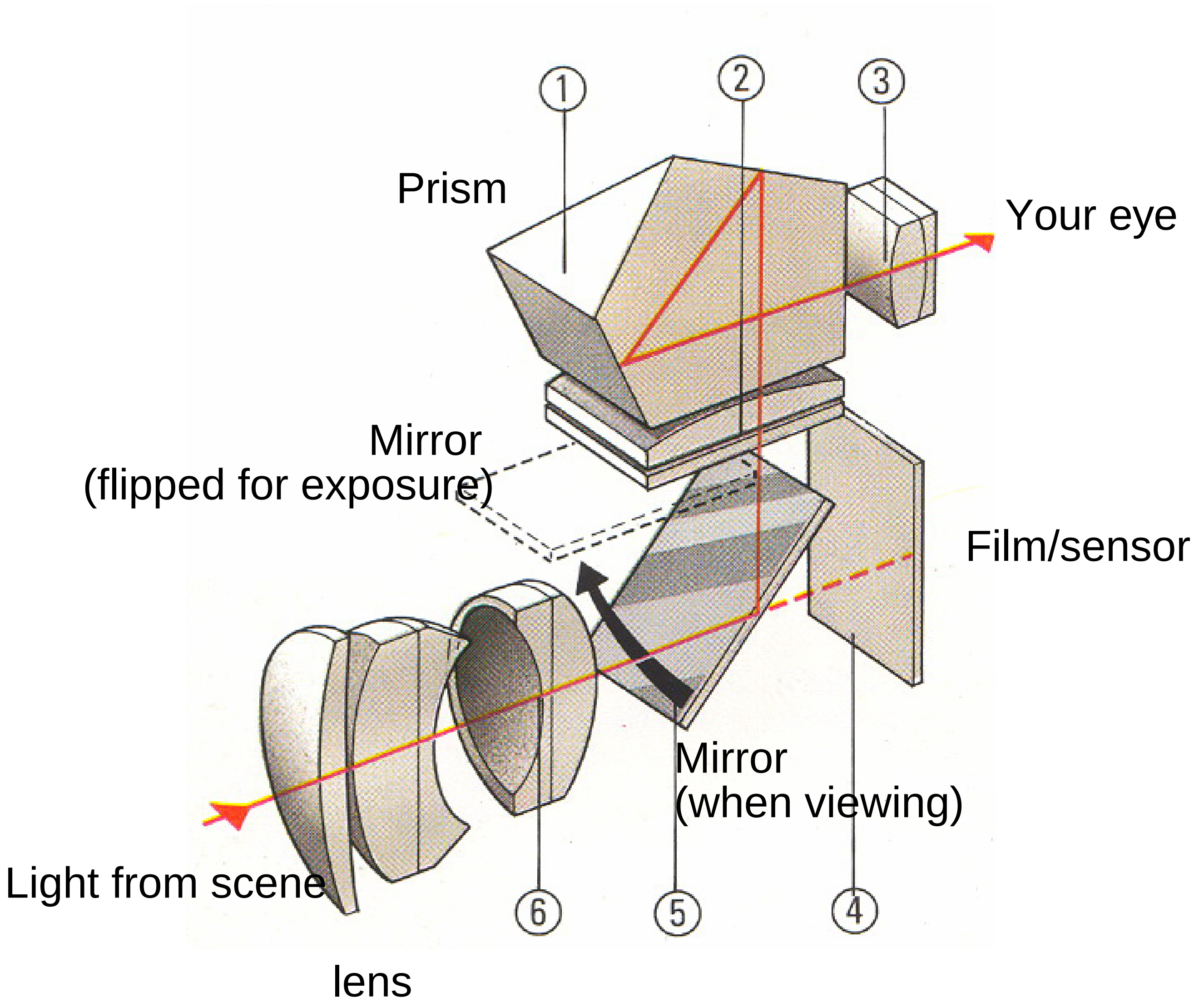
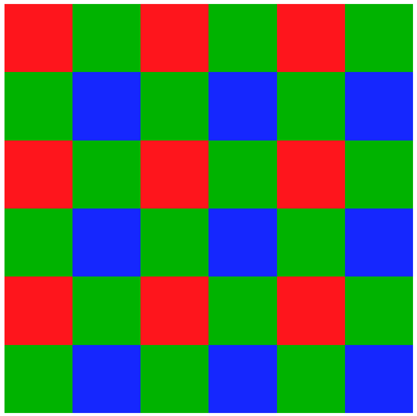
Last Lecture



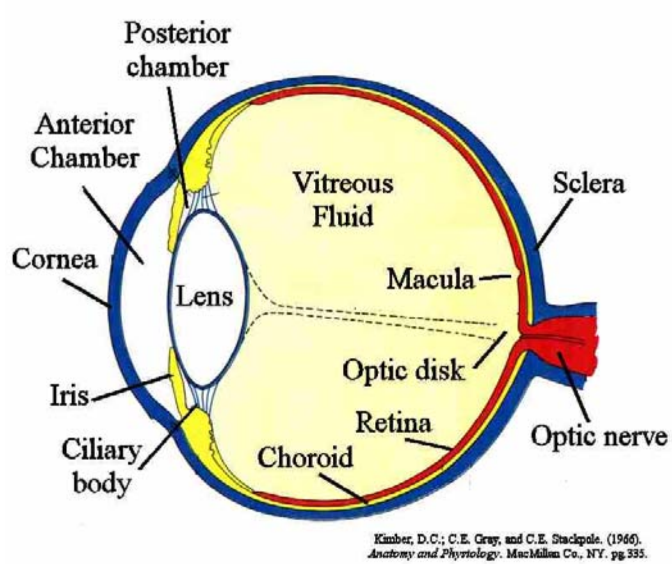
- Focal Length
- F-stop
- Depth of Field
- Color Capture



Bayer pattern

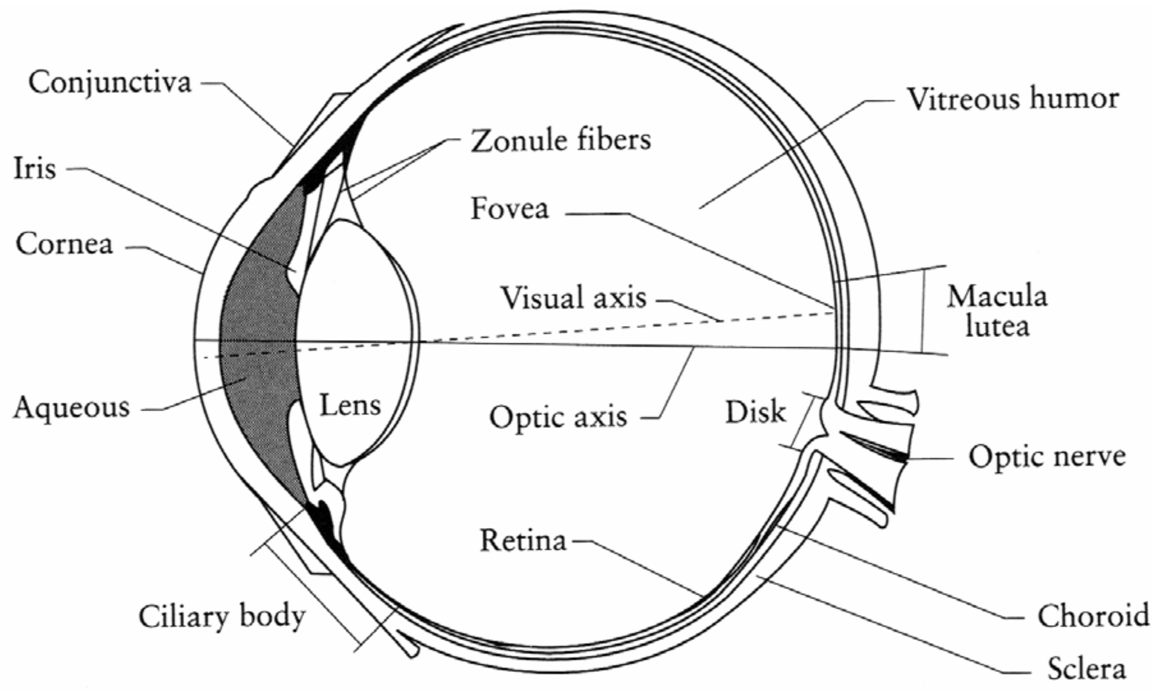
YungYu Chuang's slide

Today



photomatix.com

The Eye



- The human eye is a camera!
  - **Iris** - colored annulus with radial muscles
  - **Pupil** - the hole (aperture) whose size is controlled by the iris
  - What's the "film"?
    - photoreceptor cells (rods and cones) in the **retina**

