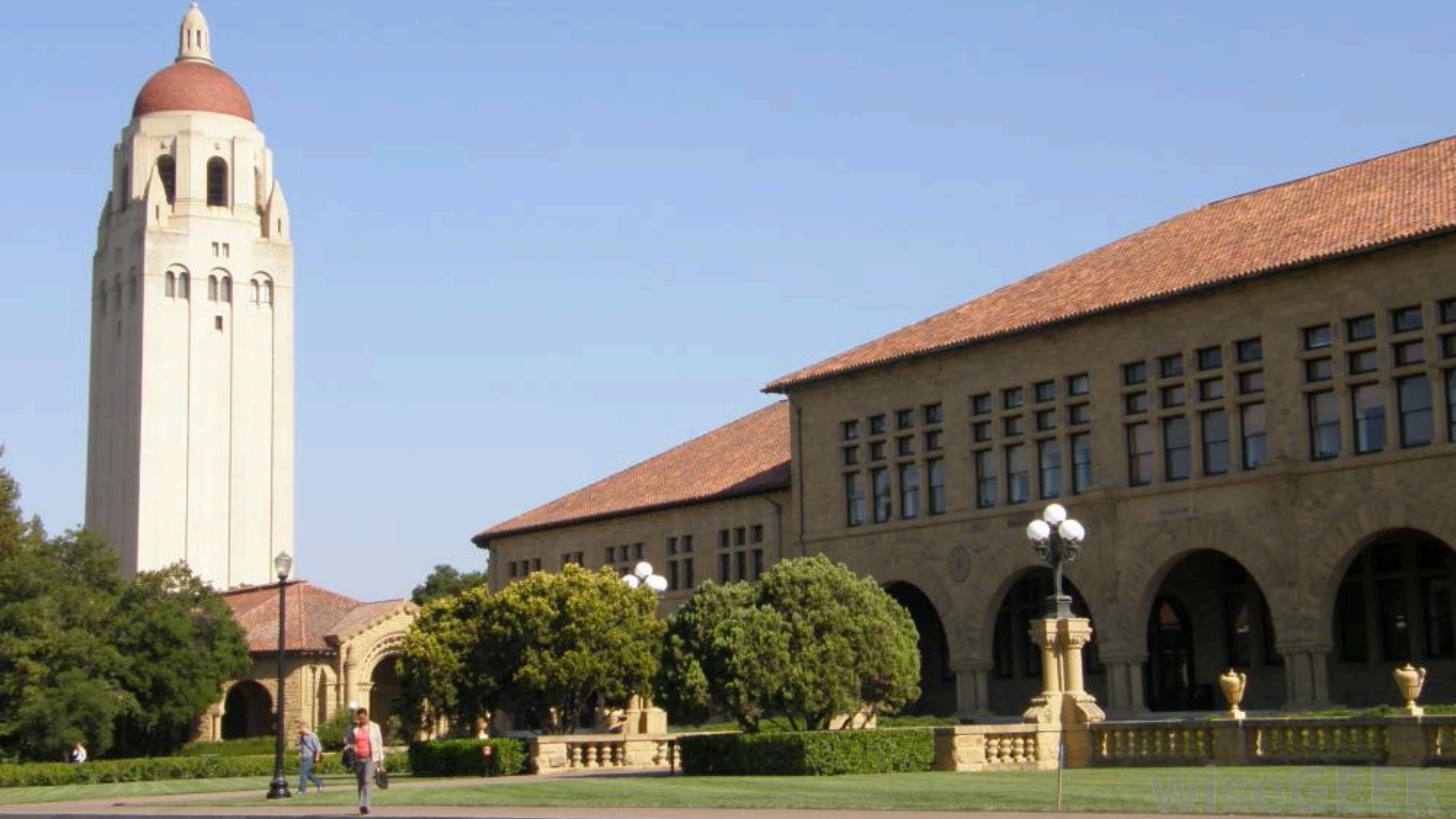
Wyoming
Colorado
Utah
Nevada
San Francisco, CA
California
New Mexico
Arizona


San Francisco


UC Berkeley


Stanford University
San Mateo
Mountain View
San Jose
Santa Clara
© 2011 Google


6/2011
© 2011 Google
© 2011 Europa Technologies
194:01
Imagery Date: 6/19/2011
1948
37°25'47.51" N 122°10'16.03" W elev 81 ft
Eye alt 956 ft
Google earth

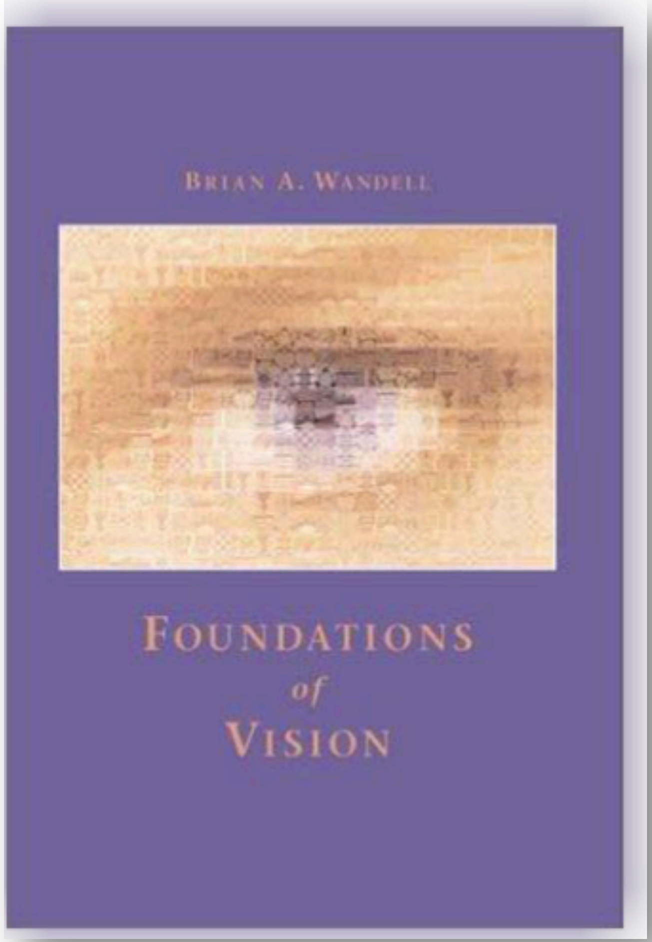

wiseGEEK

My research training began in vision science

In 1995 I wrote a textbook on vision science; it is available now through my web-page

Dr. Farrell and I have worked on the design and evaluation of digital imaging systems at Stanford Center for Image Systems Engineering; we co-teach a course

Much of my work uses magnetic resonance imaging to measure the structure and function of human visual cortex; I will speak on this topic in Beijing next week

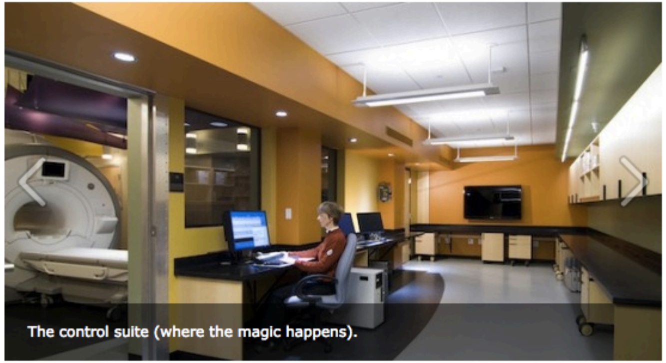


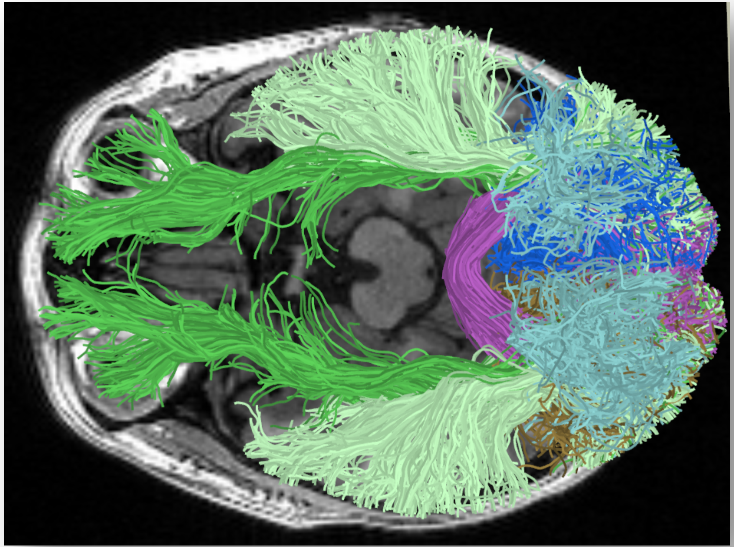
Human Neuroscience

A substantial portion of my research uses magnetic resonance imaging (MRI) to measure the structure and function of human visual cortex. We invented methods for mapping visual cortex. I teach a course on MRI and direct a center.

Welcome to the CNI

The Stanford Center for Cognitive and Neurobiological Imaging (CNI) is a shared facility, dedicated to research and teaching. The Center provides resources for researchers and students in cognitive and neurobiological sciences.

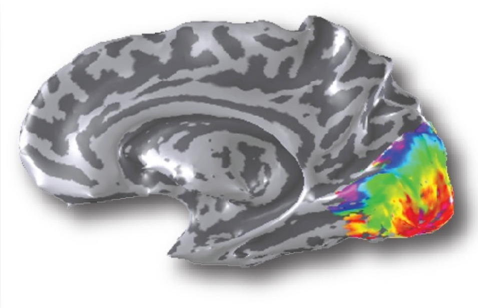


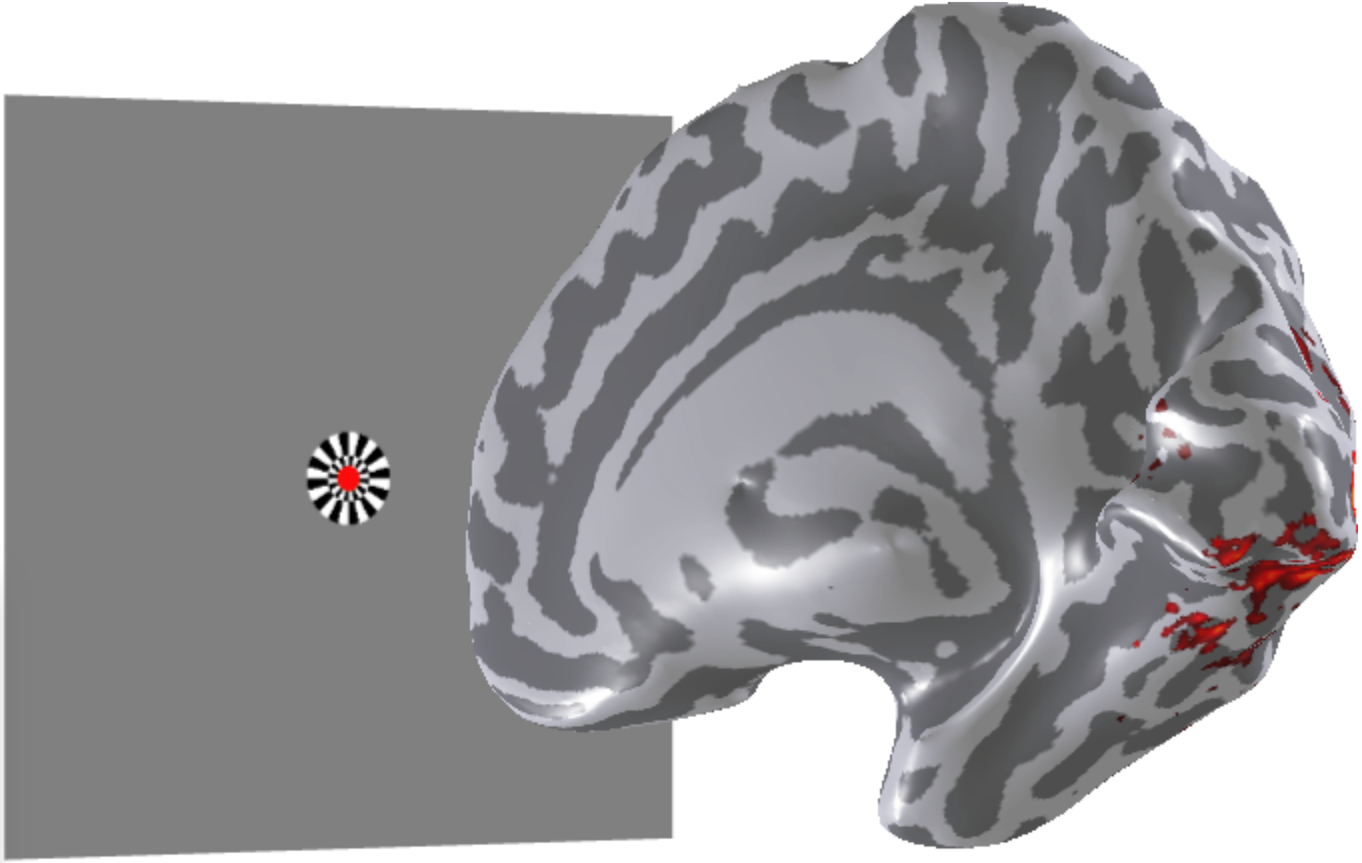


Measuring human visual cortex

(Engel et al., 1994,1997; Sereno; Tootell, DeYoe; Others)

- Inflated brain
- Gray/white are sulci/gyri





Project on Scientific Transparency

- Reproducible Research
- Personalized neuroscience
- Software for sharing data and analyses

Project on Scientific Transparency

- Reproducible Research
- Personalized neuroscience
- Software for sharing data and analyses

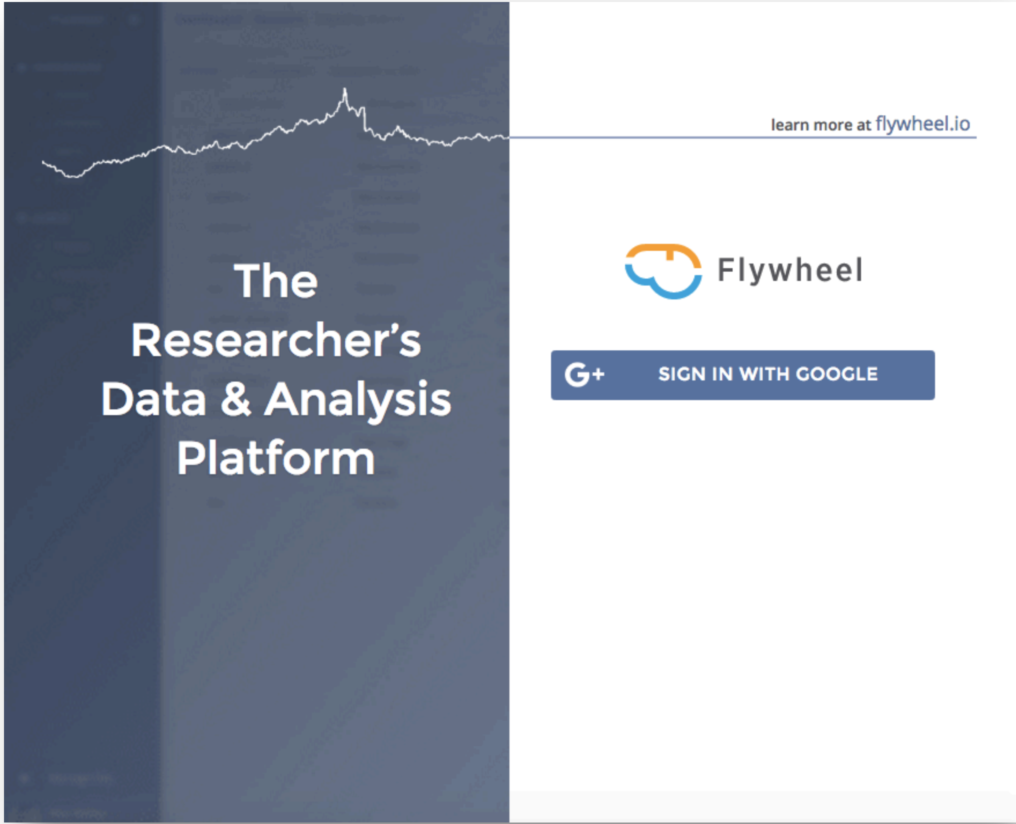


Image systems engineering

Dr. Farrell and I teach a course on Image Systems Engineering for Electrical Engineering students. Research and applications in this field are the basis of our planned collaboration.

In this presentation I introduce our work on Image systems simulation.