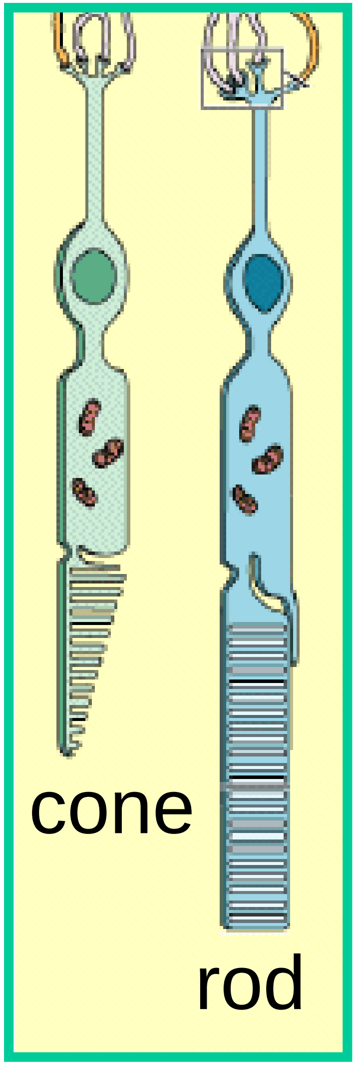
Two types of light-sensitive receptors

Rods

rod-shaped
highly sensitive
operate at night
gray-scale vision

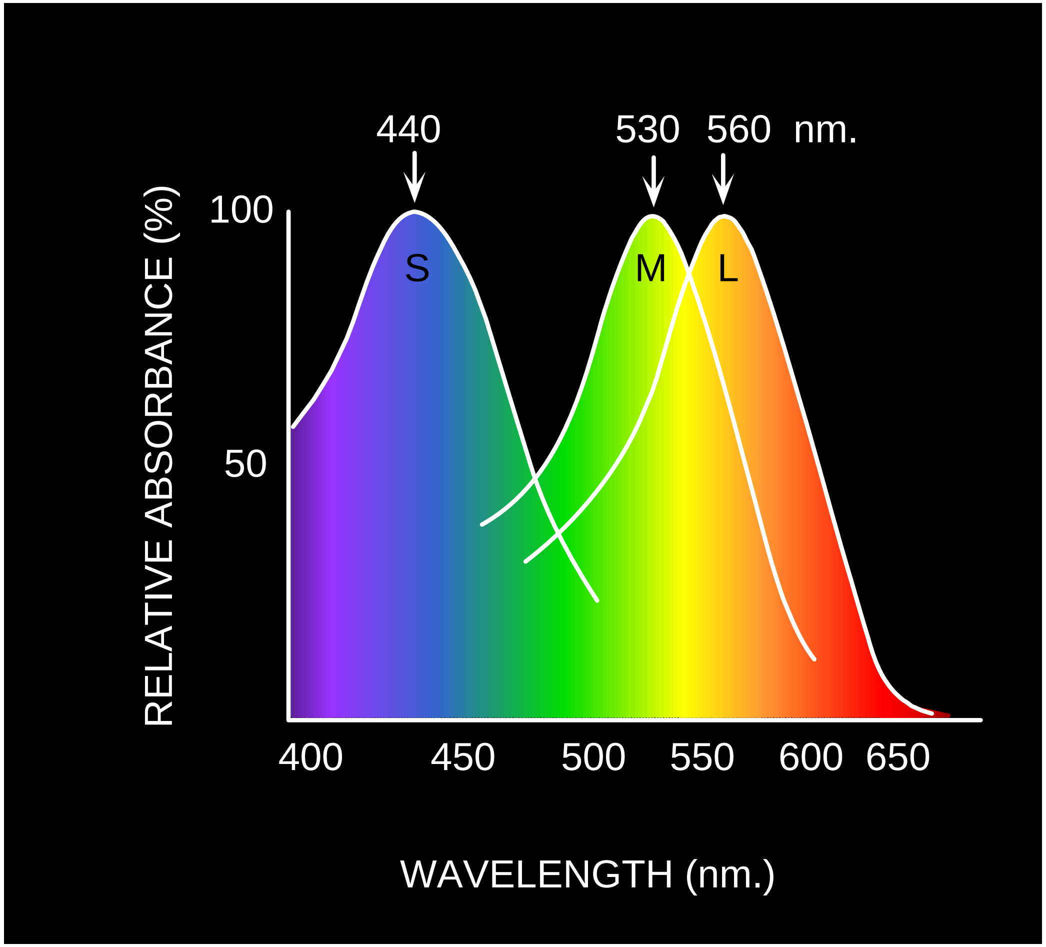
Cones

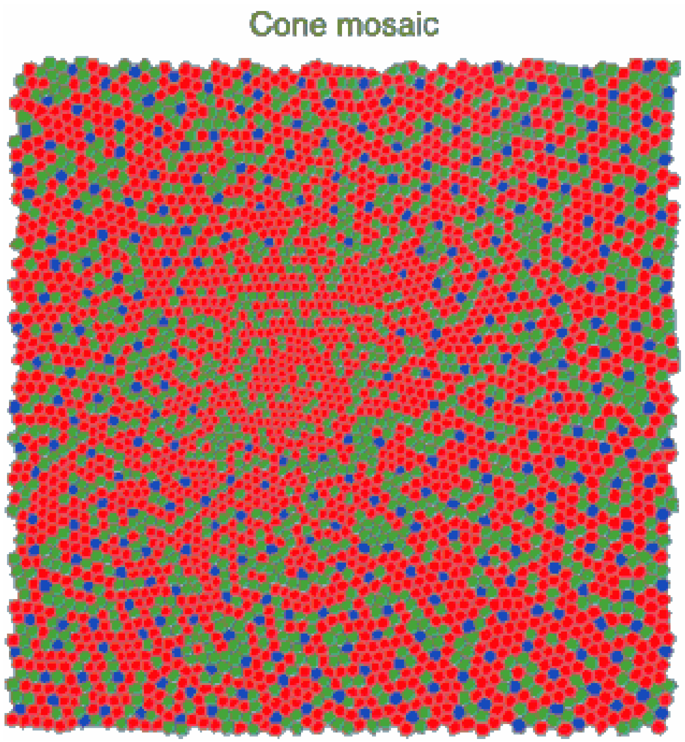
cone-shaped
less sensitive
operate in high light
color vision



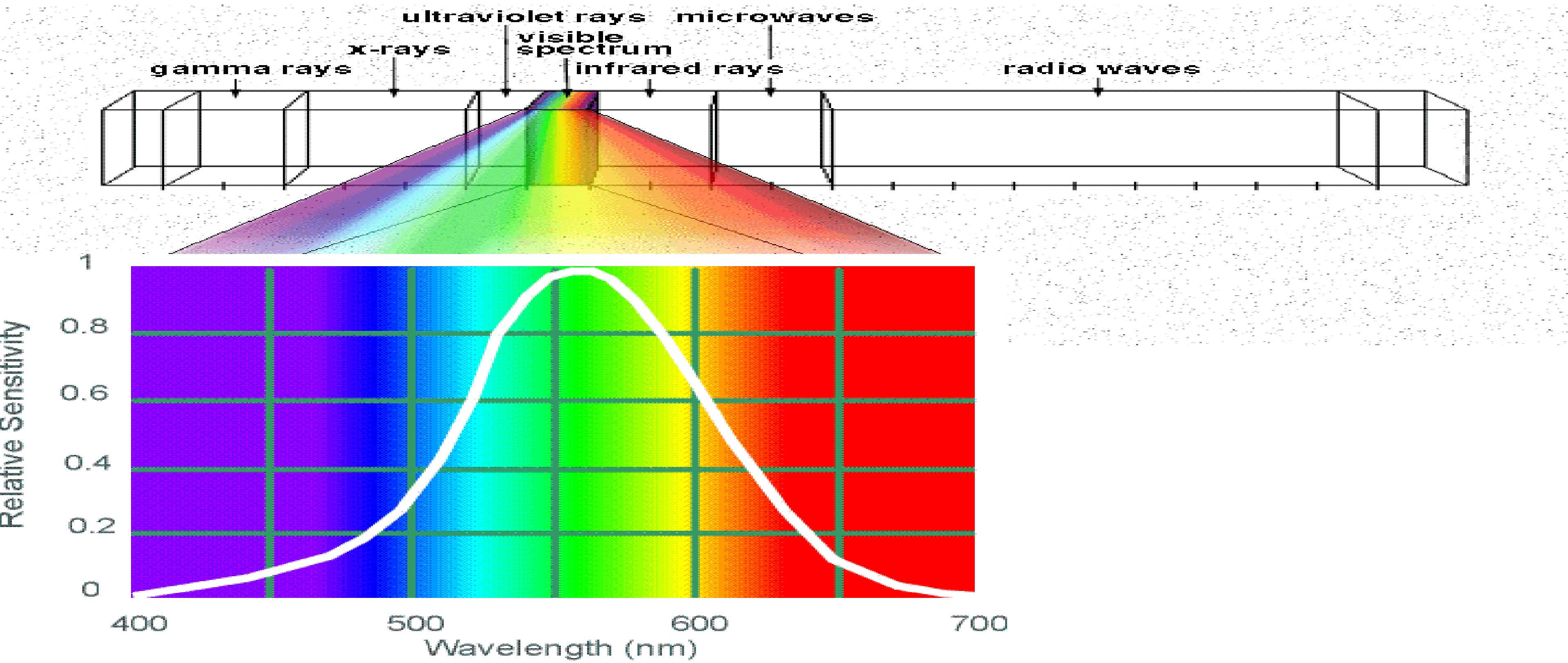
Physiology of Color Vision

Three kinds of cones:





Electromagnetic Spectrum



Human Luminance Sensitivity Function

http://www.yorku.ca/eye/photopik.htm

References

- http://www.howstuffworks.com/digital-camera.htm
- http://electronics.howstuffworks.com/autofocus.htm
- Ramanath, Snyder, Bilbro, and Sander. Demosaicking Methods for Bayer Color Arrays, Journal of Electronic Imaging, 11(3), pp306-315.
- Rajeev Ramanath, Wesley E. Snyder, Youngjun Yoo, Mark S. Drew, Color Image Processing Pipeline in Digital Still Cameras, IEEE Signal Processing Magazine Special Issue on Color Image Processing, vol. 22, no. 1, pp. 34-43, 2005.
- http://www.worldatwar.org/photos/whitebalance/index.mhtml
- http://www.100fps.com/

The World in an Eye

Ko Nishino Shree K. Nayar

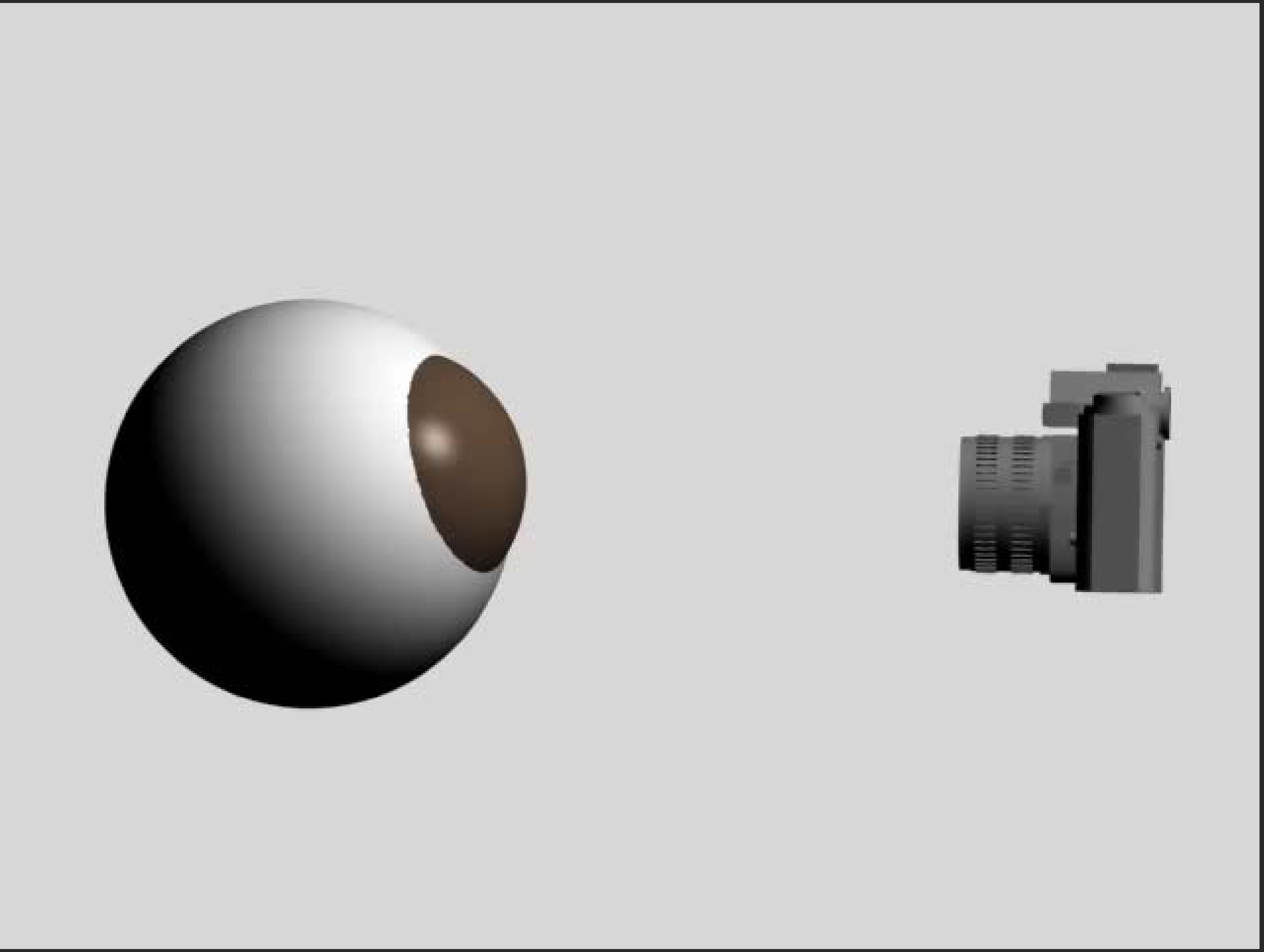
Columbia University

http://www1.cs.columbia.edu/CAVE/projects/world_eye/world_eye.php

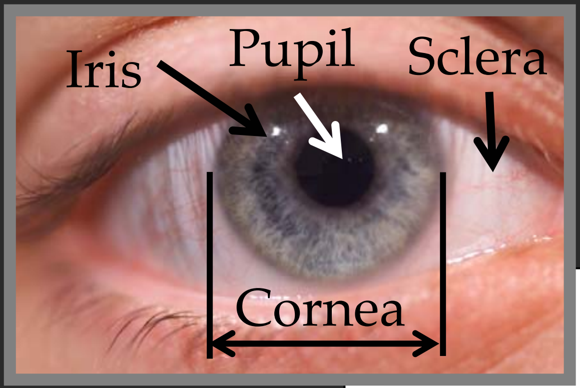
IEEE CVPR Conference
June 2004, Washington DC, USA

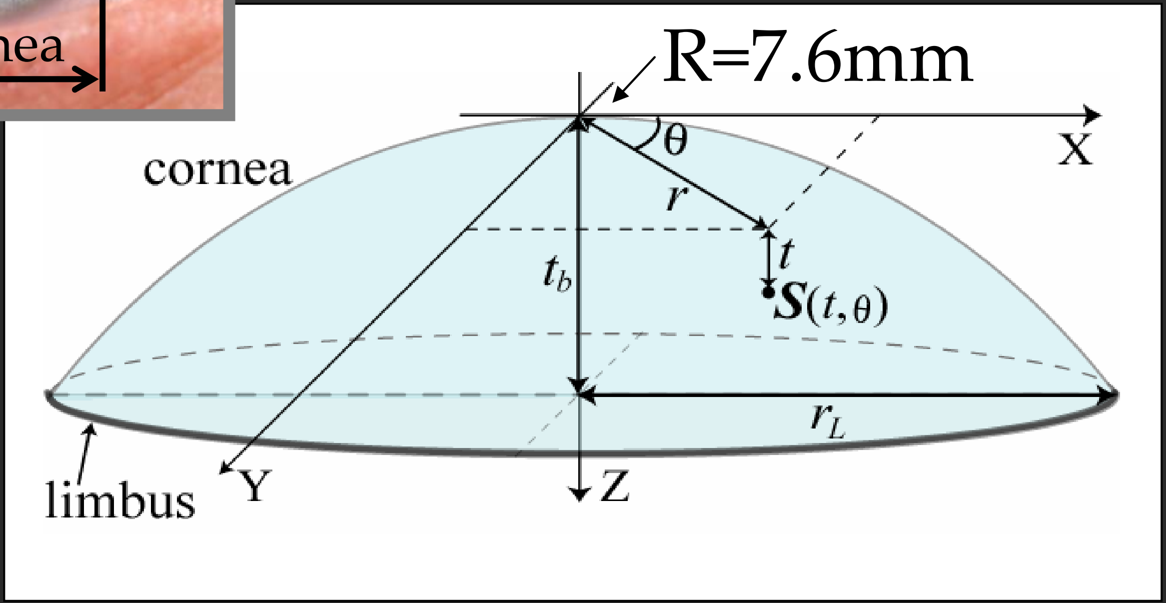
Supported by NSF

Corneal Imaging System



Geometric Model of the Cornea

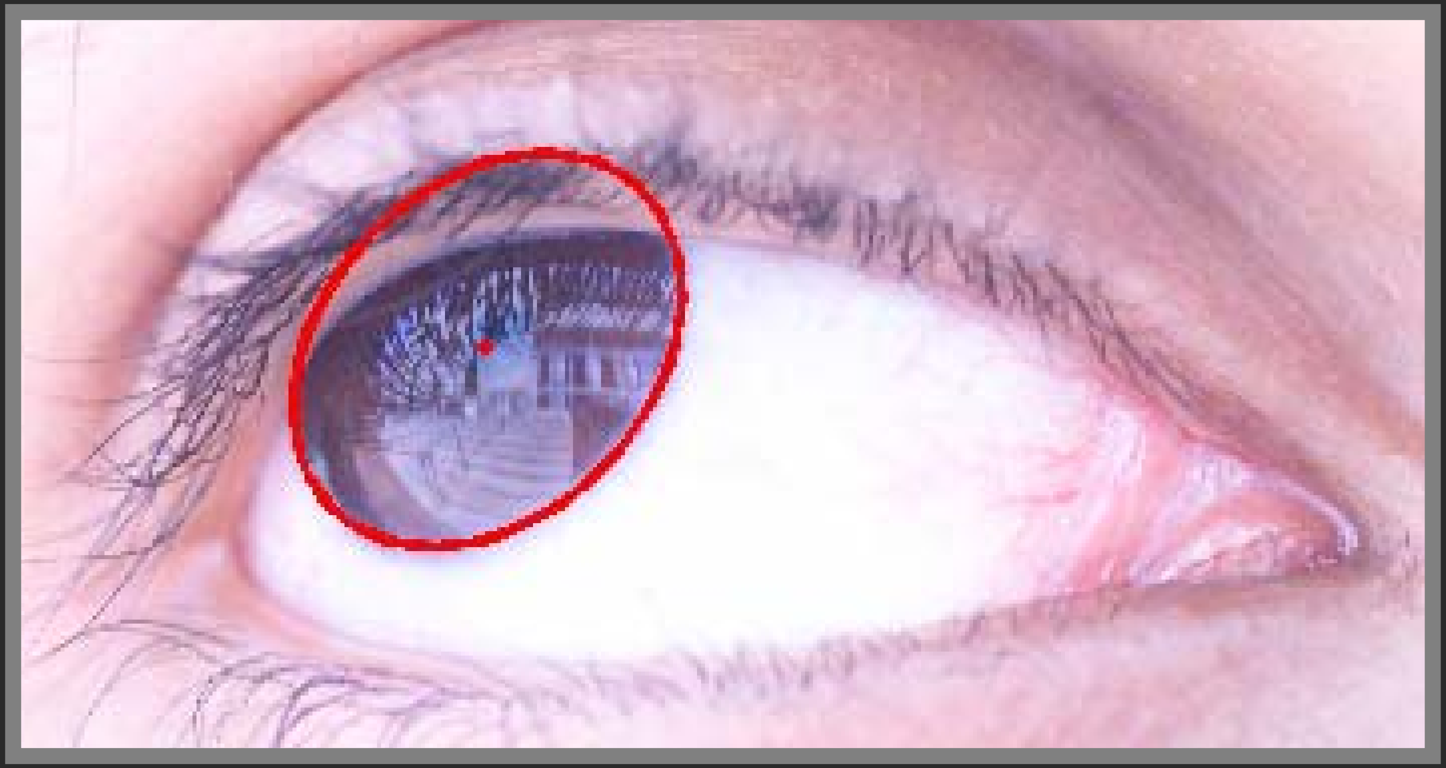




$$t_b = 2.18\text{mm} \qquad r_L = 5.5\text{mm}$$

eccentricity = 0.5

Finding the Limbus



limbus parameters e : radii (r_x, r_y) center (c_x, c_y) tilt θ

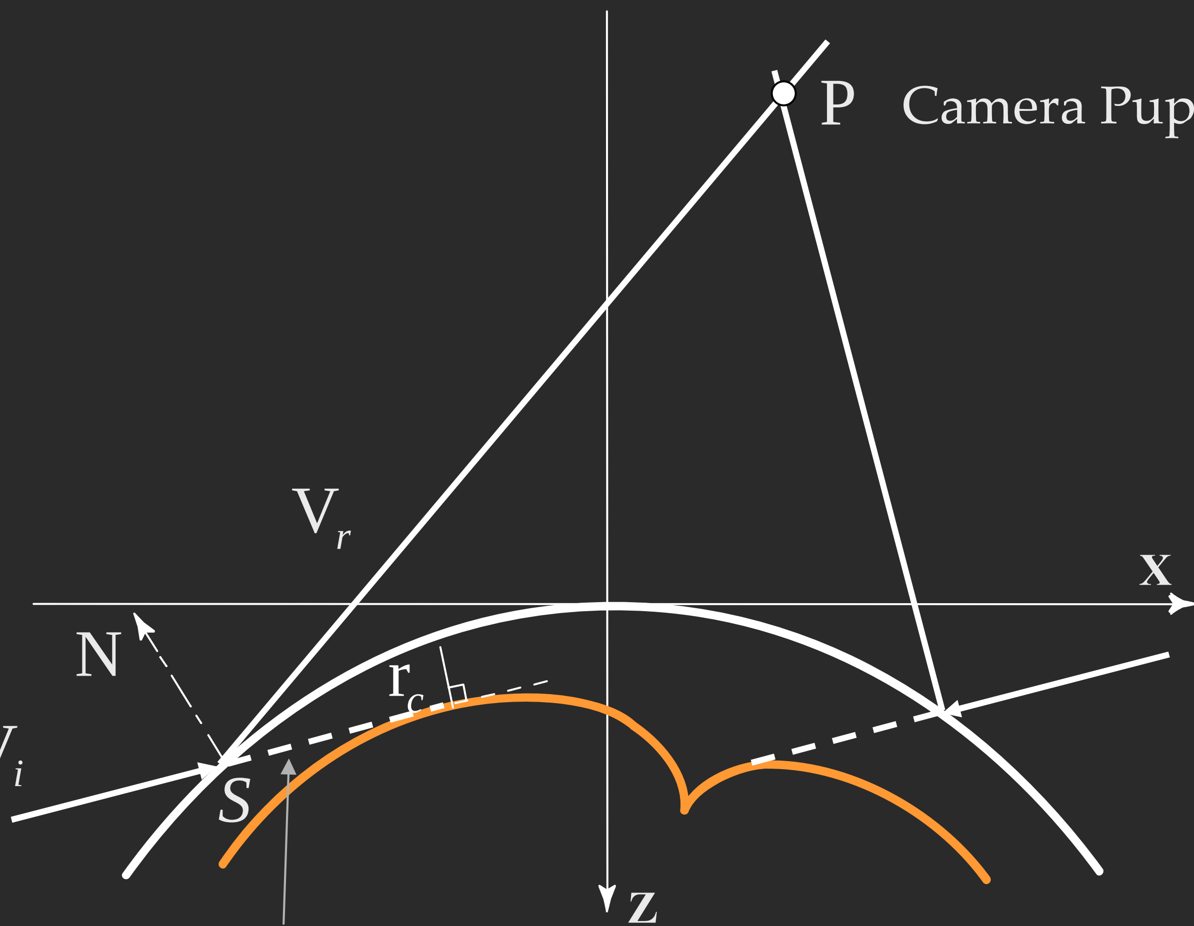
$$\max_e \left| \underbrace{g_\sigma(r_x)}_{\text{Gaussian}} * \frac{\partial}{\partial r_x} \oint_e \underbrace{I(x,y)ds}_{\text{intensity value}} + g_\sigma(r_y) * \frac{\partial}{\partial r_y} \oint_e I(x,y)ds \right|$$