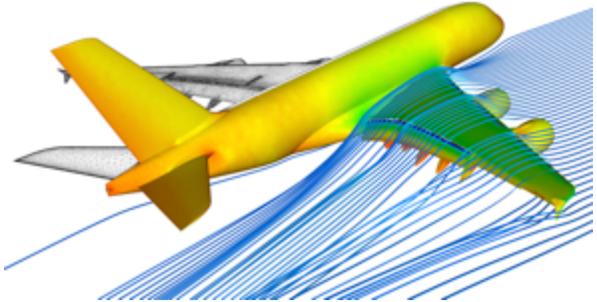
I hope that during this visit we will discuss research opportunities that combine our methods with your skills in

ISET is the Image Systems Engineering Toolbox

Simulation technology for advanced system design

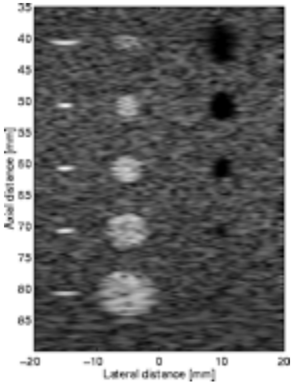
- Simulation is widespread in several advanced industries
- We are working to implement simulations with physical units and the general methodology in image systems (cameras, displays)

Numerical flow simulation on an Airbus A380



ECU (Electronic Control Unit) Simulation for Automobiles

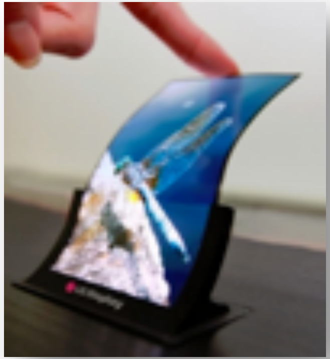
Field II Ultrasound Simulation



Value of image systems simulations

- Methodically explore design trade-offs.
- Allows flexibility in prototyping.
- Experimental isolation of variables that are difficult to control.

Displays



Cameras

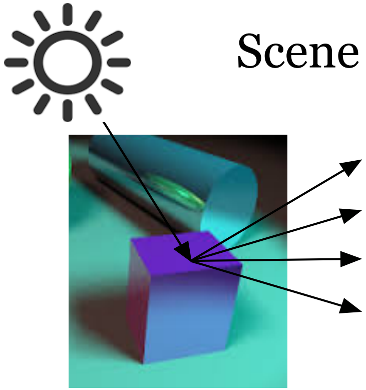
Camera arrays

Image Systems Simulation

The Image Systems Engineering Toolbox (ISET)

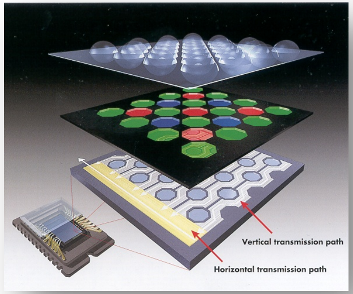
MATLAB
The Language of Technical Computing

Scene



Optics

Sensor

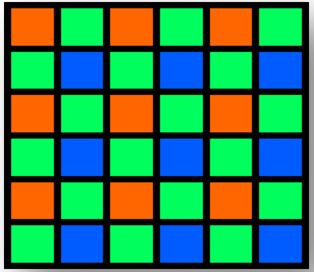


Processing

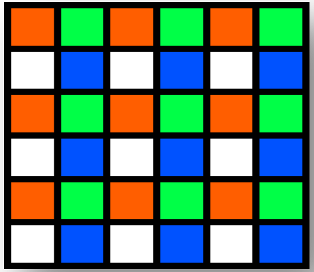
Image systems simulation: An example project (L3)

- High density (small size) and excellent electrical properties of modern pixels enable new sensors for new applications

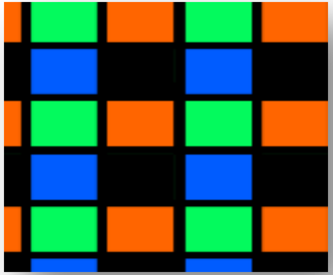
- **Challenge**: Design image processing pipelines for consumer photography that exploit the spatial-spectral statistics of the scenes and properly render data for these new types of sensors



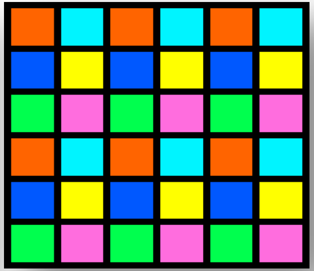
Bayer



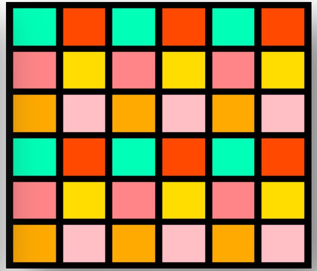
RGBW
low-light sensitivity
dynamic range



RGBX
infrared
light field



RGBCMY
multispectral



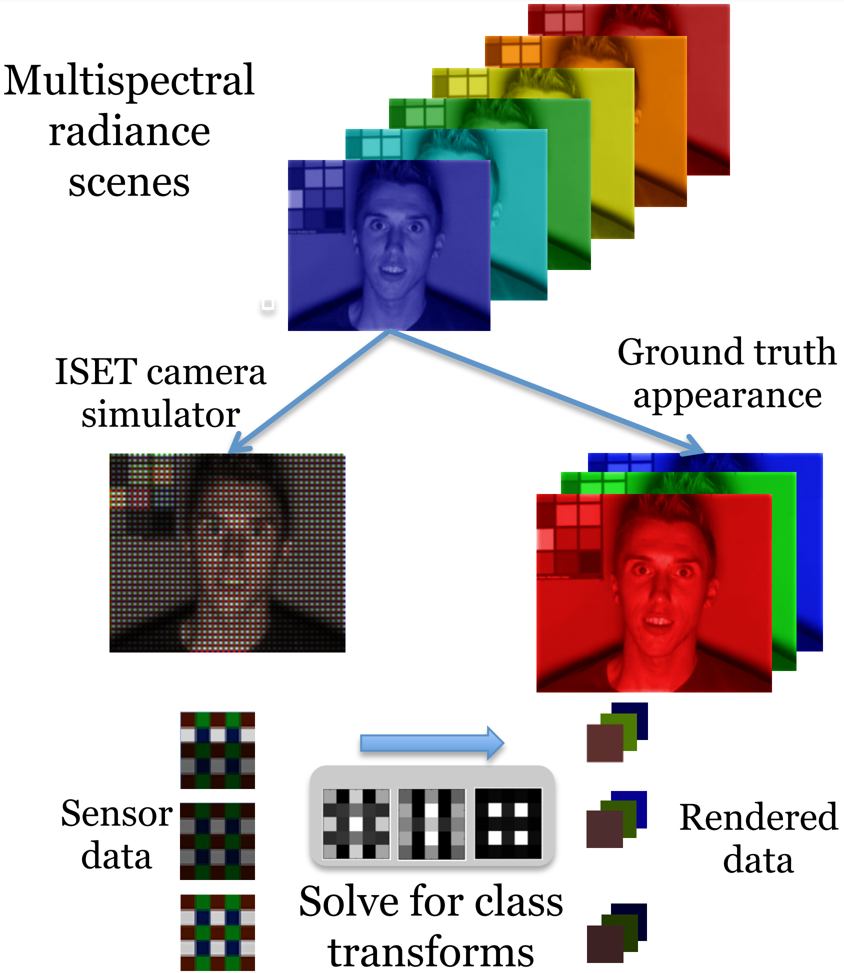
Medical
specialized application

Introduction to Image Systems Simulation:

- The multispectral input is known; in this example we use the calibrated input as the target output (reproduction)

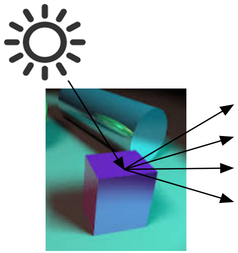
- We build a model of the camera (optics, sensor) and the image systems simulation software (ISET) produces raw sensor data

- Using many input images, we find the transformations from the sensor data classes to the desired output



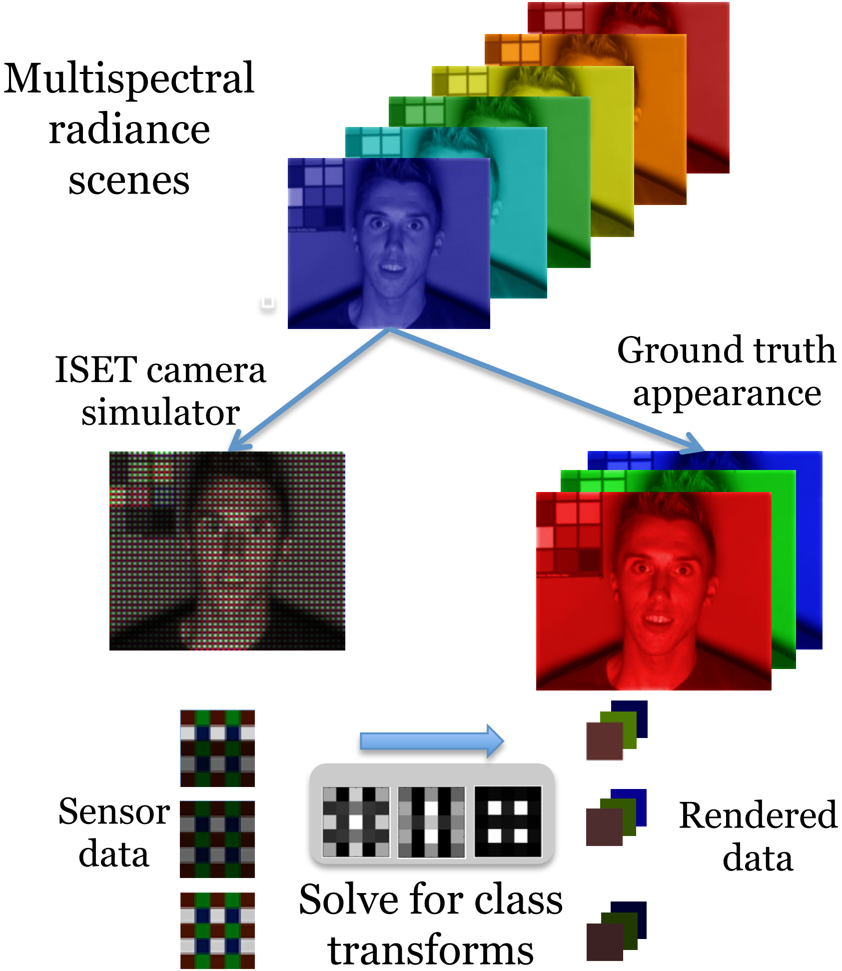
Introduction to Image Systems Simulation:

- The multispectral input is known; in this example we use the calibrated input as the target output (reproduction)



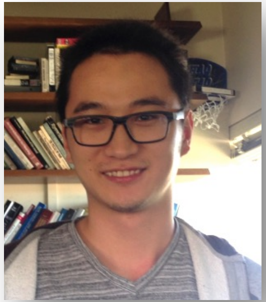
- We build a model of the camera (optics, sensor) and the image systems simulation software (ISET) produces raw sensor data

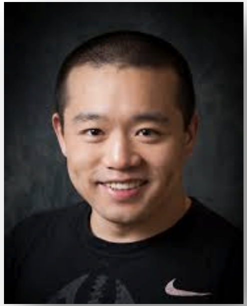
- Using many input images, we find the transformations from the sensor data classes to the desired output



For a description see the arXiv publication

- Haomiao, Qiyuan, Joyce and I designed and analyzed several image processing pipelines for novel sensor types in this paper





Learning the image processing pipeline

Haomiao Jiang, Qiyuan Tian, Joyce Farrell, Brian Wandell
Department of Electrical Engineering, Stanford University
Psychology Department, Stanford University

Abstract—Many creative ideas are being proposed for image sensor designs, and these may be useful in applications ranging from consumer photography to computer vision. To understand and evaluate each new design, we must create a corresponding image processing pipeline that transforms the sensor data into a form that is appropriate for the application. The need to design and optimize these pipelines is time-consuming and costly. We explain a method that combines machine learning and image systems simulation that automates the pipeline design. The approach is based on a new way of thinking of the image processing pipeline as a large collection of local linear filters. We illustrate how the method has been used to design pipelines for novel sensor architectures in consumer photography applications.

pipelines that are optimized for novel camera architectures. The general idea of the framework was first proposed by Lansel et al. in 2011[16]. Here, we introduce the framework in the form of a set of software tools that use simulation and learning methods to design and optimize image processing pipelines for these new camera systems.

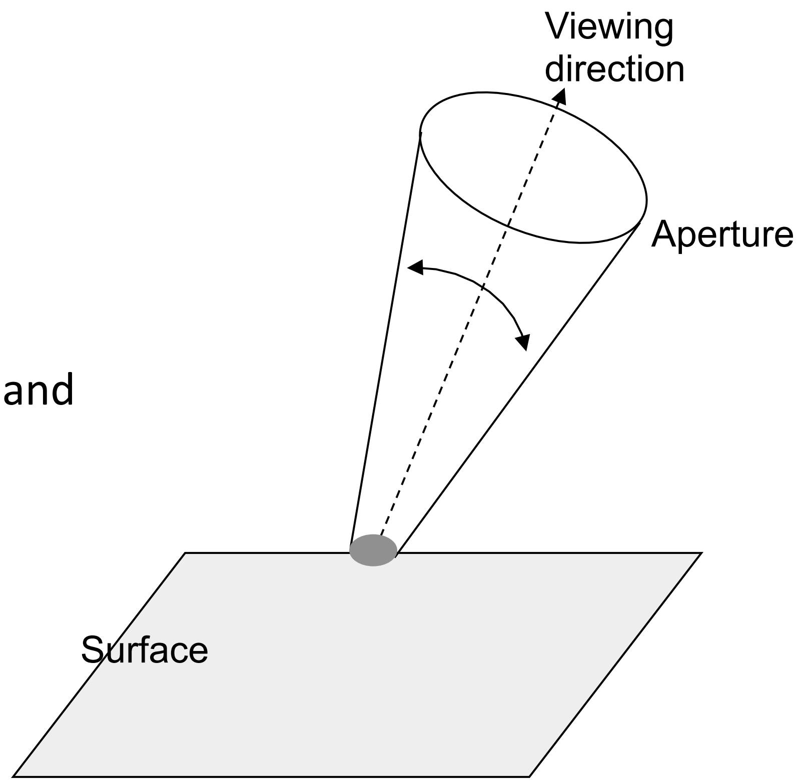
This paper is organized into three sections that define our contributions. First, we explain the image processing architecture: the input data are grouped by their local features into one of a set of local classes, where locality refers to both position on the sensor array (space), pixel type (color)

Classical spectral radiance

Definition: Light emitted or reflected from an extended source in a given direction. The light is specified in units of

energy/steradian/surface area/second/waveband

The surface area is foreshortened according to the viewing direction



Limitations: Scene data

- Scene radiance data is a significant limitation for certain applications
- Specialty cases can be difficult to acquire (e.g. under water, inside the body, automotive)
- Calibrated video data for simulation are unavailable
- **New approach:**
  Quantitative computer graphics

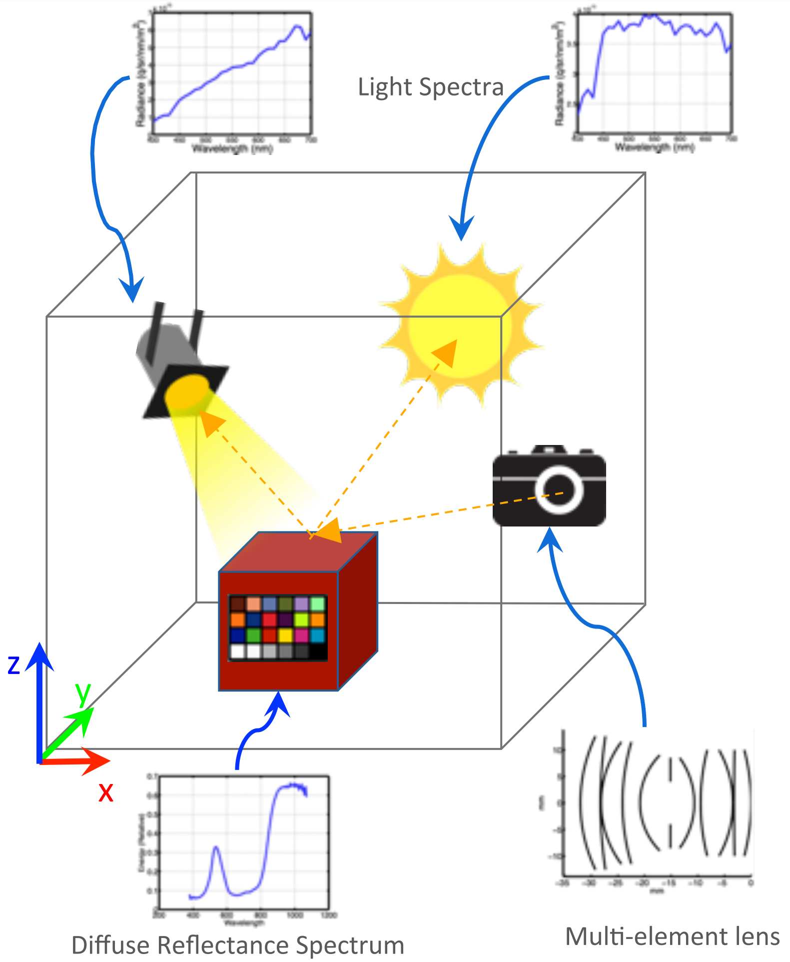
VNIR
(415:4:1000) nm

SWIR
(1000:4:2500) nm

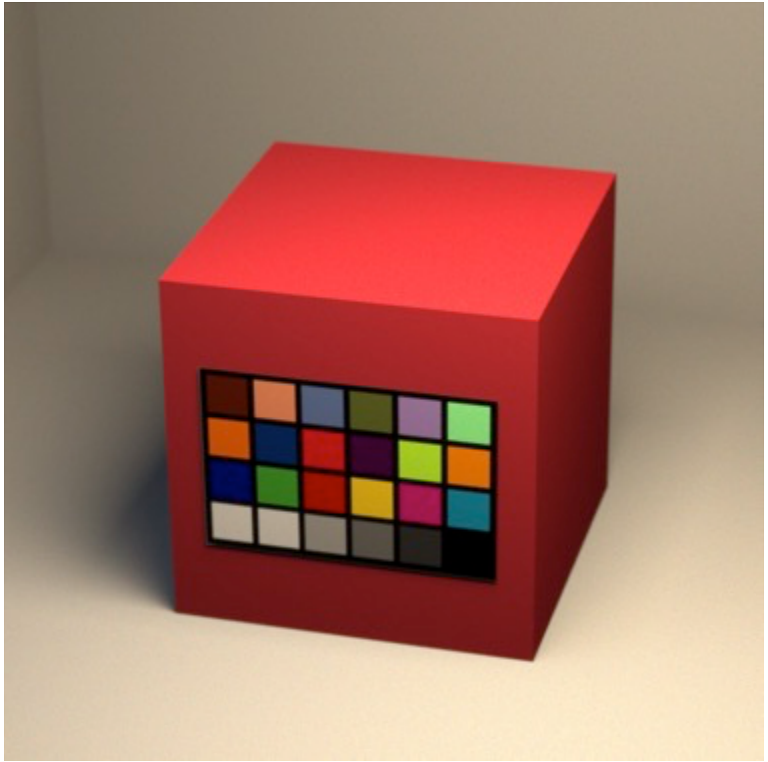
Blender and Physically based ray tracing (PBRT) (Pharr et al.)

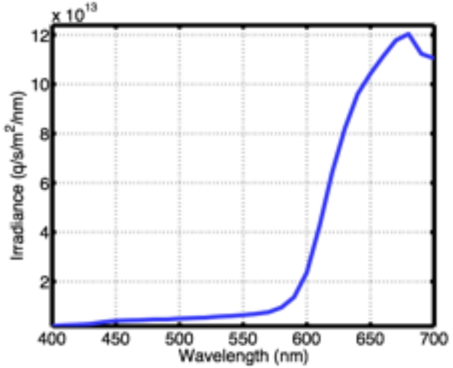
We are designing calibrated scenes for evaluating novel image acquisition systems (e.g., light field cameras, sensor arrays)

We are extending this system to different imaging applications (under water, endoscopes) and we hope to work with you on videos and **automotive** extensions.

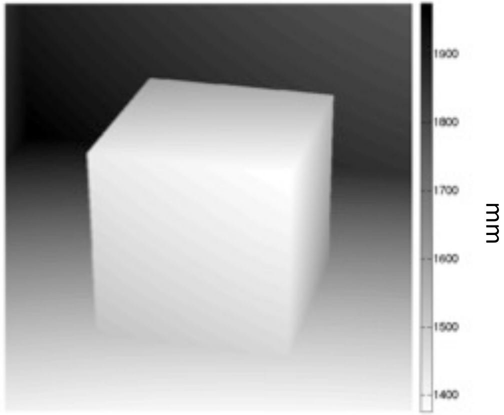


Quantitative computer graphics: Realistic and ground truth





- *Sensor irradiance Units: q/s/nm/m²*

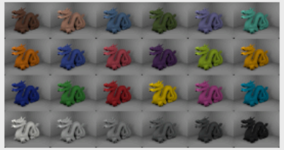


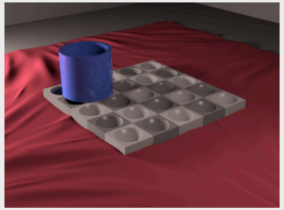
Depth

Control Blender/PBRT models and automate test scene design

(Heasly, Brainard, Tian, Blasinski)

Rendertoolbox3





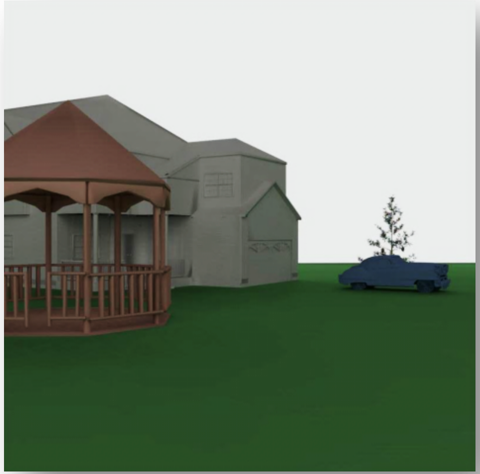
Welcome to RenderToolbox3!

RenderToolbox3 is a set of free and open-source Matlab tools that facilitate 3D rendering with physically-based renderers.

A particular focus is on easy manipulation of surface spectral reflectances, surface materials, and illuminant spectral power distributions. One target application for the toolbox is visual psychophysics, where often manipulating these variables is of experimental interest.

To learn more, please visit the RenderToolbox3 wiki, which contains a full set of documentation.

To obtain the software, please visit the main GitHub page.



Platform independence is important

Docker containers

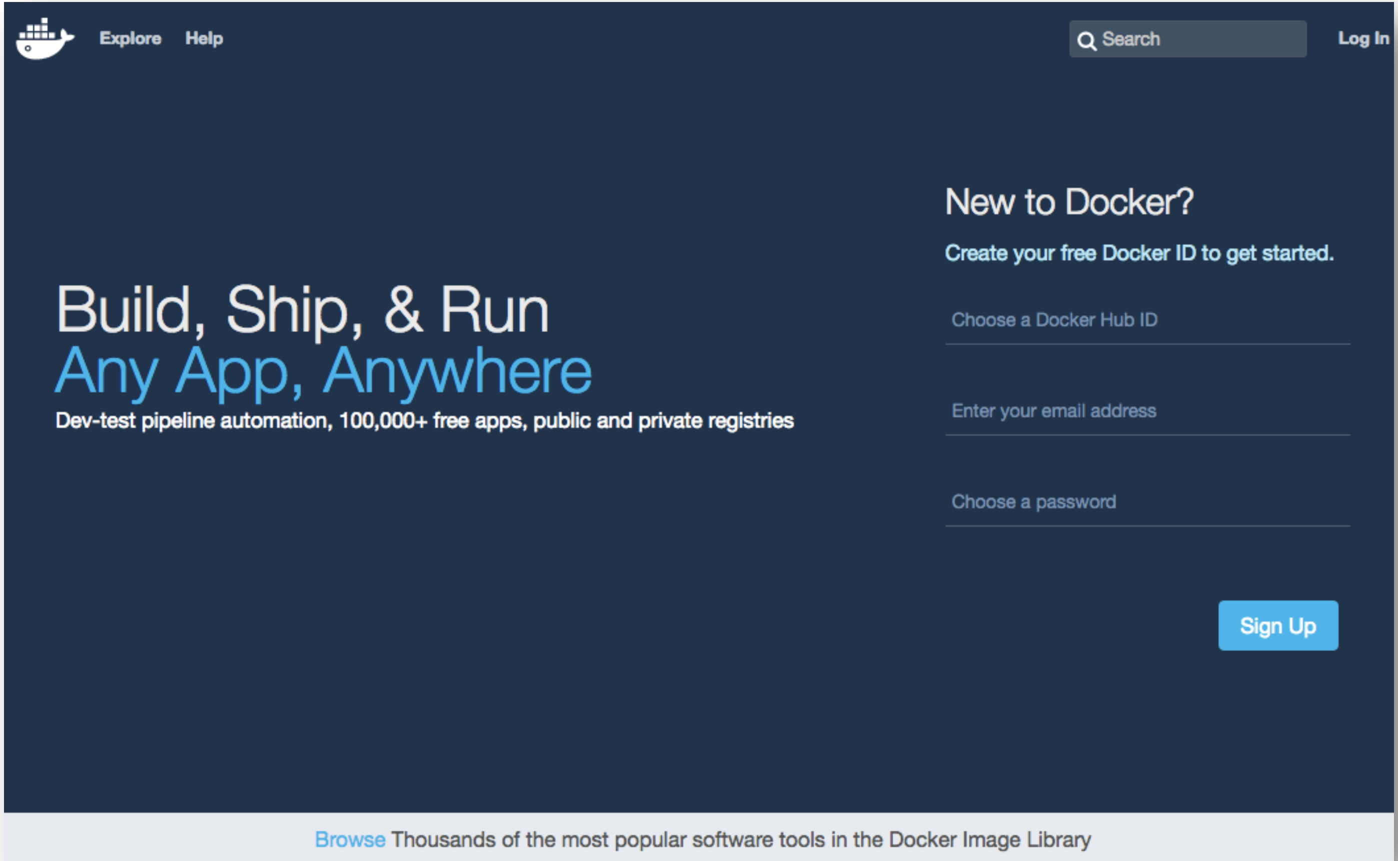
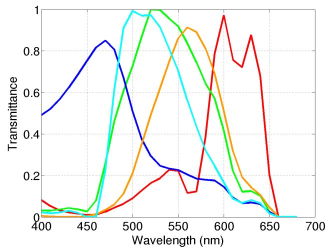


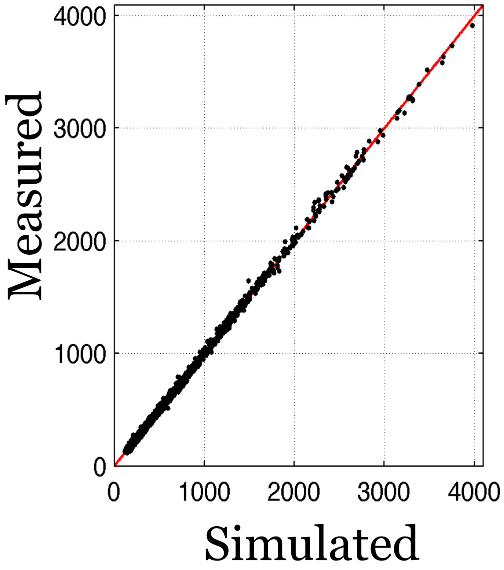
Image system simulation components

- Camera (optics and sensor) models for simple optics (isoplanatic with some geometric distortion) and single sensor CMOS imagers have been tested against data and work well.
- We are exploring more complex optics models (ray tracing, Zemax)

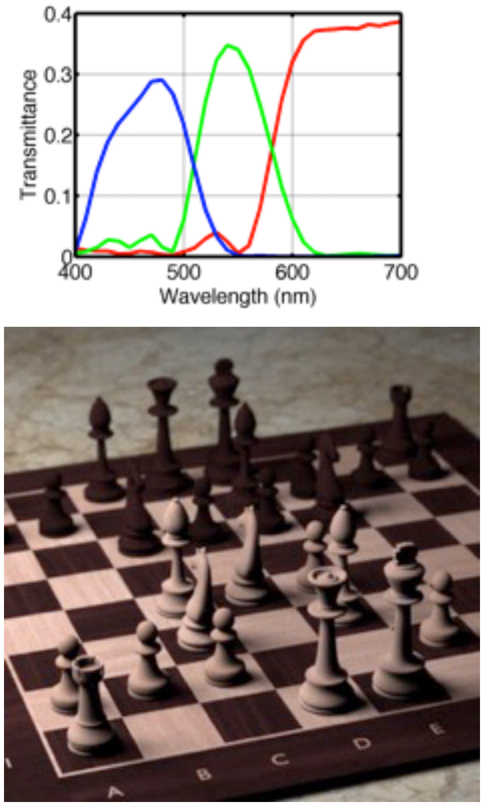
Simulated parameters (ISET)

Property	Value
Pixel Width/ Height (μm)	5
Fill Factor	0.5
Dark Voltage (V/sec)	0.0004 8
Read Noise (V)	0.0017
Dark Signal Non-uniformity (V)	9.07e-5
Photo Response Non-uniformity (%)	0.017
Conversion Gain (V/e-)	0.0001
Voltage Swing (V)	1
Well Capacity (e-)	24,000
Analog Gain	2.12
Analog Offset (V)	0.056
F Number	2.8
Focal Length (m)	0.05

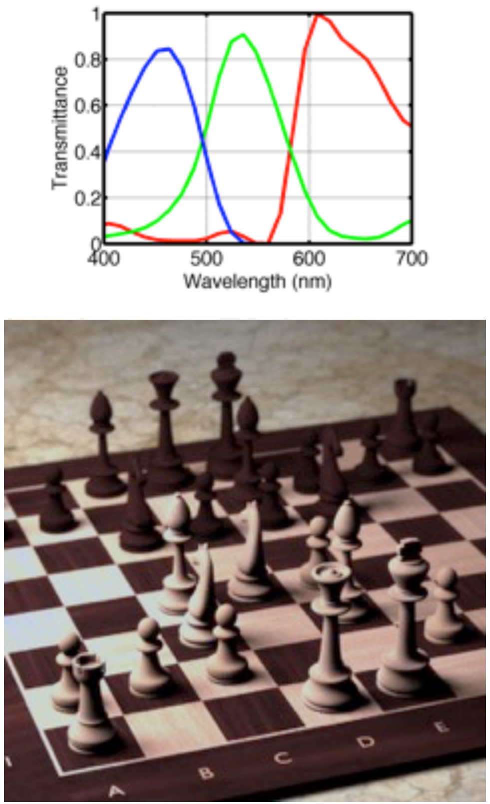




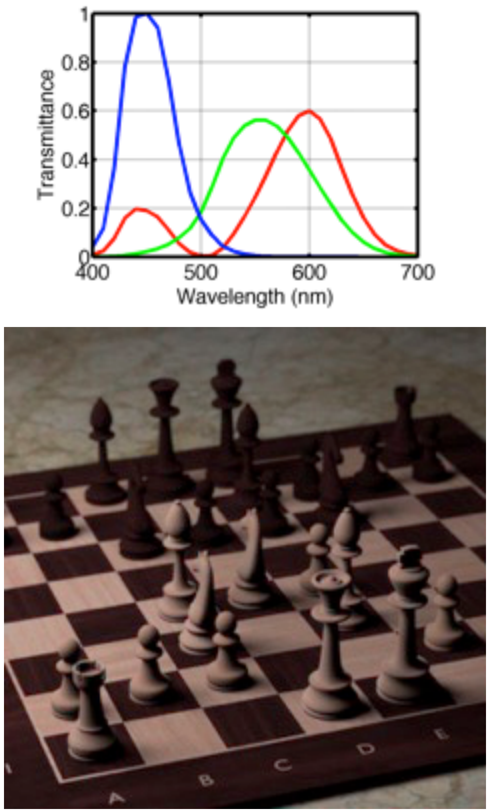
Camera simulation: Varying color filter transmittance



Default RGB



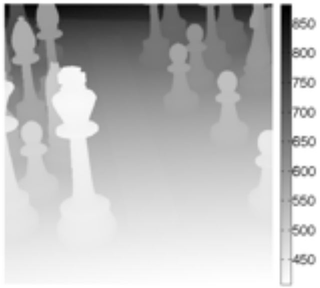
Nikon D200



XYZ Curves

Optics simulation

- Varying aperture size (depth of field)



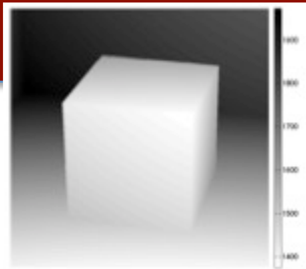
Depth map

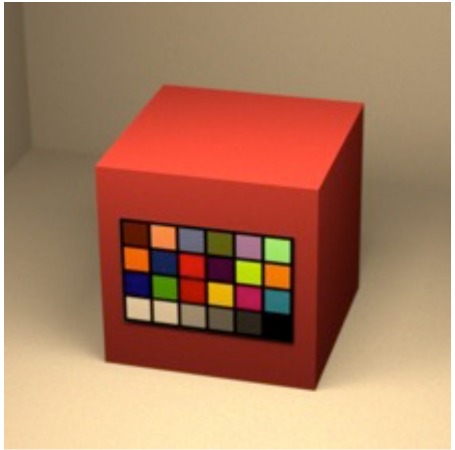
f/25
(small aperture)

f/5
(medium aperture)

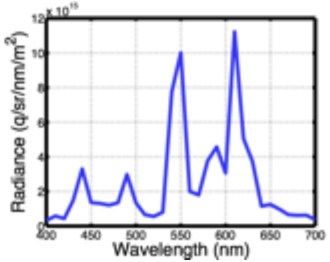
f/2.5
(large aperture)