Self-calibration:

3D Coordinates, 3D Orientation

How does the World Appear in an Eye?

Imaging Characteristics: Viewpoint Locus

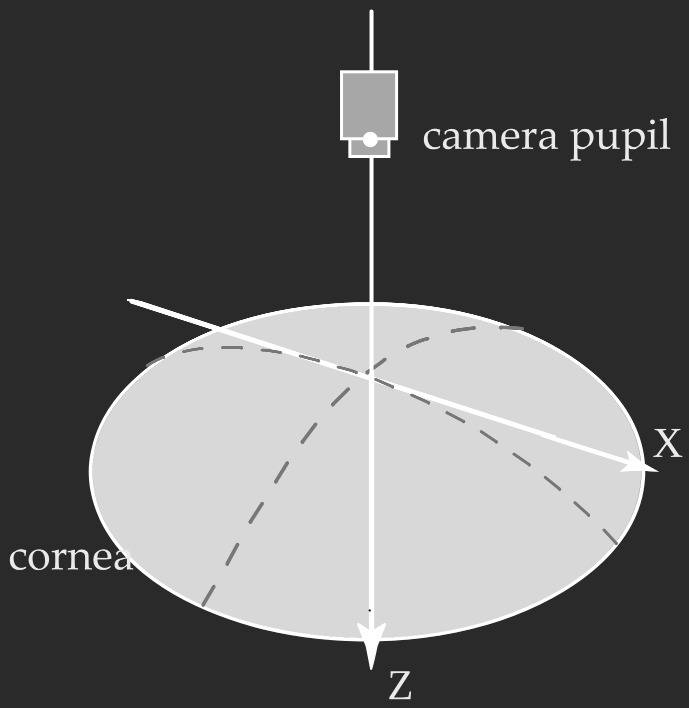


$$\mathbf{V}(t,\theta,r) = \mathbf{S}(t,\theta) + r\mathbf{V}_i(t,\theta)$$

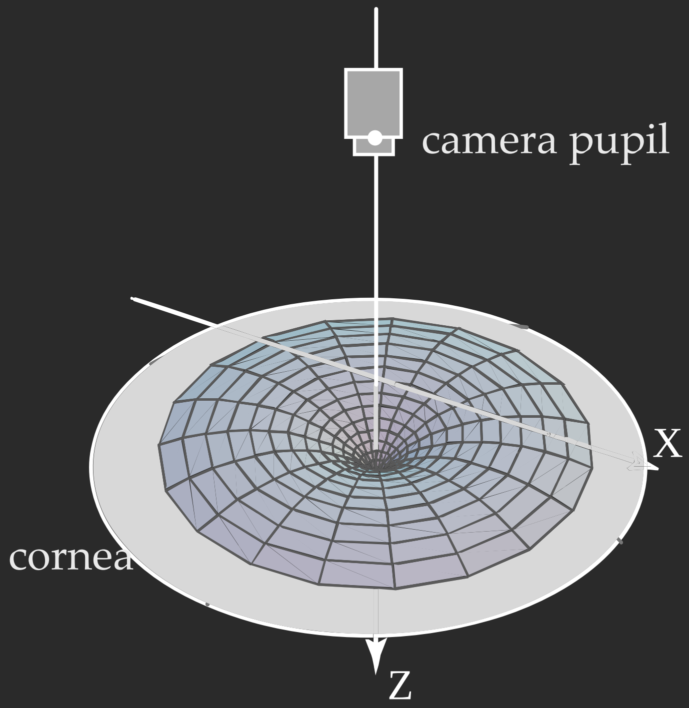
$$\det J\left(\mathbf{V}(t,\theta,r_c)\right) = 0$$

[Burkhard and Shealy 73; Swaminathan et al. 01]