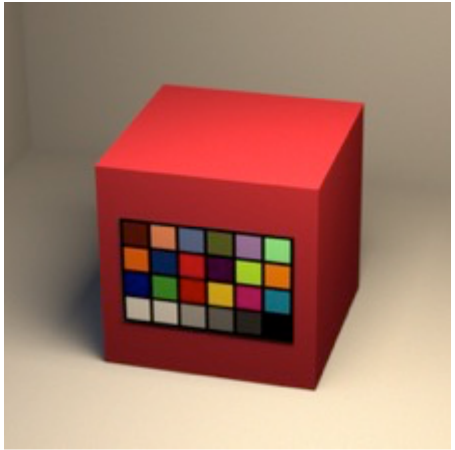
Varying illumination spectrum



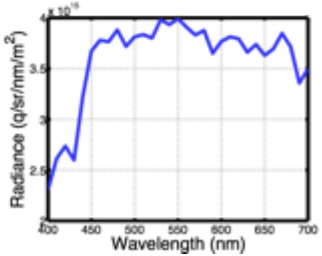


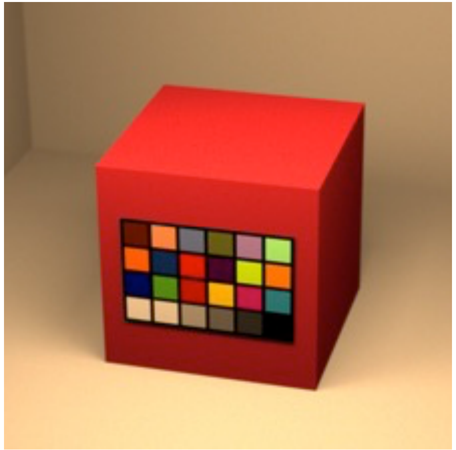
Fluorescent



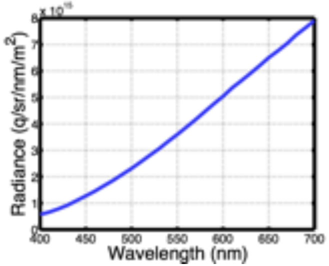


D65



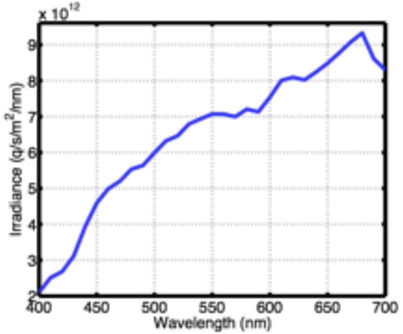


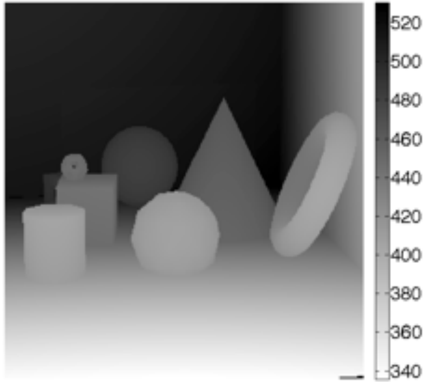
Tungsten



Quantitative computer graphics

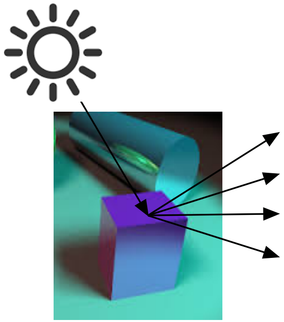
- Generate many scenes with many points-of-view, illuminations, arrangements of objects
- Ground truth





The Image Systems Engineering Toolbox for Biology (ISETBIO)

Software designed for predicting the responses of the front end of biological visual systems and analyzing the information to the brain



Scene representations

Physiological optics

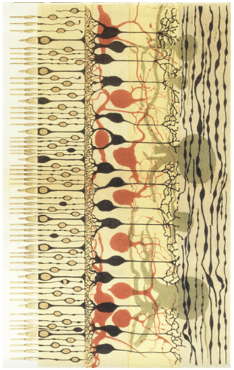
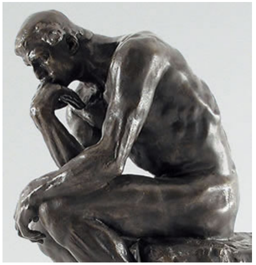


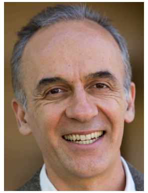
Photo transduction

Retinal processing



Inference

David Brainard



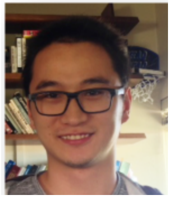
E.J. Chichilnisky

Fred Rieke

Brian Wandell

Jon Winawer

Nicolas Cottaris



Haomiao Jiang

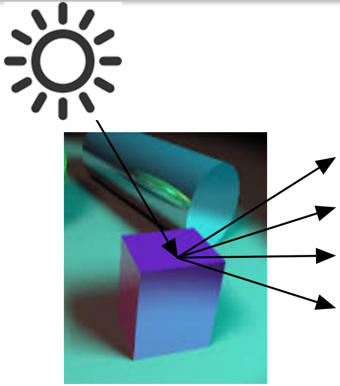
Xiaomao Ding

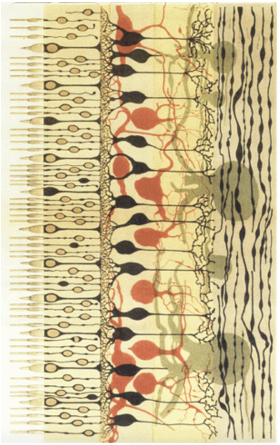
James Golden

Joyce Farrell

SIMONS FOUNDATION

Seeing the big picture: A computational framework for integrating vision





Spectral radiance, theory of light, refraction, optics, Snell's law, diffraction, Airy disk, photons, energy, Planck's constant, wave theory, chromatic aberration, light fields, wavefronts, multidimensional linear algebra, rods and cones, trichromacy, color-matching functions, photopigments, transduction, photocurrent, retinal anatomy, receptive fields, convolution, normalization, linear-nonlinear models, linear classifier theory, ideal observer theory ...

SIMONS FOUNDATION

Modeling human optics

Lens spectral transmissivity

Corneal and lens refraction

Accommodation

Chromatic aberration

Wavefront characterization

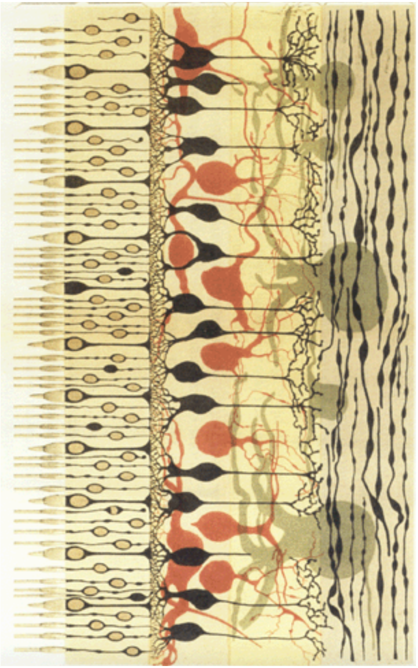
- Standard observer
- Defocus
- Other aberrations (Zernike Polynomials)

Physiological optics

Modeling the retinal image processing pipeline

Macular pigment
Eye movements
Cone absorptions
Retinal inhomogeneity (fovea, periphery)
Cone photocurrent
Bipolar cell current
Retinal ganglion cell mosaics (spiking)

- On/Off midgets
- On/Off parasol
- Small bistratified



Retinal processing

Modeling a computational observer

Classifiers

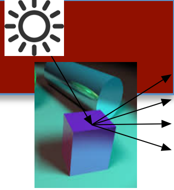
- Support Vector Machines (SVM)
- Bayesian estimators

Machine learning

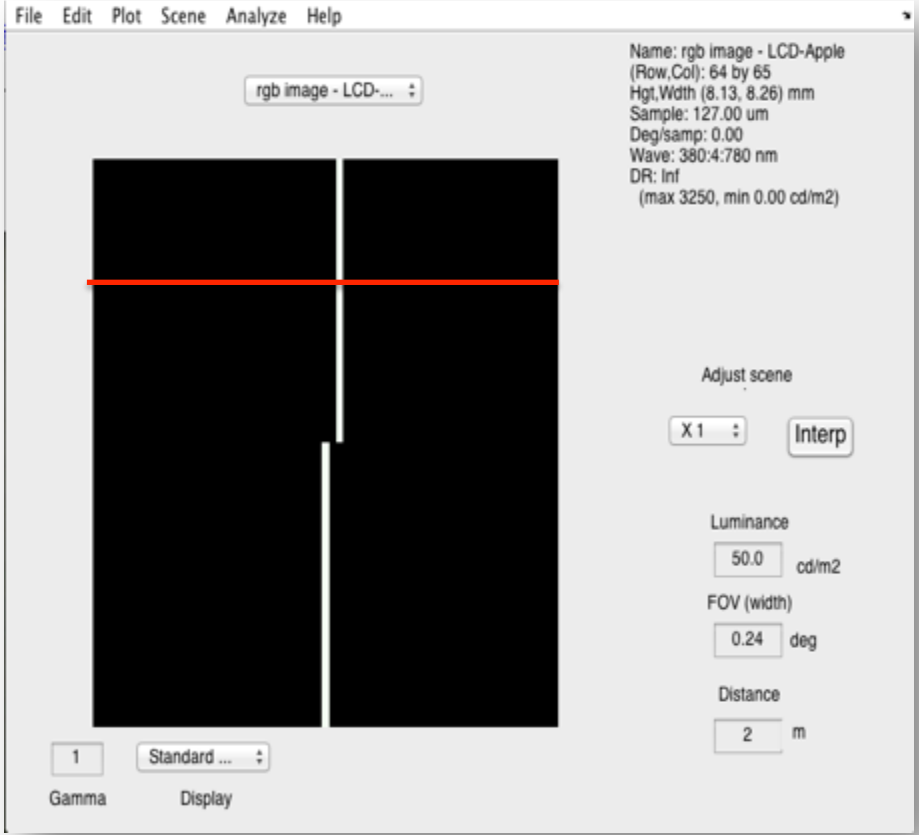
- TensorFlow (Deep learning)

Computational Observer

Example: Positional acuity (Vernier)

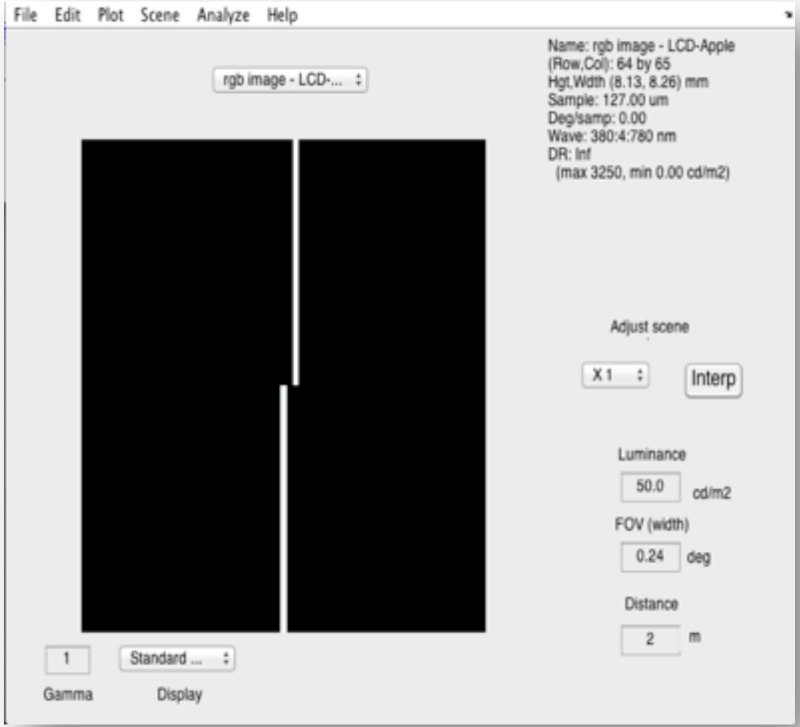


Display stimulus as scene spectral radiance




```
%% Create a calibrated display object

% Display object
dpi = 110;
d = displayCreate('OLED-Sony', 'dpi', dpi);

% Show image
p.offset = 2; p.sceneSz = 64; p.display = d;
scene = sceneCreate('vernier','object',p)

ieAddObject(scene);  sceneWindow;
```

Image formation




```
%% Create the retinal irradiance

% Human optics
oi = oiCreate('wvf human')
oi = oiCompute(oi,scene);
```

Image formation

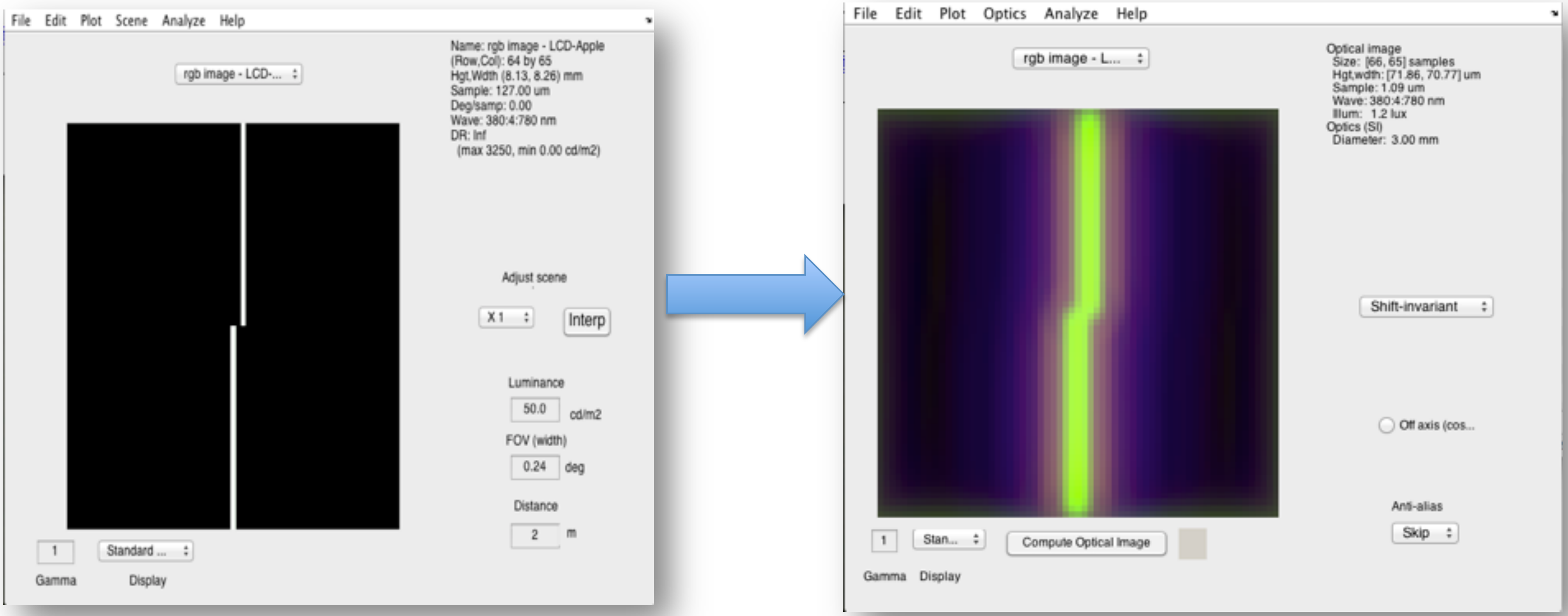
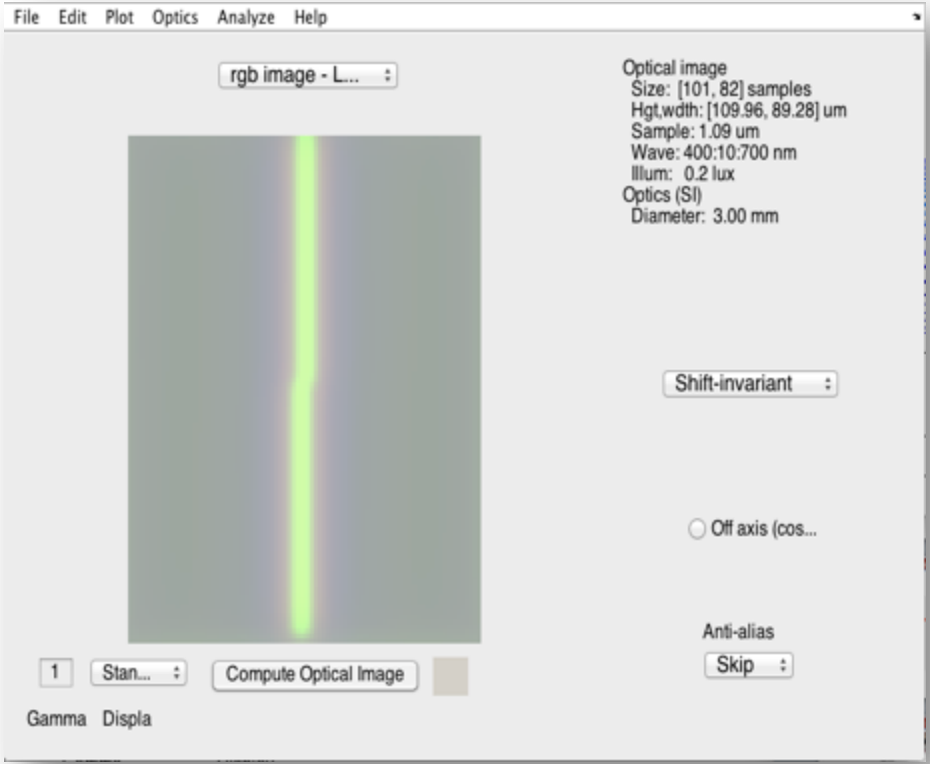


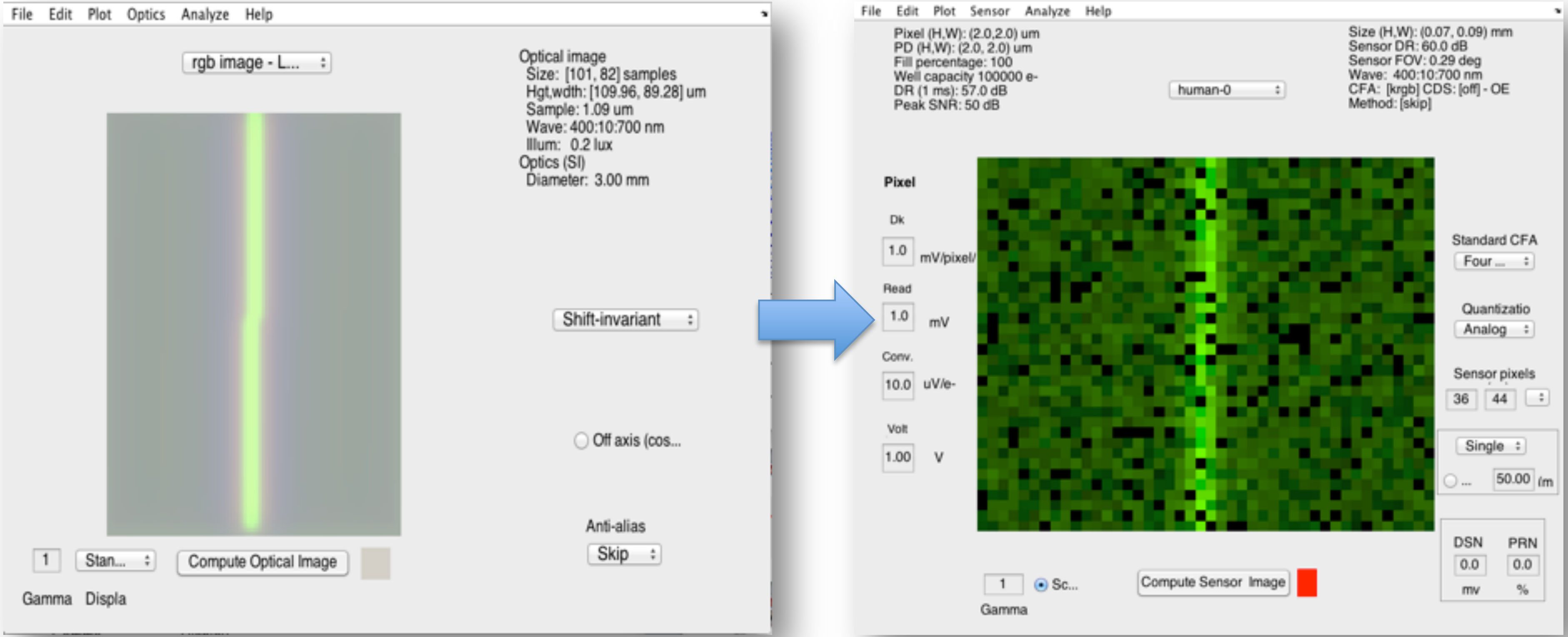
Image formation



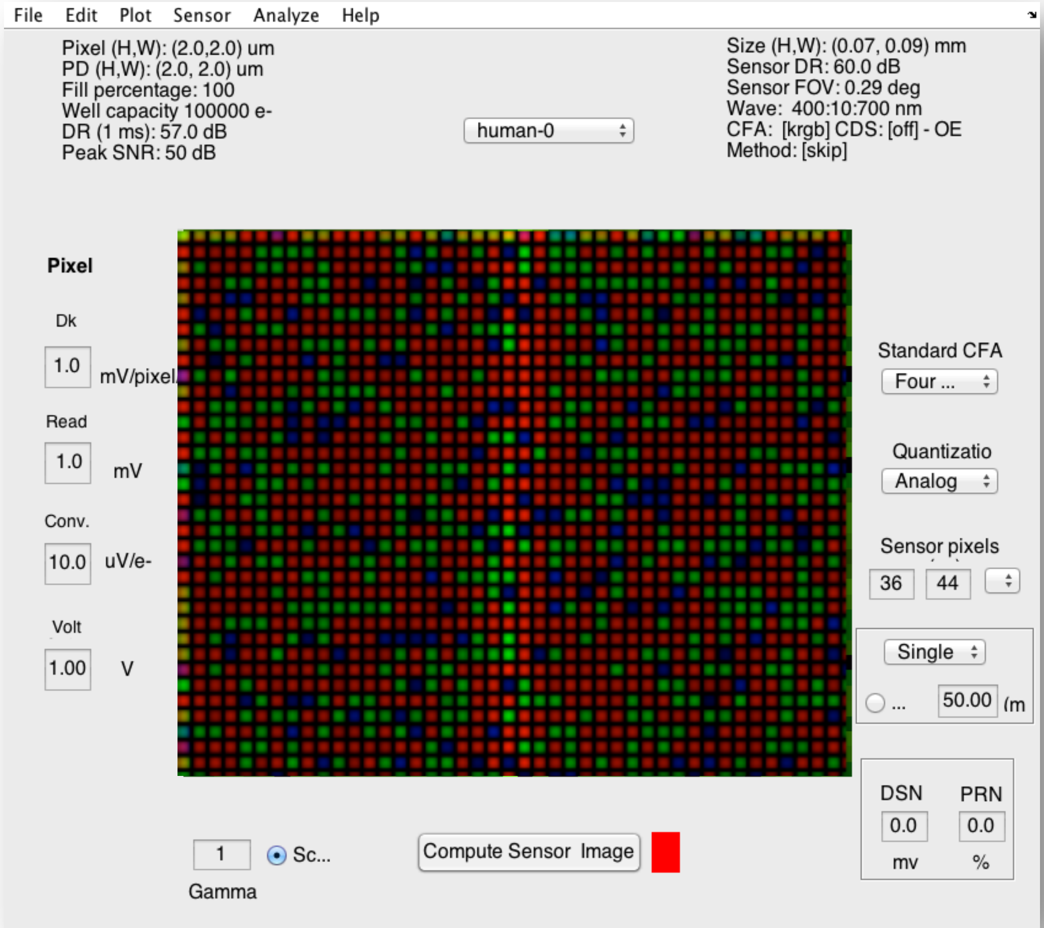
Let’s do this experiment on a uniform gray background to set the mean adaptation level and isolate the cones

```
%% Calculate the cone absorptions

cones = sensorCreate(‘human’)
cones = sensorCompute (cones,oi);
```

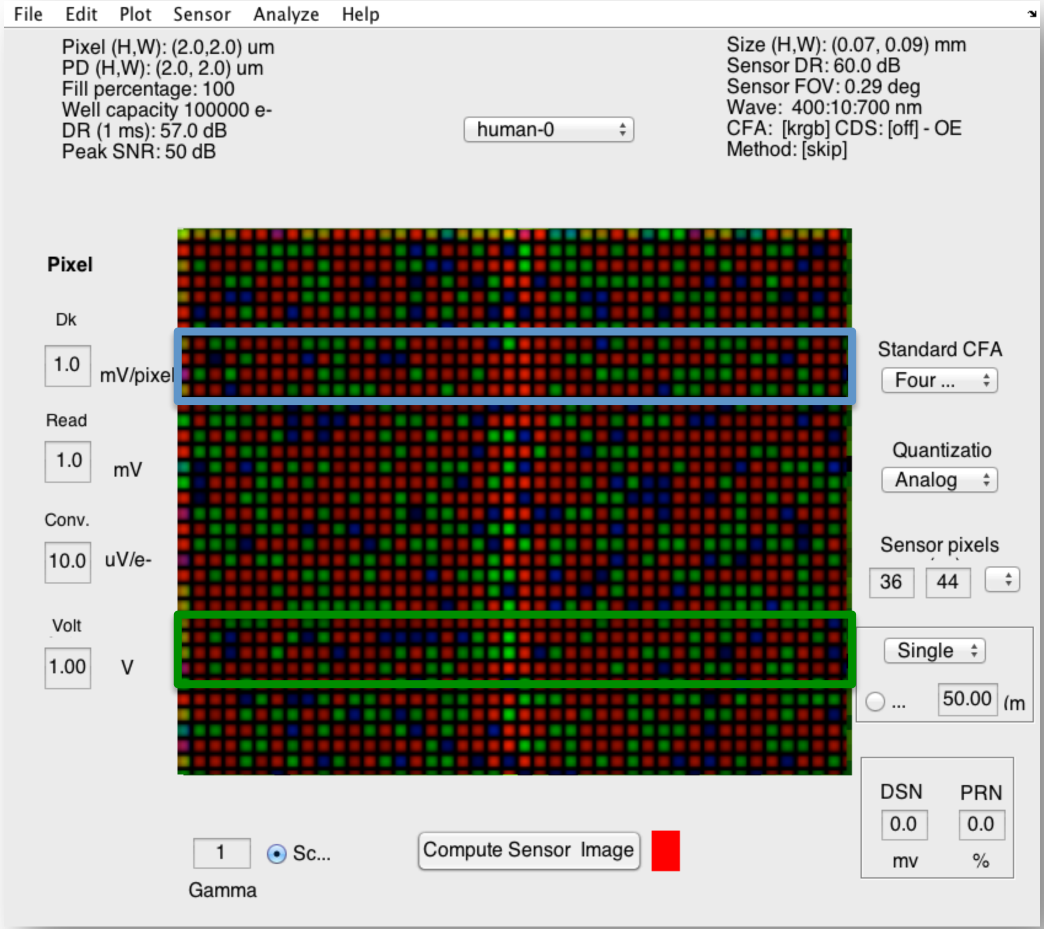
Phototransduction

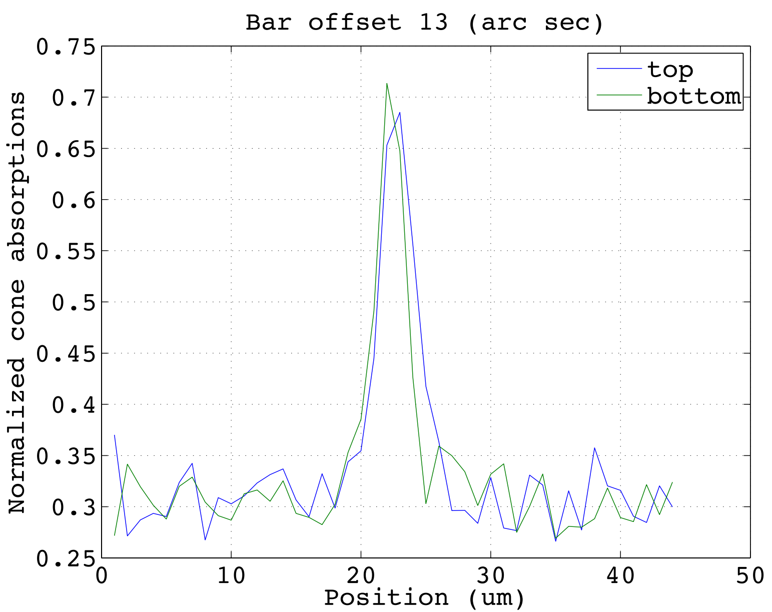


Phototransduction

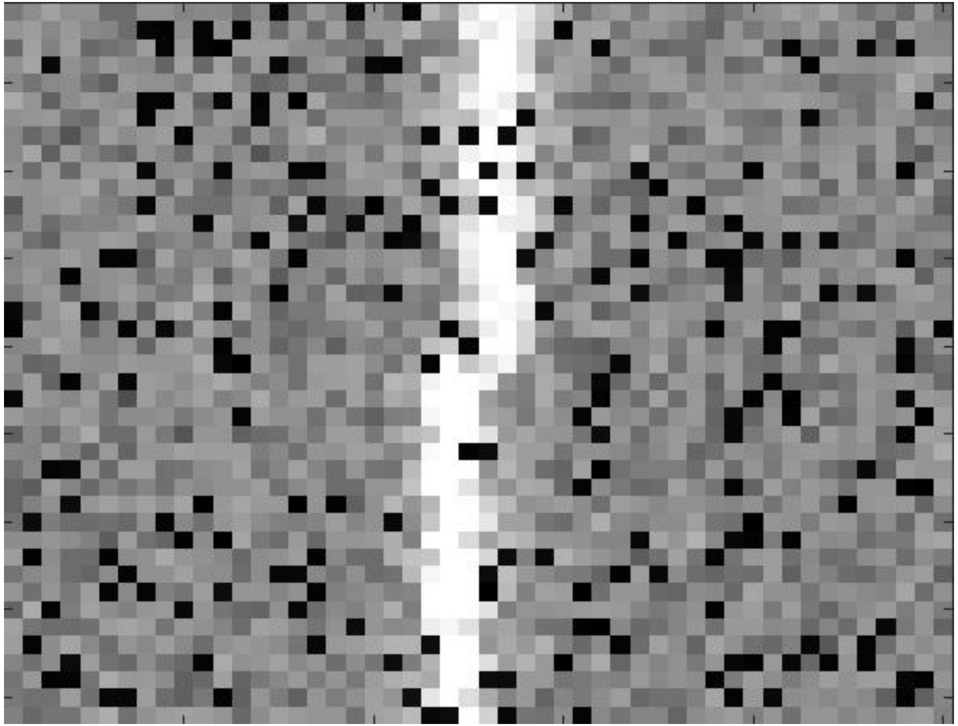


Phototransduction





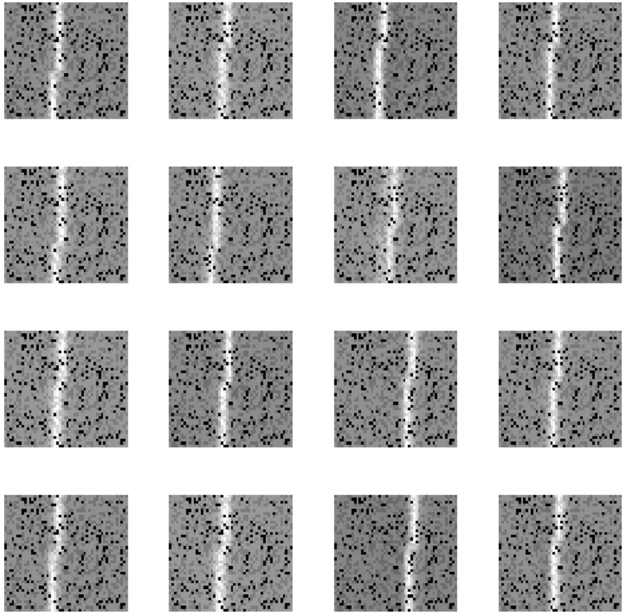
Photon absorptions and eye movements

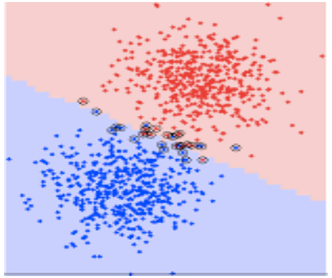


Classification

Classifier

- Linear SVM
- Training samples





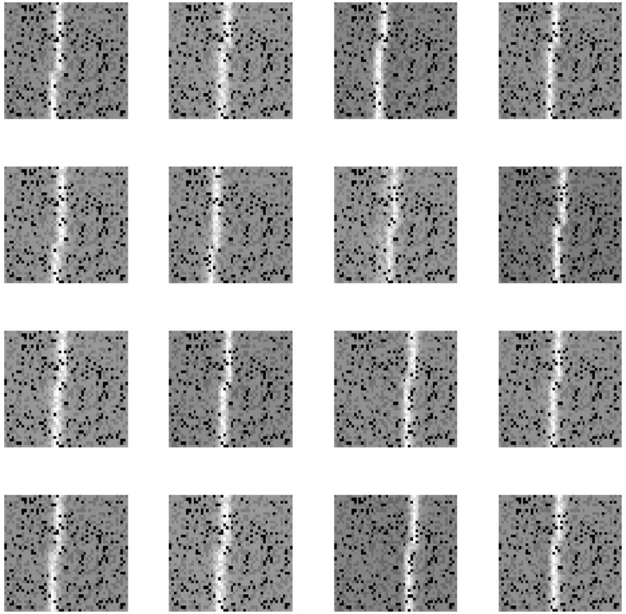
Experiment

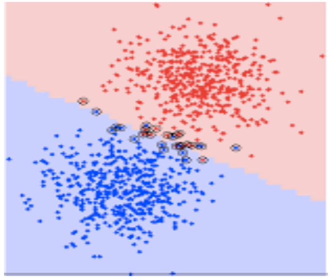
- Apple-LCD display
- 0.1 deg field of view
- 50 ms exposure
- 24 cd/m^2 mean luminance
- 0.3 m = viewing distance
- Cone aperture: 2 um, …

Classification

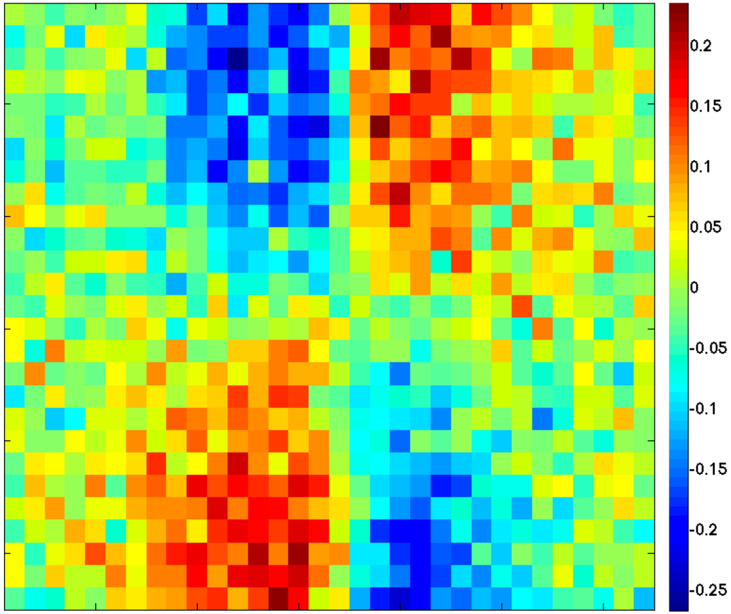
Classifier

- Linear SVM
- Training samples





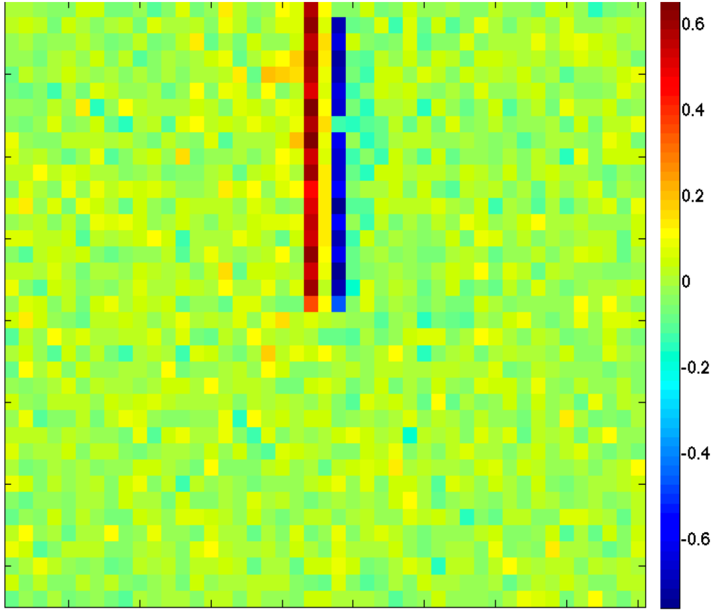
Classifier weights
(eye movements)