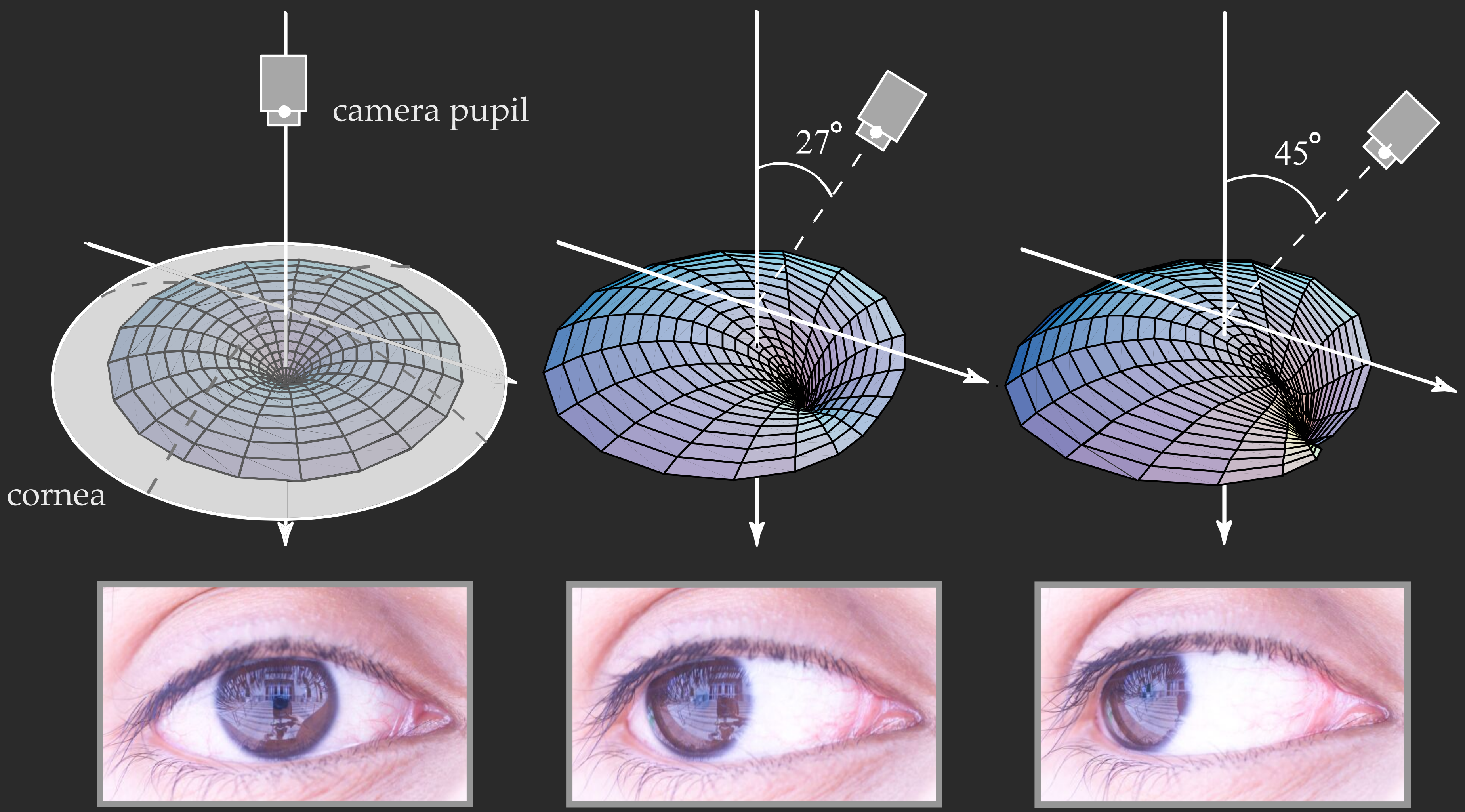
Viewpoint Loci



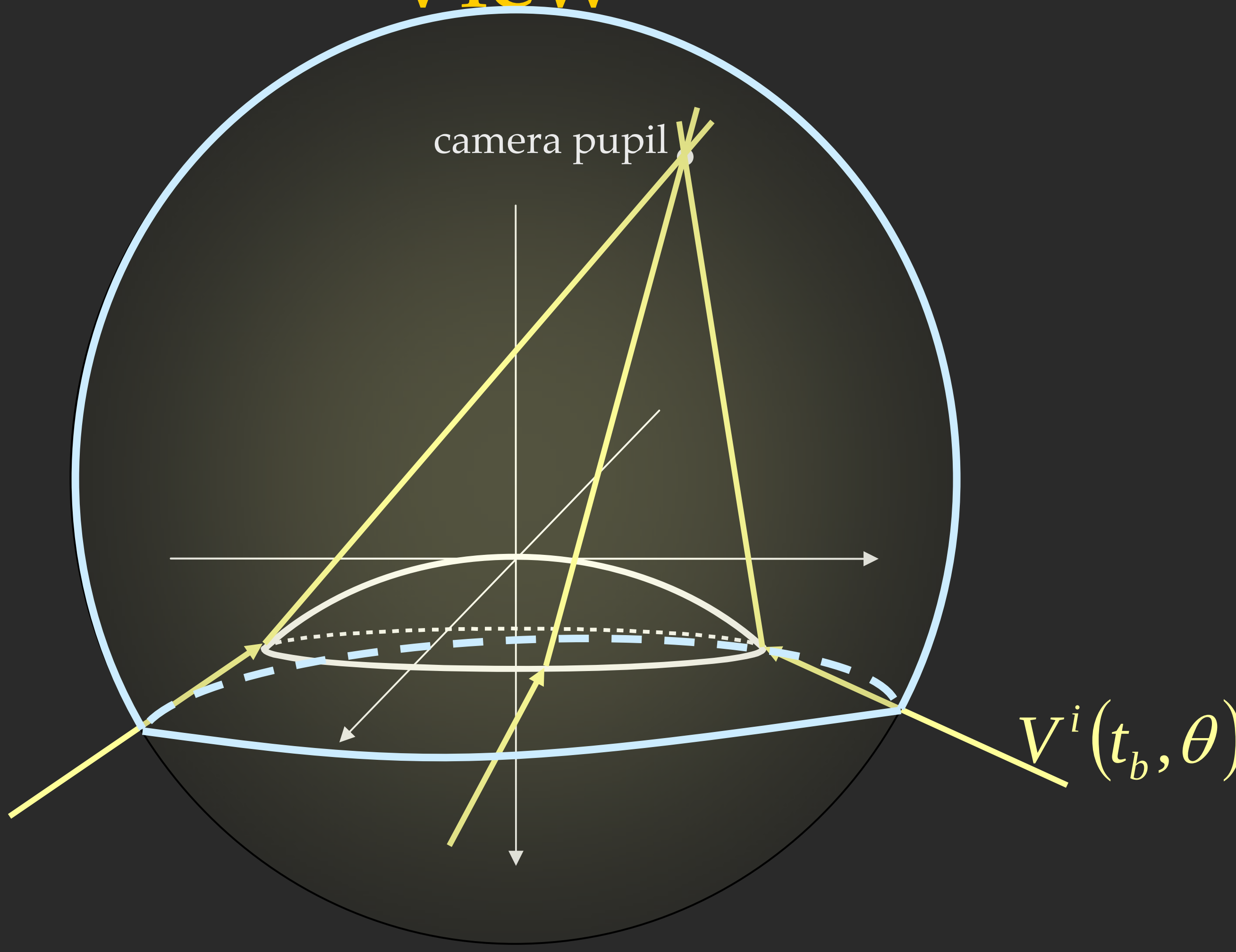
Viewpoint Loci



Viewpoint Loci



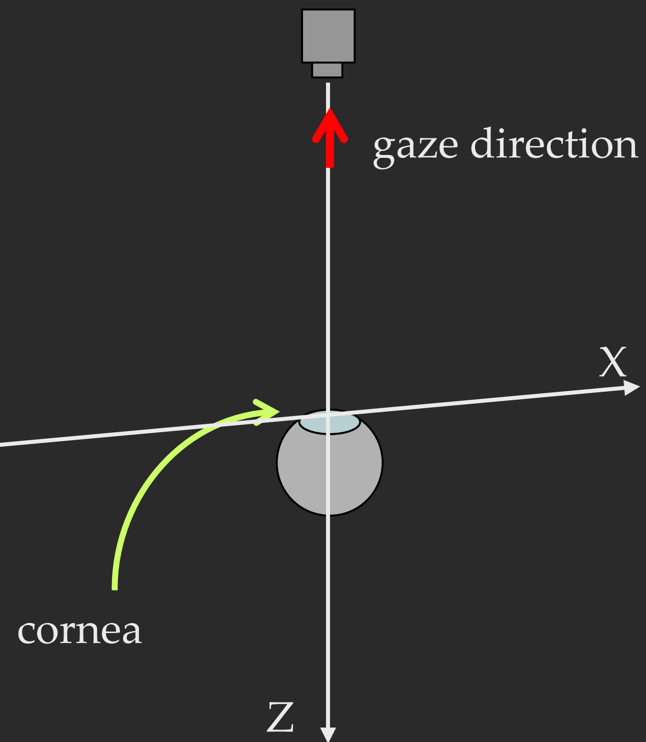
Imaging Characteristics: Field of View



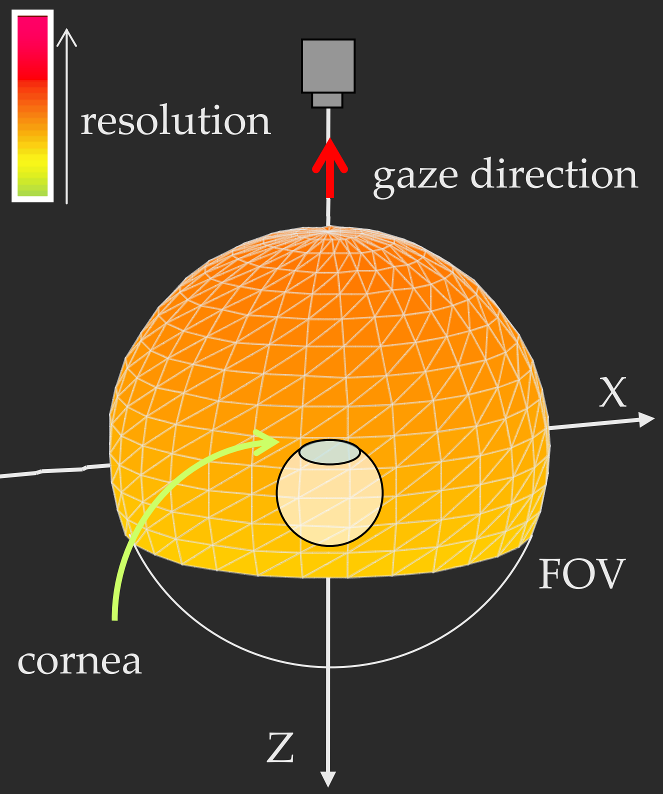
$$\text{FOV} = \int_0^{2\pi} \left(-\underline{V_z^i(t_b,\theta)} + 1 \right) d\theta$$

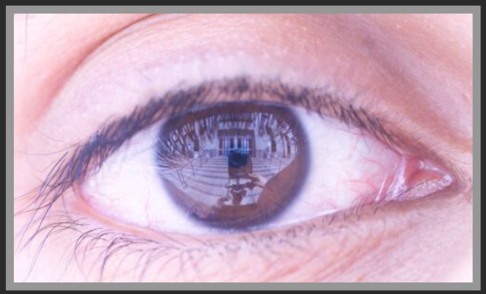
closed loop of limbus
incident light rays