Computational observer
accuracy: 75%

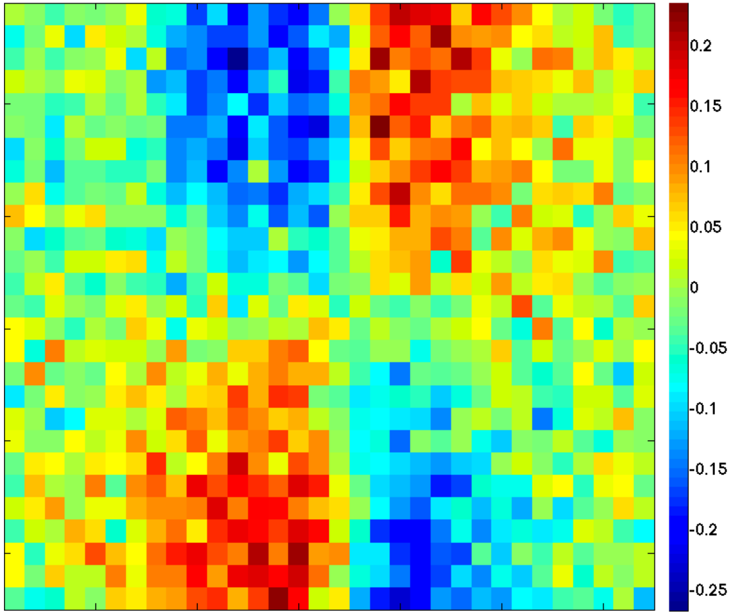
Classification

Classifier weights
(no eye movements)



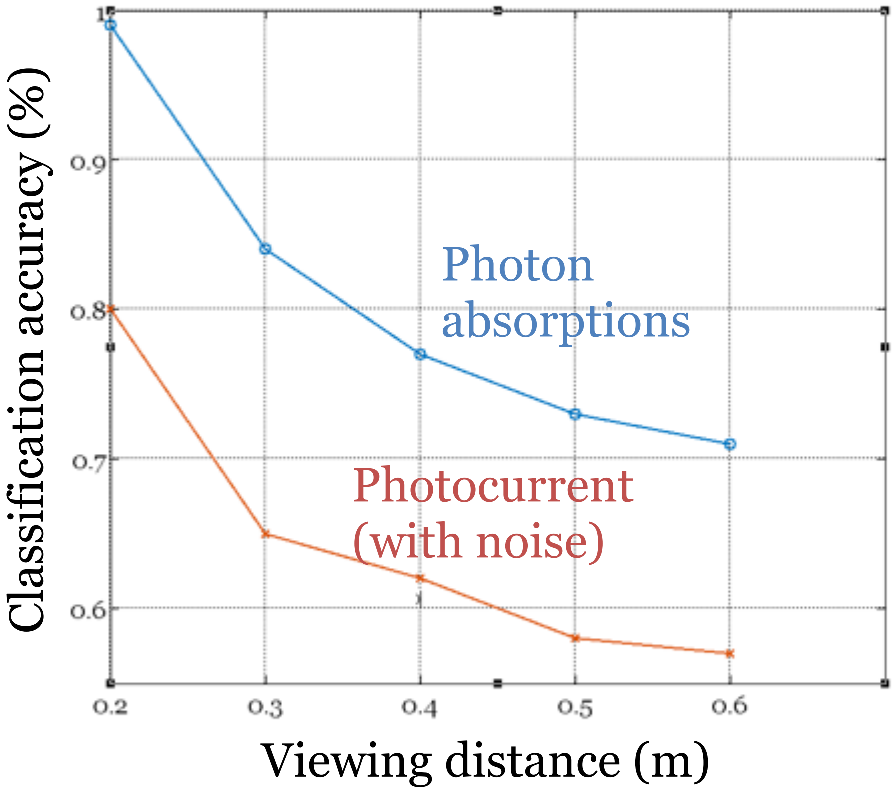
Computational observer
accuracy: 100%

Classifier weights
(eye movements)

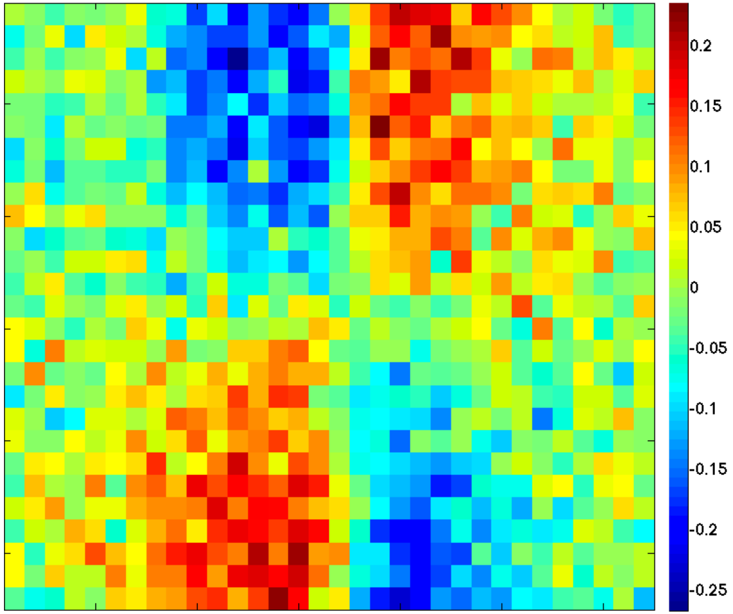


Computational observer
accuracy: 75%

Classification



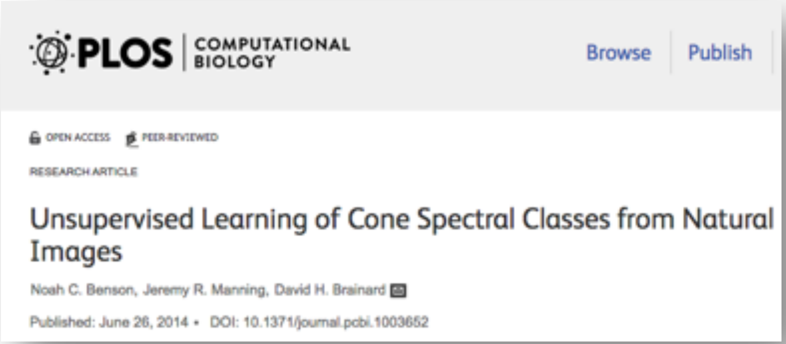
Classifier weights (eye movements)



Computational observer accuracy: 75%

Scene radiance from image databases

scene


PLOS | COMPUTATIONAL BIOLOGY   Browse   Publish

OPEN ACCESS   PEER-REVIEWED

RESEARCH ARTICLE

Unsupervised Learning of Cone Spectral Classes from Natural Images

Noah C. Benson, Jeremy R. Manning, David H. Brainard

Published: June 26, 2014 • DOI: 10.1371/journal.pcbi.1003652

• Built from spectral images in 3 open databases (Brelstaff et al. 1995; Foster et al. 2004; Chakrabarti & Zickler 2011).

• Neglect 3D scene structure for now (assume far away).

Convert scene radiance to retinal irradiance

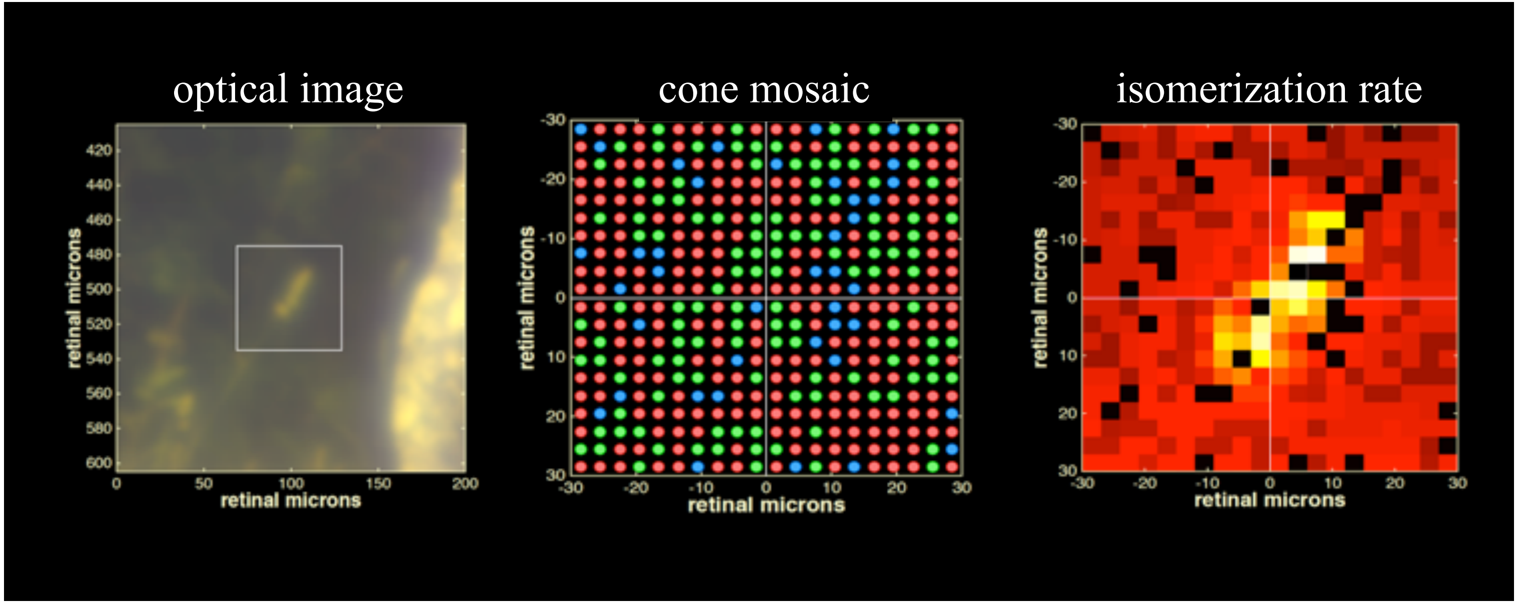
optical image

- Model blurring (including axial chromatic aberration) by the eye's optics; also pupil size, geometry of eye.

- Yields spectral retinal irradiance in physical units (e.g., quanta/[sec-m^2-nm]).

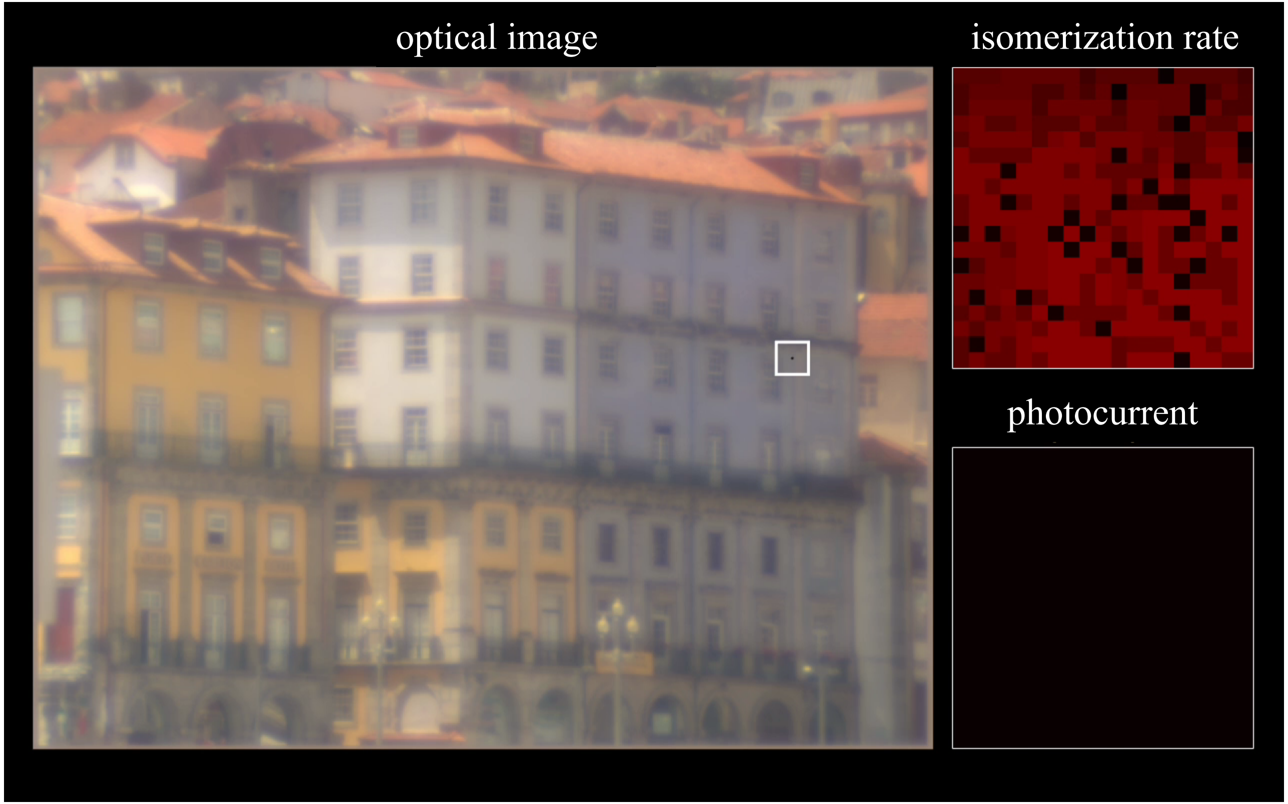
```
% Create optical image object
% and compute from scene.
oi = oiCreate('human');
opticalimage = oiCompute(oi,scene);
```

Define cone positions and types, compute isomerization rates




```
% Create human sensor object
% and compute from optical image.
sensor = sensorCreate('human');
sensor = sensorCompute(sensor,oi);
```

Retinal image -> isomerizations -> photocurrent, with eye movements

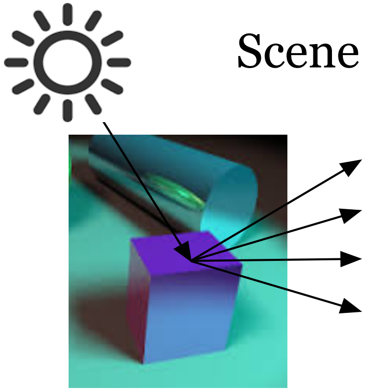



```
% Compute photocurrent
% at each eye position
sensor = ...
sensorSet(sensor, ...
'positions', ...
sensorPositions);

sensor = ...
coneAbsorptions(sensor, ...
opticalImage);

os = osCreate;
os = osCompute(os,sensor);
```

Computational Image Systems Engineering Toolbox (CISET)

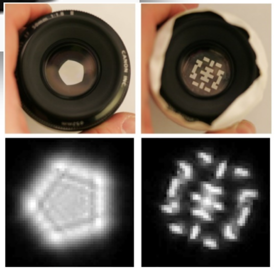
There are many new devices and applications for VR, AR, computer vision, and vehicles

Scene



360 camera arrays

RGB-D

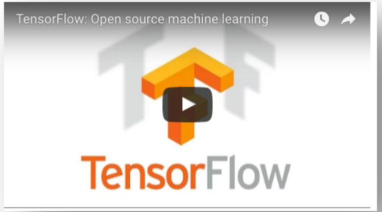


(a) Conventional (b) Coded

Coded apertures

Processing

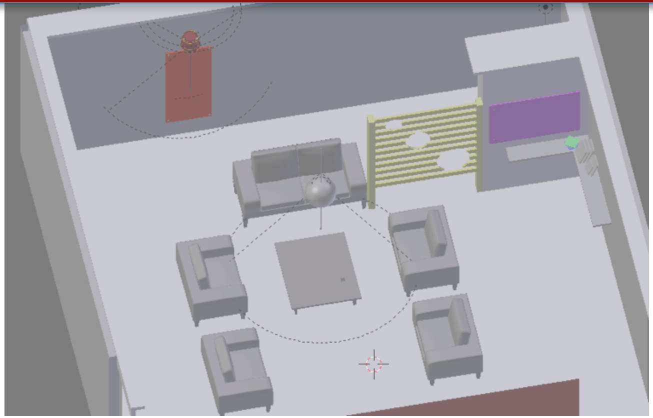
OpenCV



Simulating camera arrays

- 16 Cameras (15° intervals)
- 35 mm simple lens (f/5.8)
- 28 x 19 mm sensor
- ~40° FOV
- 200 mm camera rig radius

3D model view from Blender



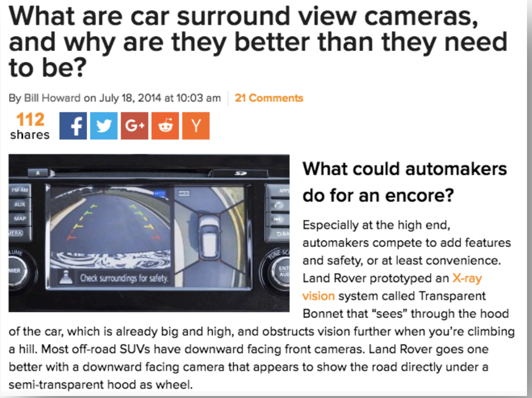
PBRT Render

Simulating camera arrays

Multiple outward looking cameras for VR

Cars already ship with surround view

Design for other control purposes?



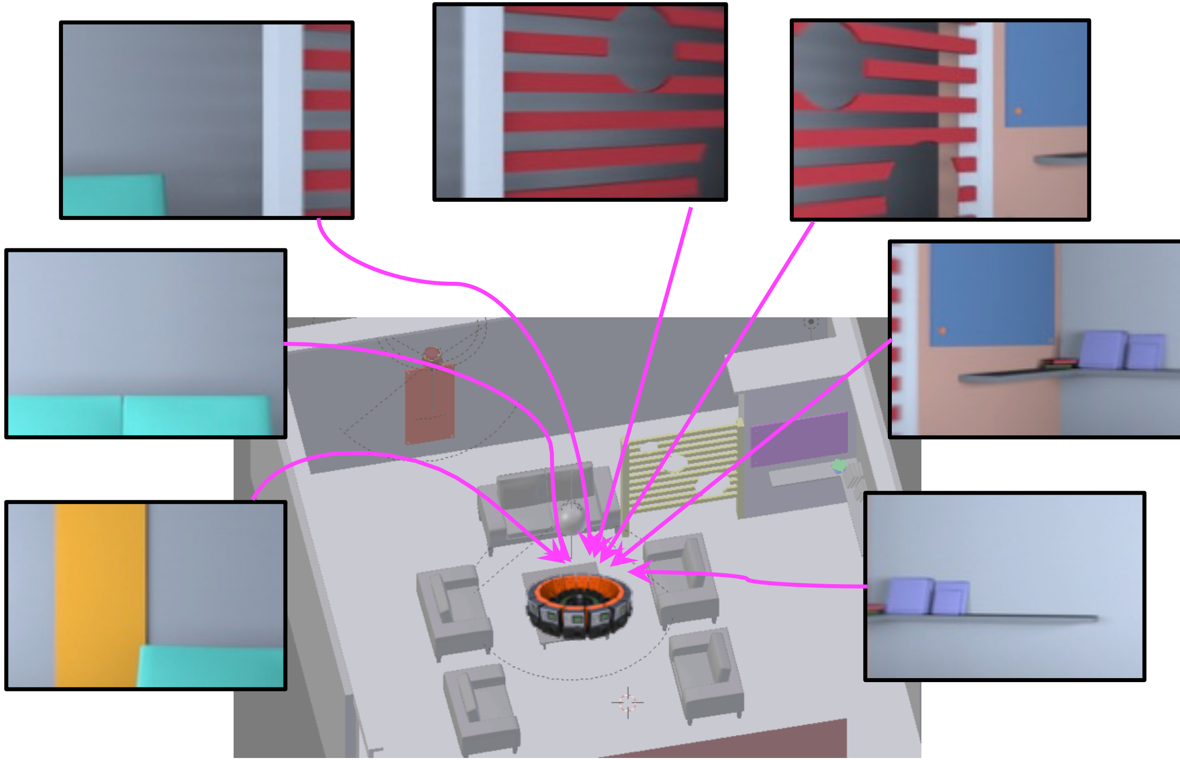
What are car surround view cameras, and why are they better than they need to be?

By Bill Howard on July 18, 2014 at 10:03 am | 21 Comments

112 shares

What could automakers do for an encore?

Especially at the high end, automakers compete to add features and safety, or at least convenience. Land Rover prototyped an X-ray vision system called Transparent Bonnet that "sees" through the hood of the car, which is already big and high, and obstructs vision further when you're climbing a hill. Most off-road SUVs have downward facing front cameras. Land Rover goes one better with a downward facing camera that appears to show the road directly under a semi-transparent hood as wheel.

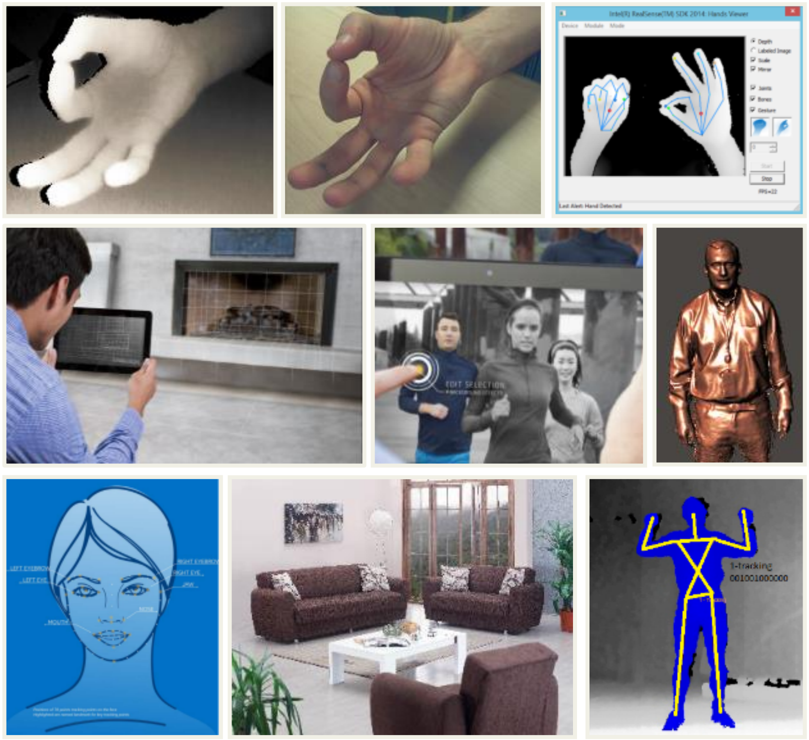


Very thin, small commercial imagers for RGB-D

AR and drone applications

Model: F200
(Close-range usages)

Model: R200
(Longer-range usages)



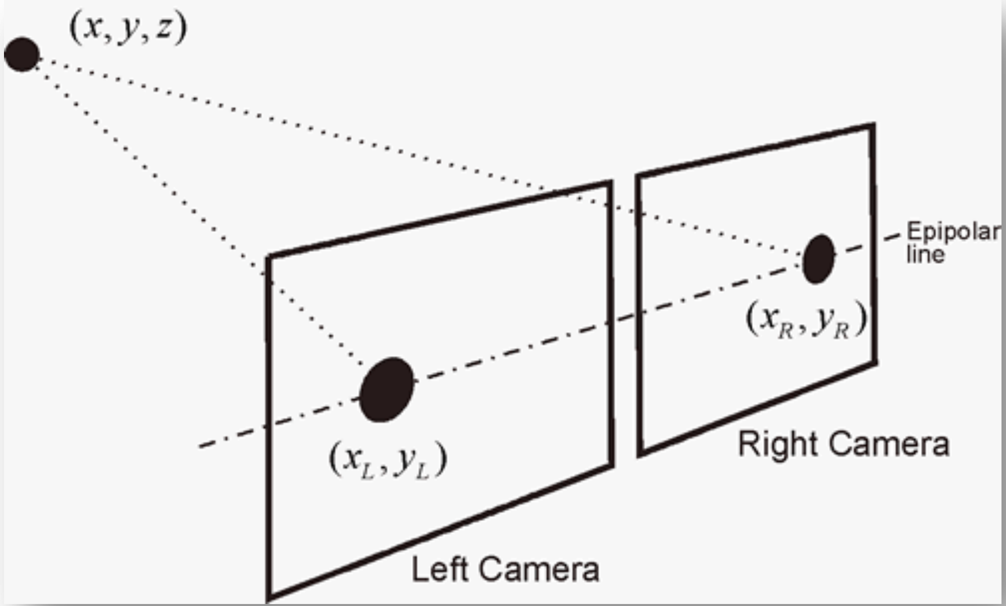
Software Libraries:
“Understand What You See”

Simulations for new classes of cameras (RGB-Depth)

Classic stereo-3D Imaging

Like having two eyes

Works over a distance that depends on the baseline



Simulations for new classes of cameras (RGB-Depth)

Structured/coded light

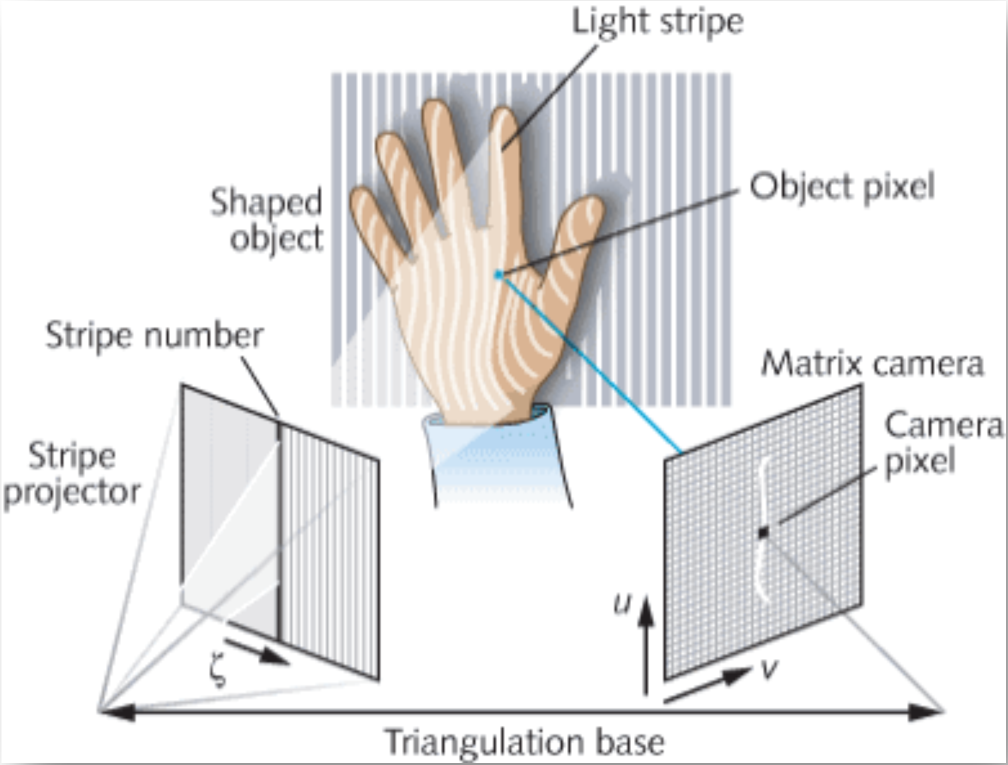
Project known patterns into the environment

Grid is shown, but it could be a texture pattern

Infer depth

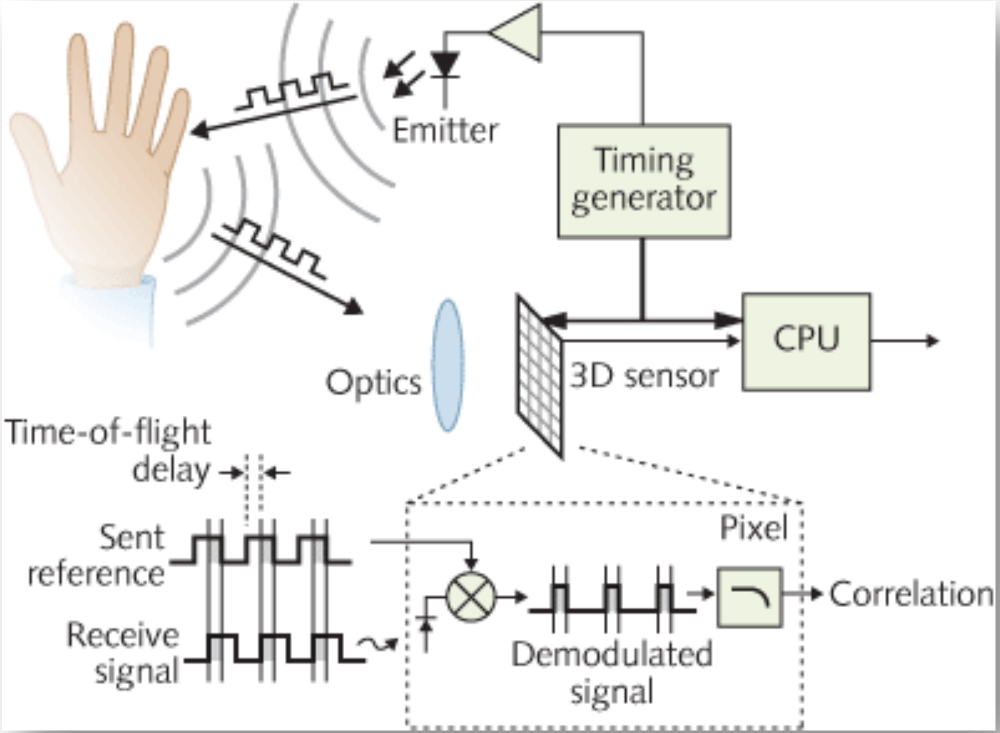
Limited by environmental illumination

Best when projected light is in non-visible wavelength



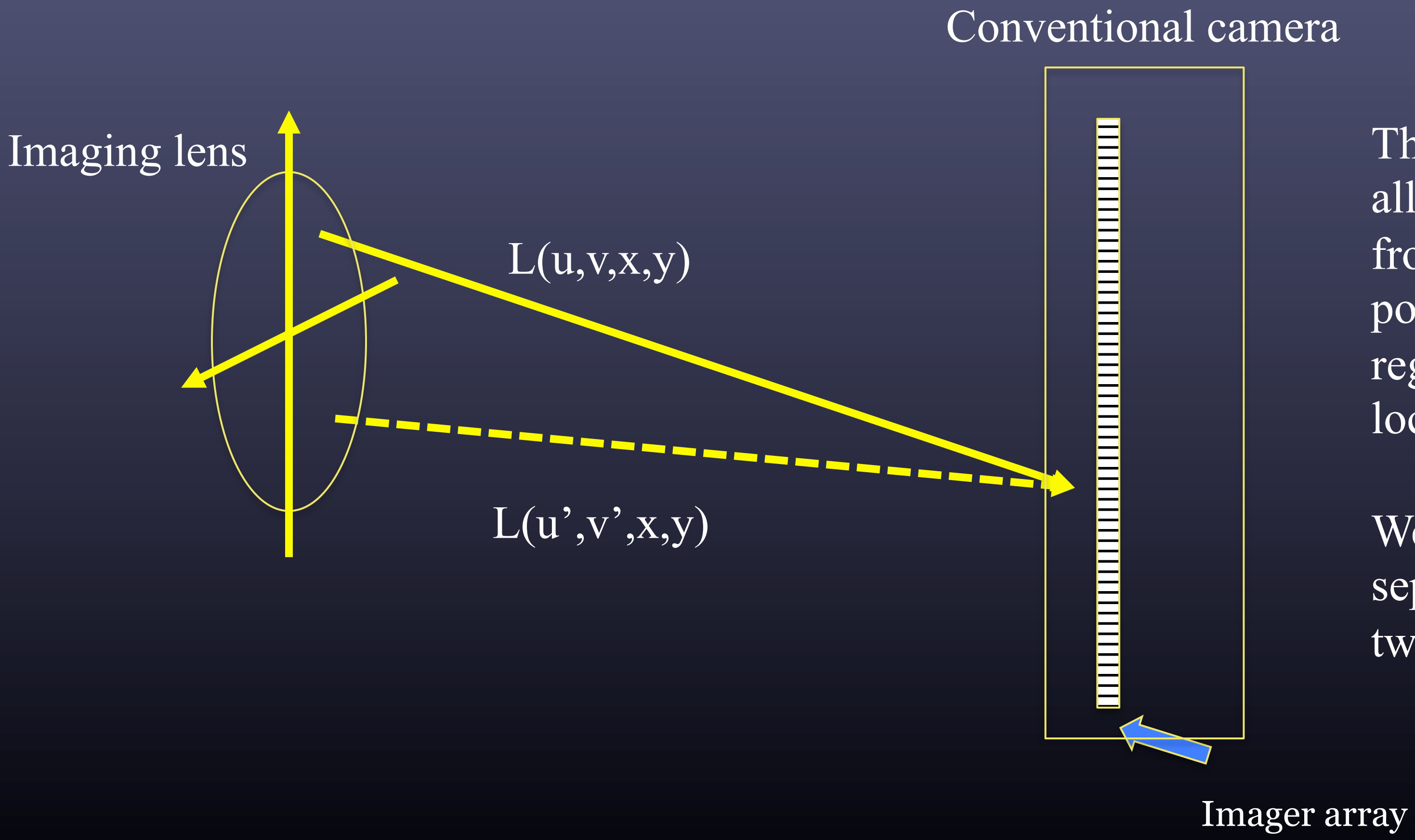
Simulations for new classes of cameras (RGB-Depth)