Resolution and Field of View

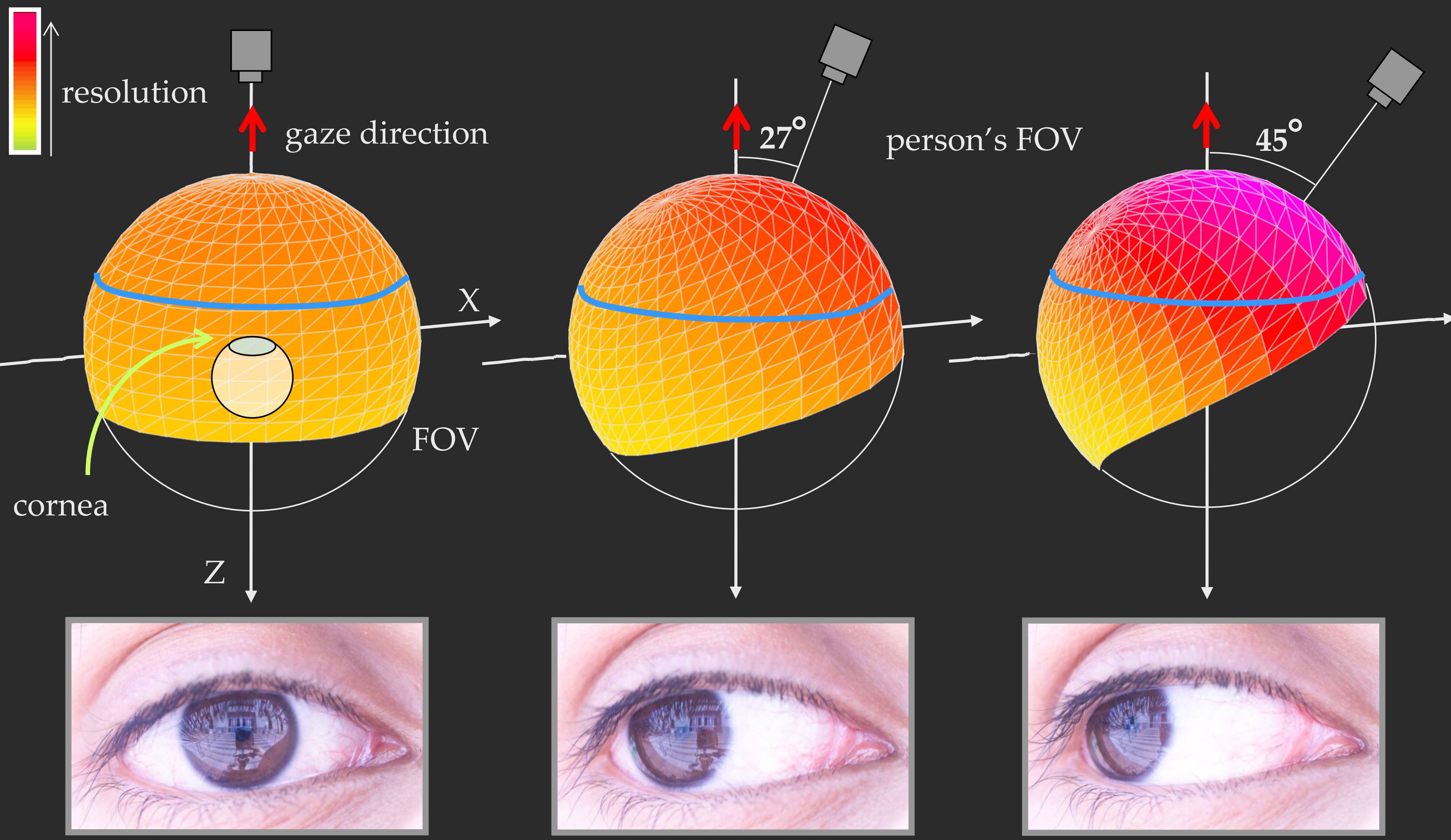


Resolution and Field of View

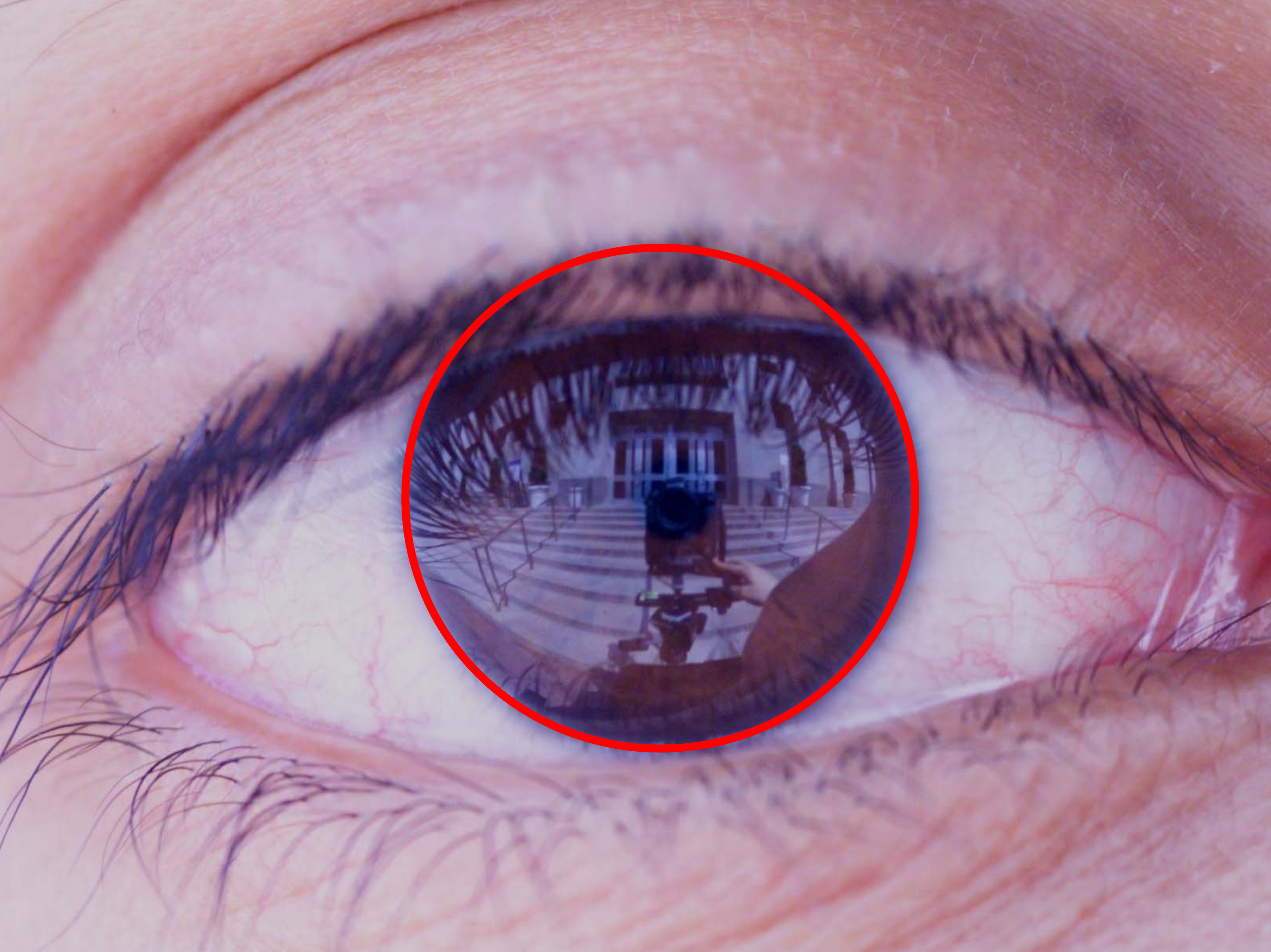




Resolution and Field of View



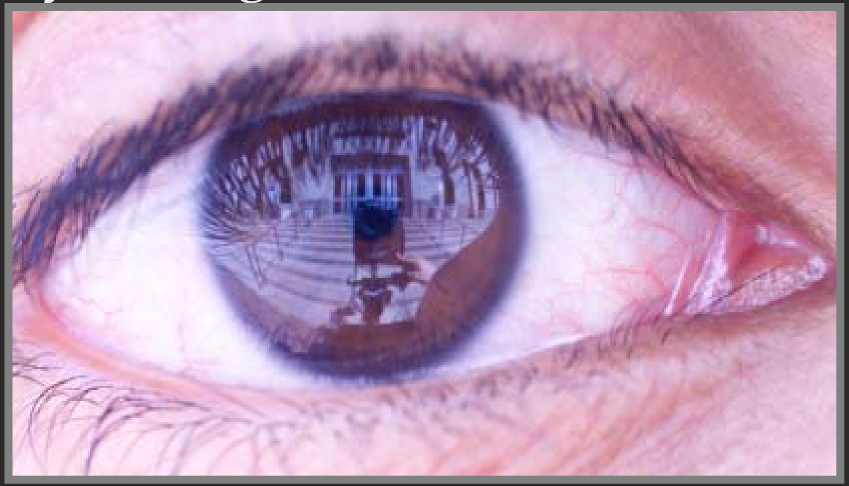
What does the Eye Reveal?



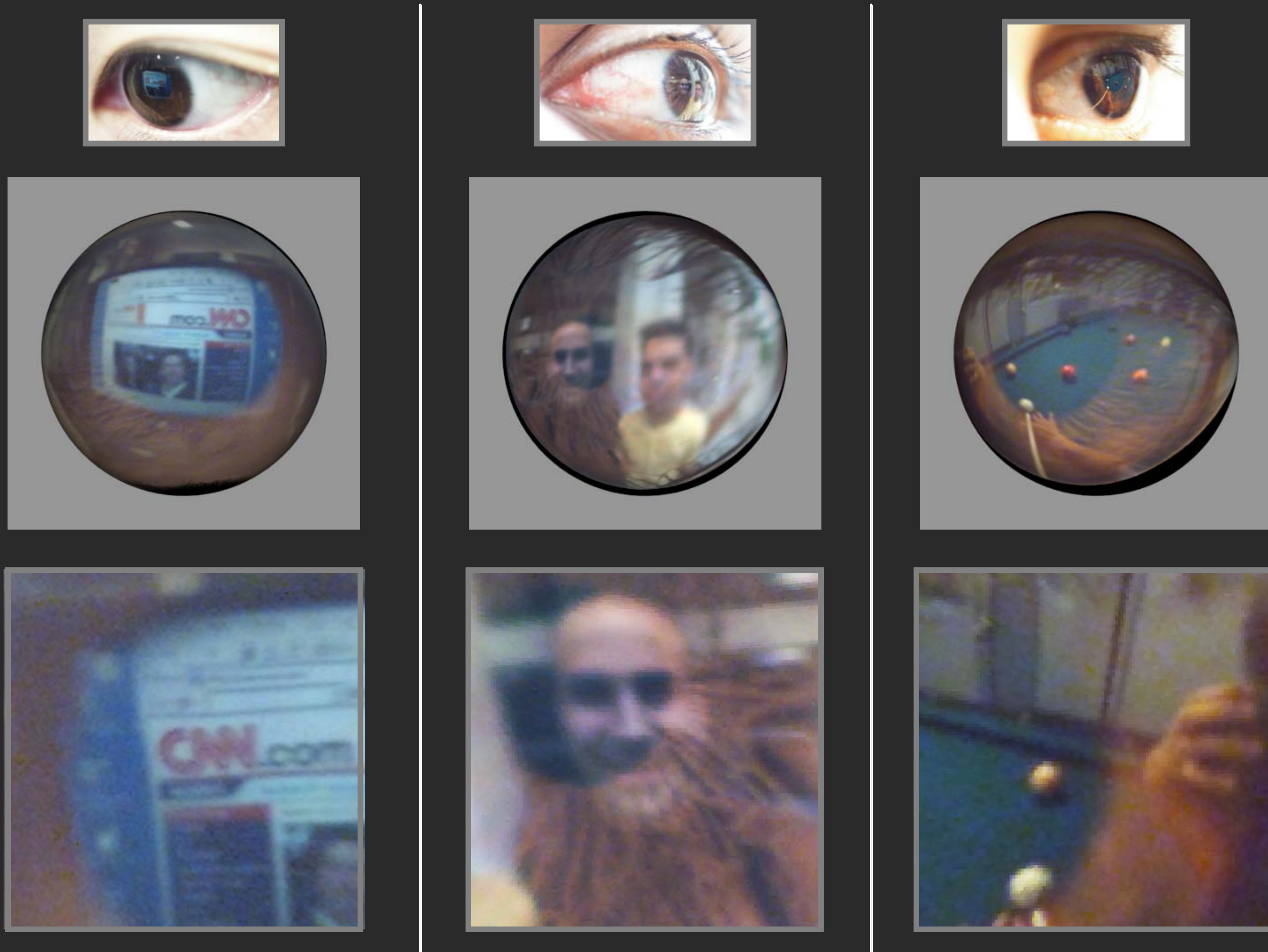
Environment Map from an Eye

What Exactly You are Looking At

Eye Image:

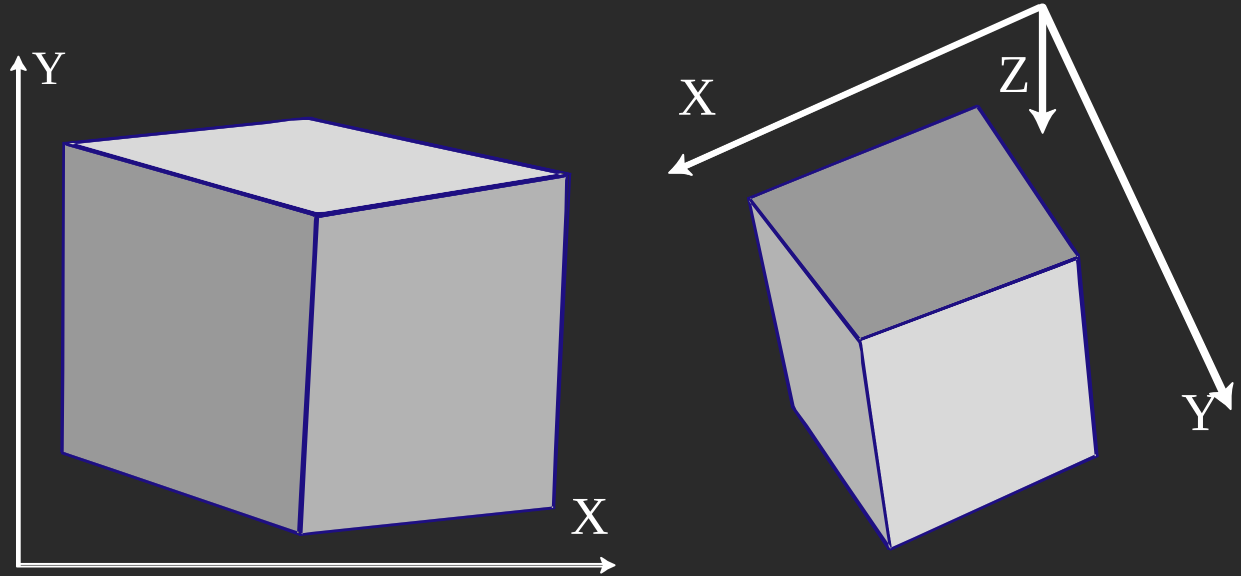


Computed Retinal Image:


CNN.com

Watching a Bus

From Two Eyes in an Image …



Reconstructed Structure (frontal and side view)

Eyes Reveal …

– Where the person is

– What the person is looking at

– The structure of objects

Implications

Human Affect Studies: Social Networks

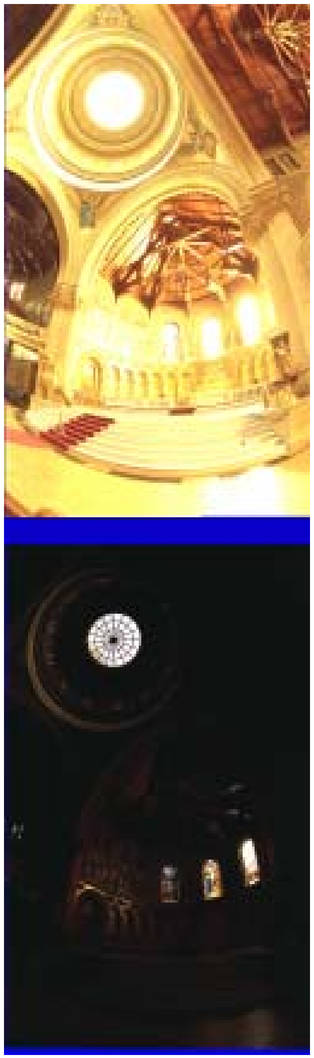
Security: Human Localization

Advanced Interfaces: Robots, Computers

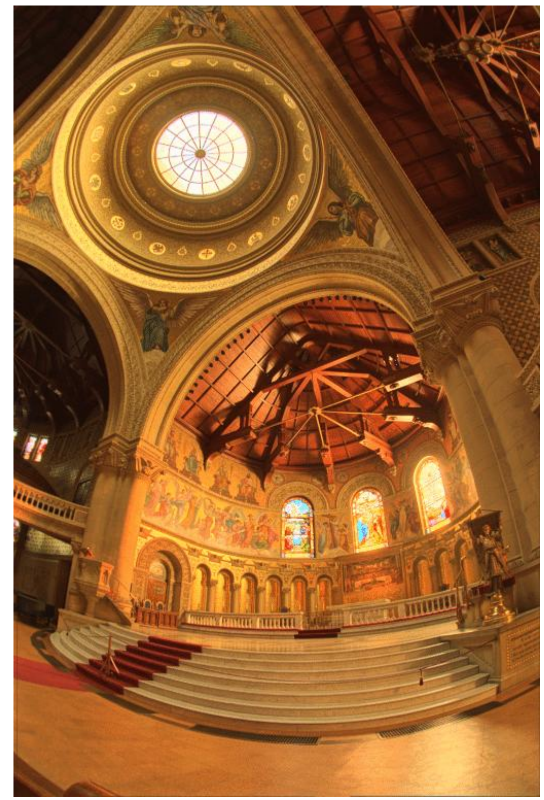
Computer Graphics: Relighting [SIGGRAPH 04]

Questions?

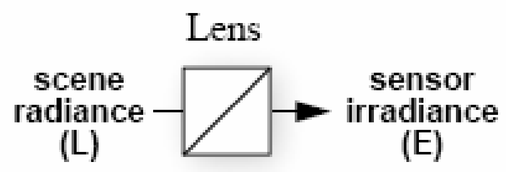
What do we see?



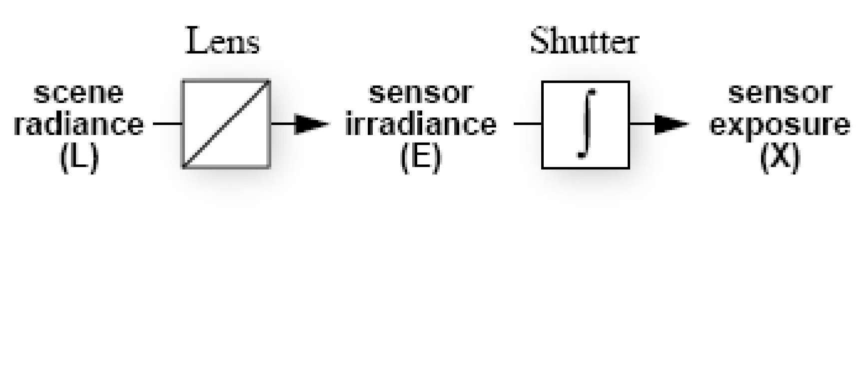
Vs.