- Time-of-flight (Canesta/Microsoft)
- Embedded within the imager itself



Light field imager

Measures the intensity of each L(u,v,x,y) ray

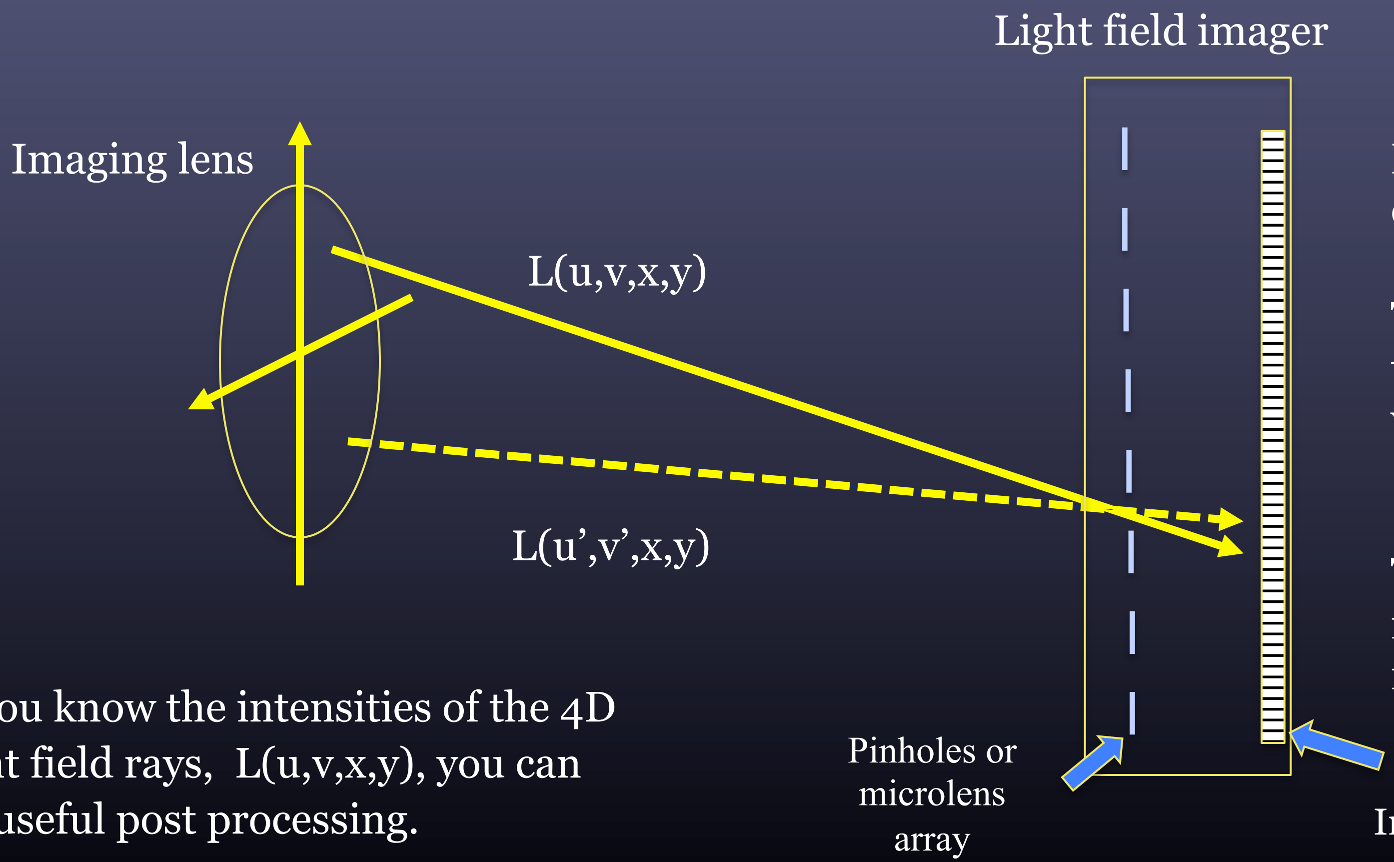


The pixel sums all the intensities from many (u,v) positions in the region near pixel location (x,y).

We do not separate out the two rays.

Light field imager

Measures the intensity of each L(u,v,x,y) ray



In a light field camera --

The position in the lens is a (u,v) value

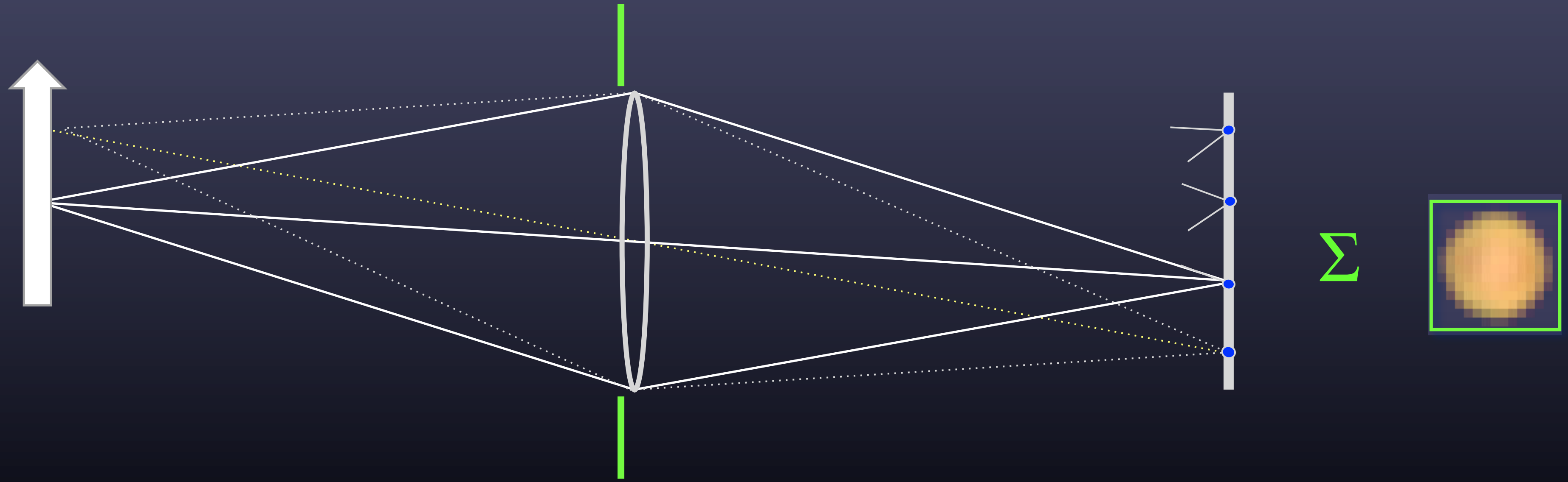
The position of the microlens is the region near (x,y)

If you know the intensities of the 4D light field rays, L(u,v,x,y), you can do useful post processing.

Simulating a reduced aperture

Reducing the aperture increases the f/#

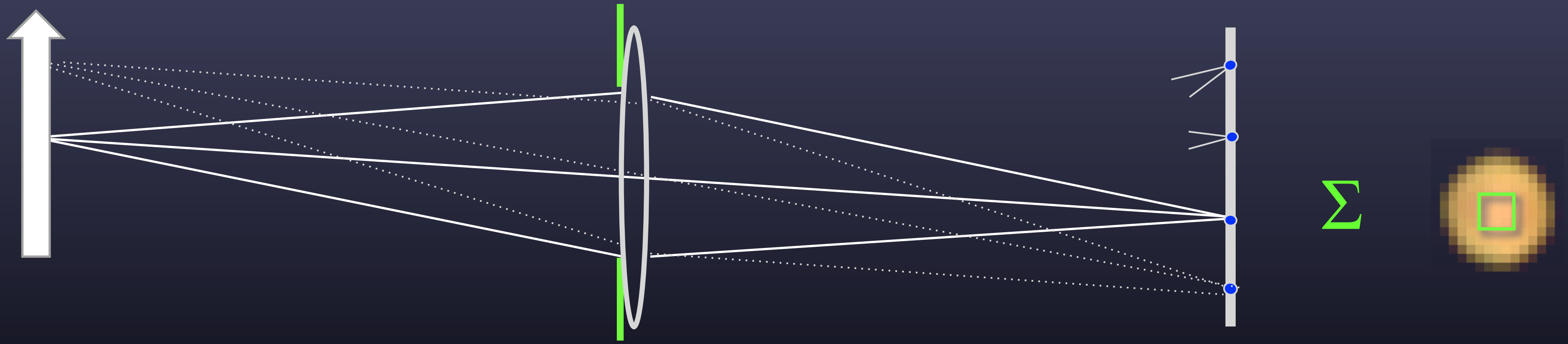
Simulate by summing only the central rays in each microlens



f/# = focal length / aperture

Simulating a reduced aperture

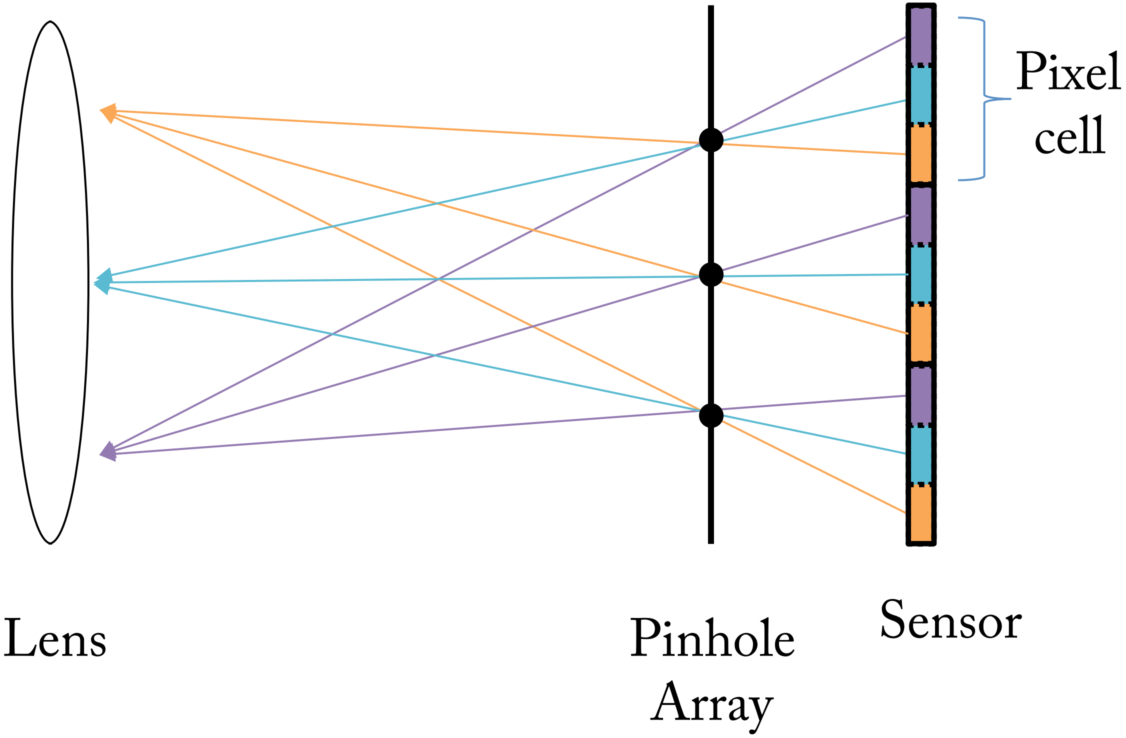
Reducing the aperture increases the f/#
Simulate by summing only the central rays in each microlens



f/# = focal length / aperture

Light field camera

- The pixels with the same color look at the image through a pinhole on lens
- This is like having many different points of view on the scene
- The baseline separations are small, but there are many different cameras in a very small form factor.



Blender-PBRT-ISET light field spectral simulation (Andy Lin)

Andy Lin and Trishia Lian succeeded at developing the software for a quantitative simulation of the light field imager and integrated this with ISET

Henryk Blasinski is using the method with TensorFlow to estimate depth information

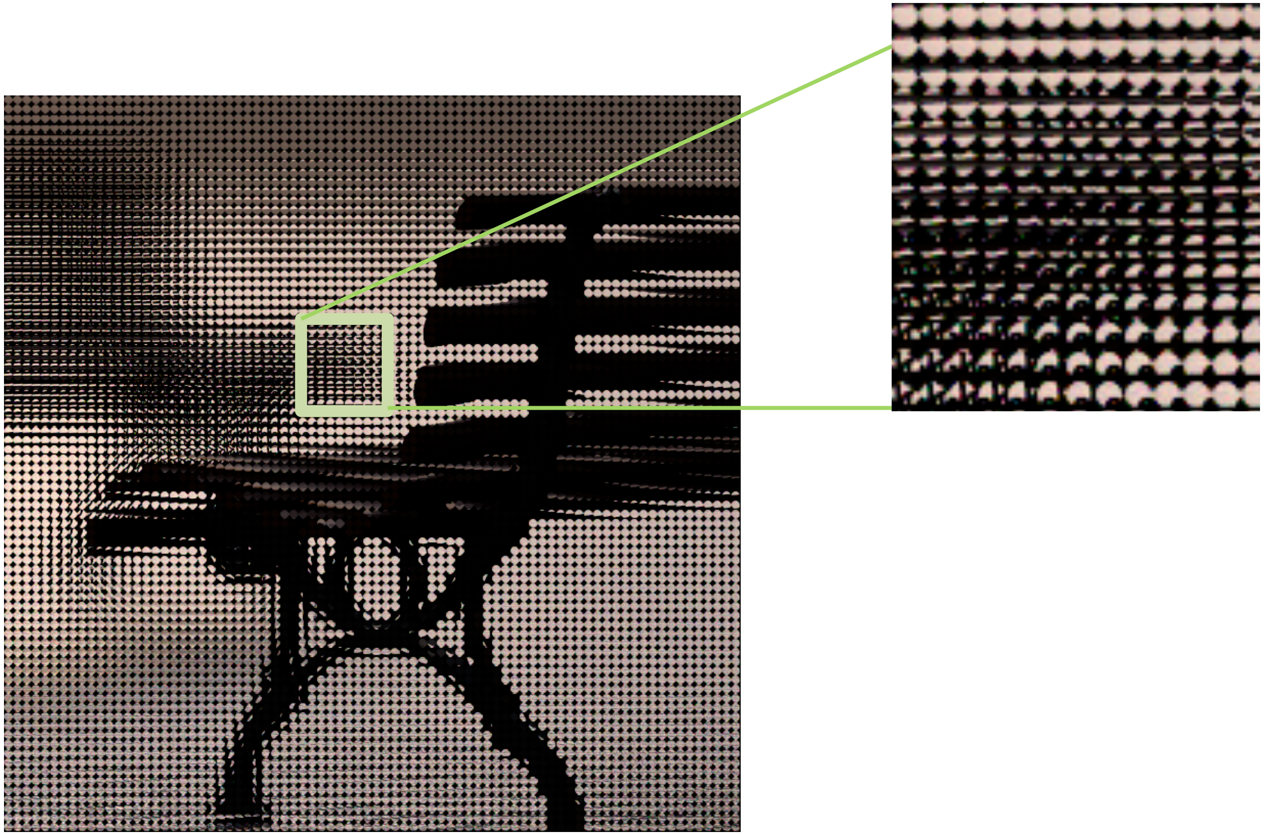


Image System Simulation for training data synthesis

Specify light field camera parameters

Choose a object (ShapeNet)

- Select a point of view, illumination, position, scale
  - Render the data from the light field camera (PBRT extension)
- Repeat (pov, …)

Repeat shapeNet (~1000)

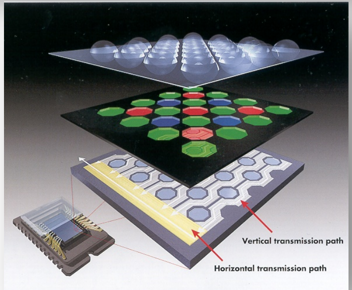
Train for depth estimation

Airplane example shapes

Light field camera rendering

Image systems simulation for intelligent vehicles

- A Jilin-Stanford database of controlled video clips for designing and evaluating image sensors for vehicle applications
  - *Requires new models*
  - *Extension to video clips*
  - *Knowledge about driving applications*
- Physical models of optics and sensors
  - *Hardware design and simulation (ISET)*
  - *Extension to new sensors (lidar,TOF)*
- Algorithms for image interpretation
  - *OpenCV*
  - *TensorFlow*
  - *Custom*



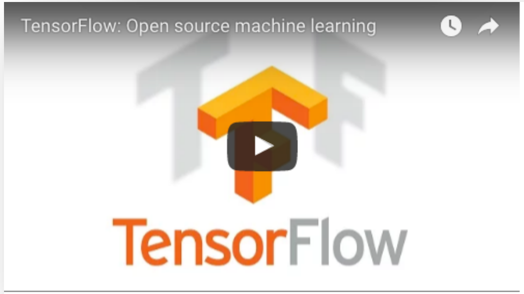
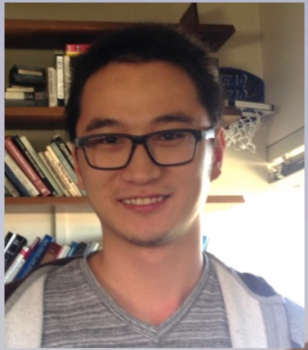


Image Systems Simulation

We thank Olympus, Qualcomm, Facebook, Google, the Brown Institute and the Simons Foundation for their support

Q. Tian

H. Jiang



J. Farrell

H. Blasinski

Trisha Lian