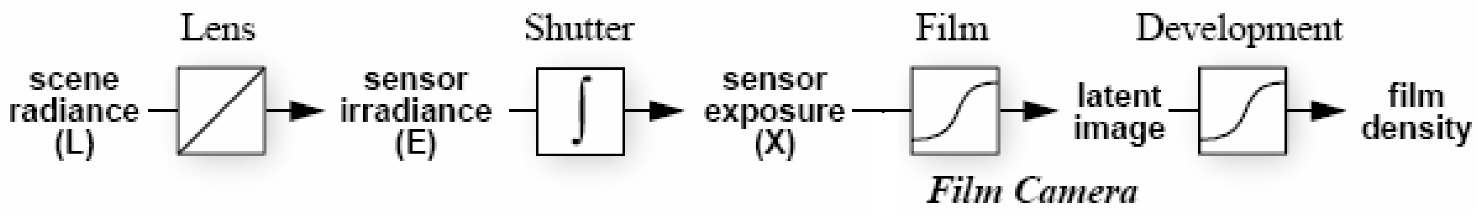
Camera pipeline



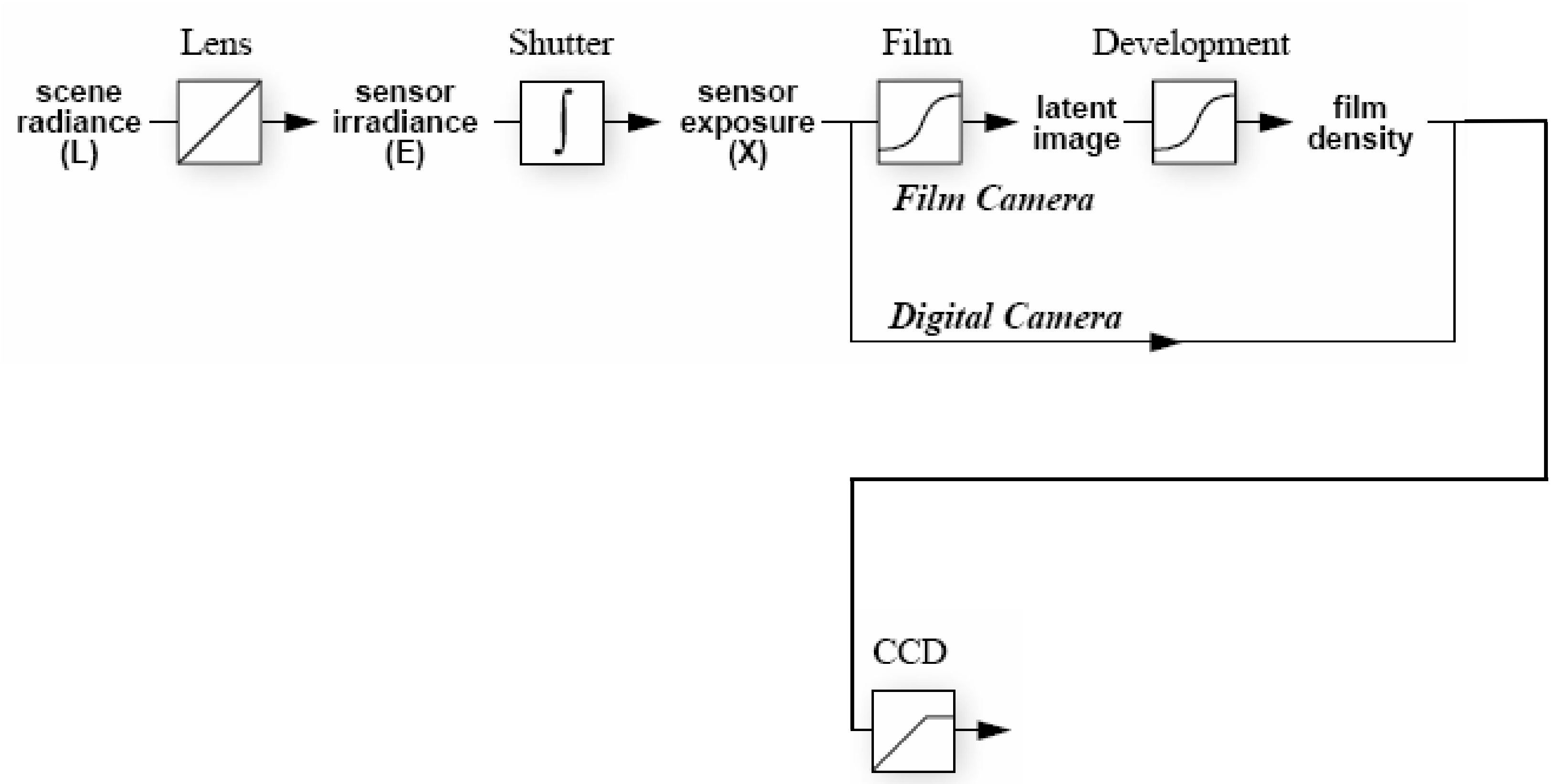
Camera pipeline



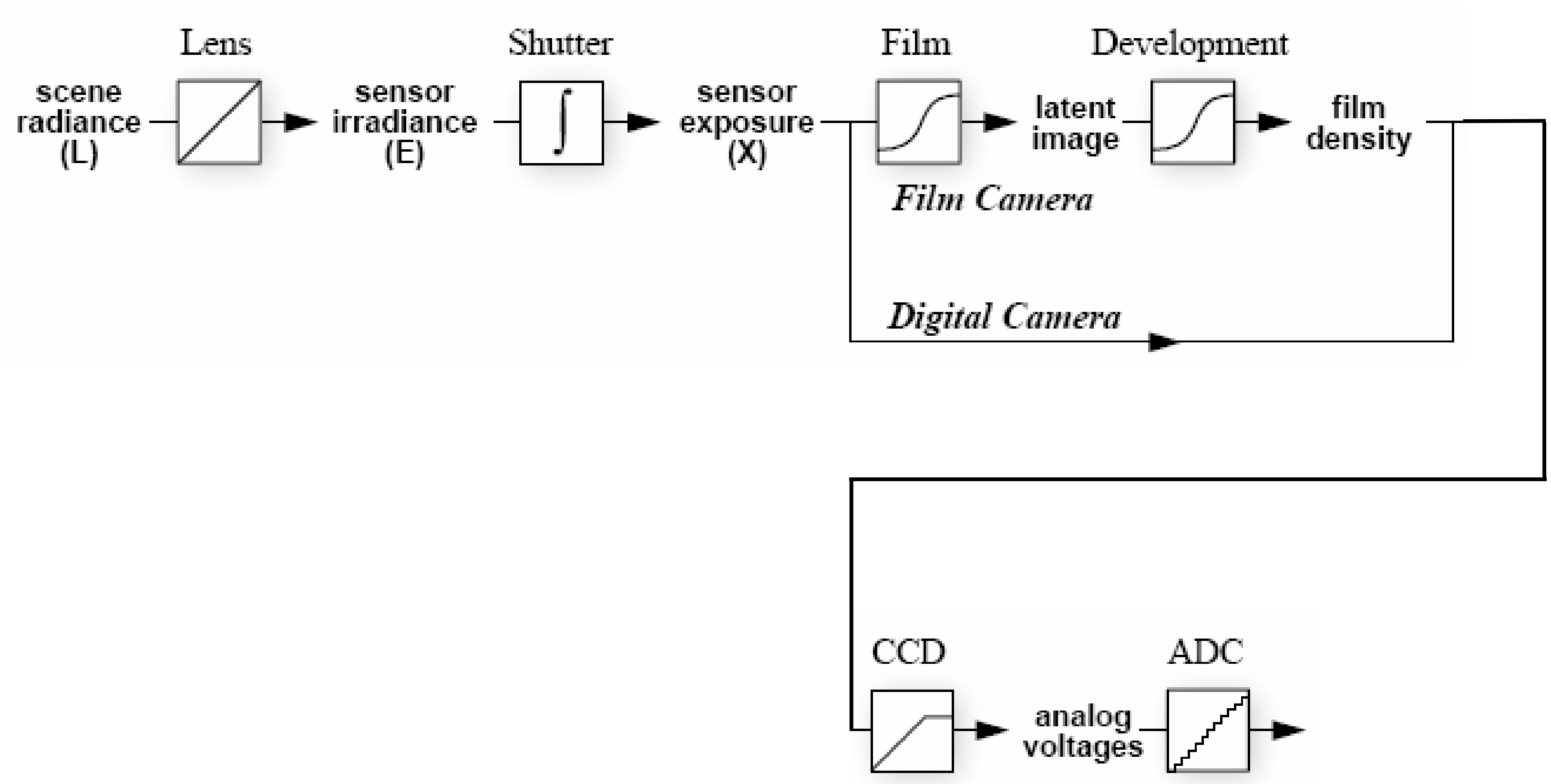
Camera pipeline



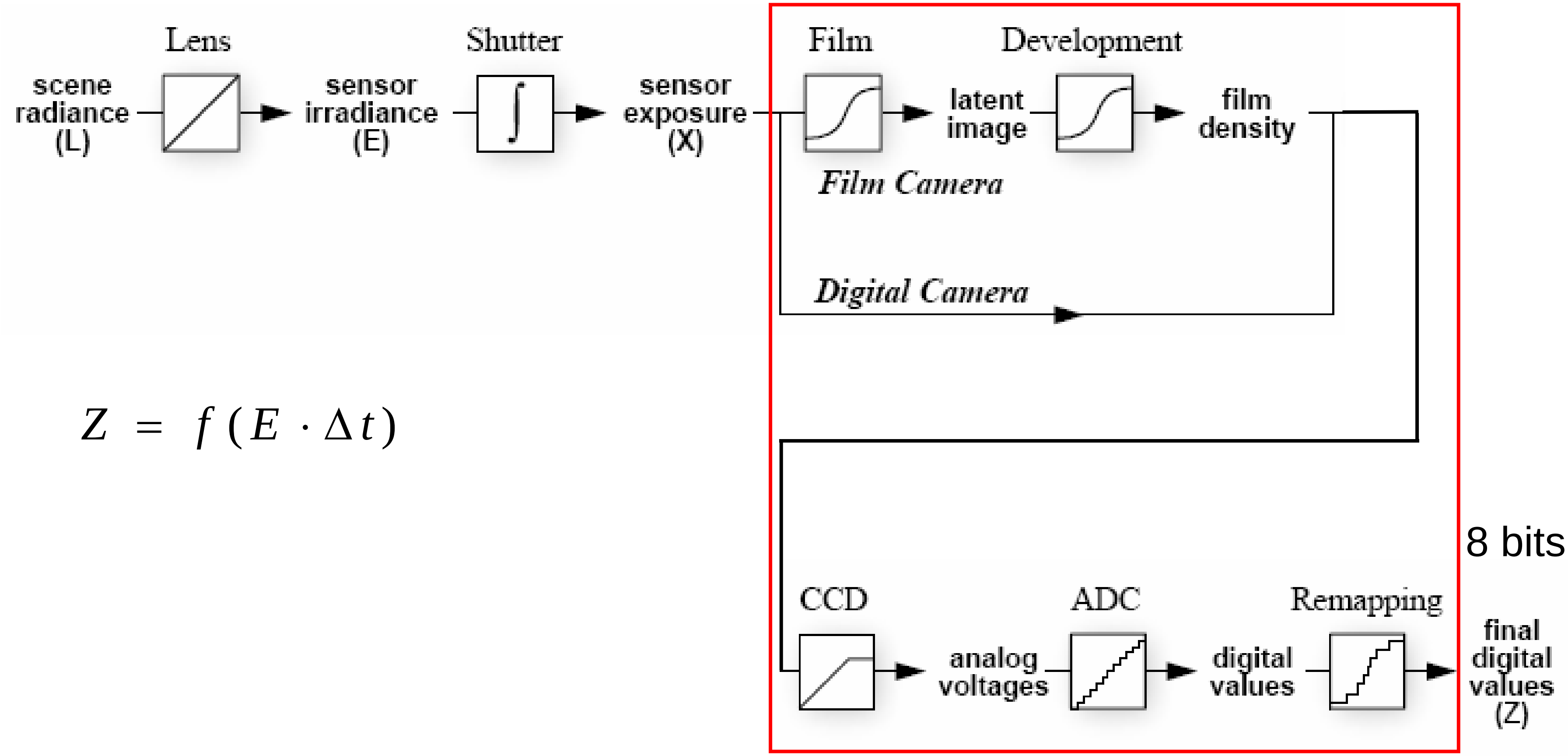
Camera pipeline



Camera pipeline



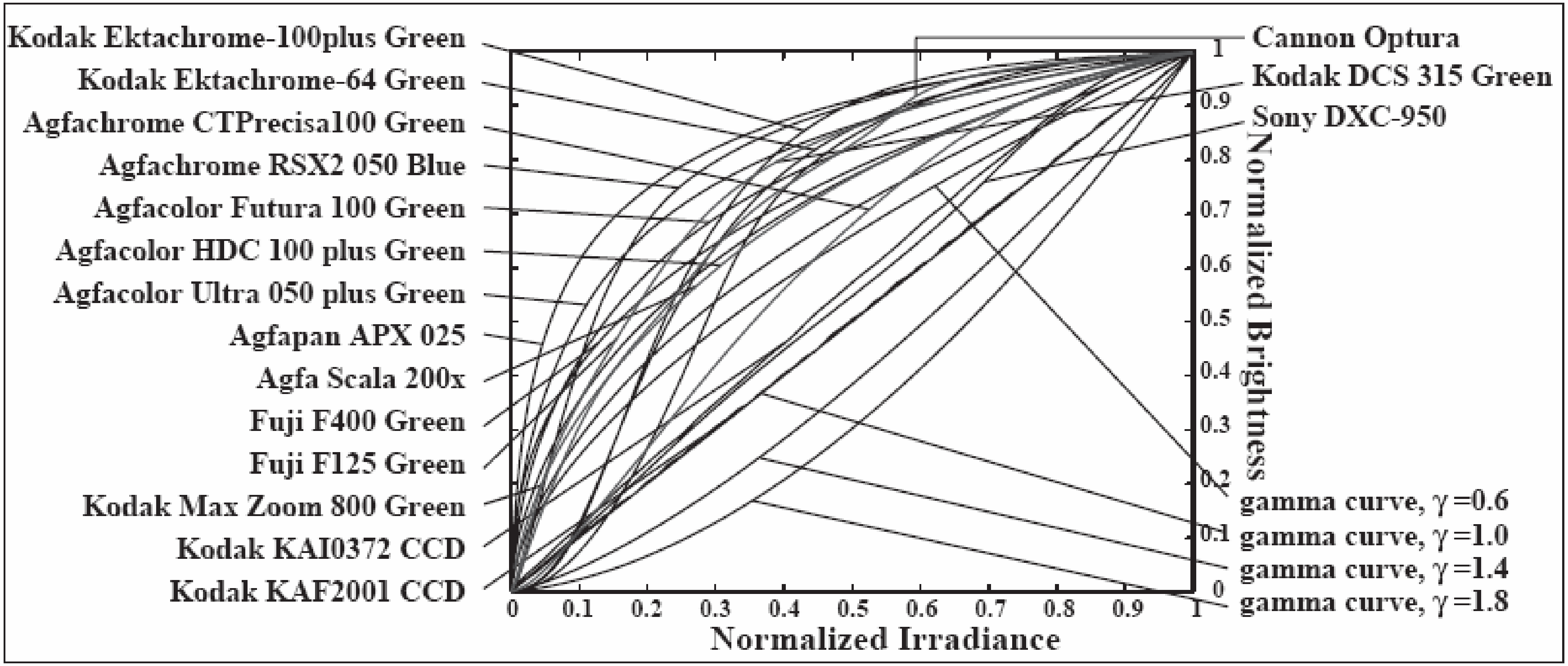
Camera pipeline



$$Z = f(E \cdot \Delta t)$$

Real-world response functions

In general, the response function is not provided by camera makers who consider it part of their proprietary product differentiation. In addition, they are beyond the standard gamma curves.



Camera is not a photometer

- Limited dynamic range

  $\Rightarrow$ Perhaps use multiple exposures?

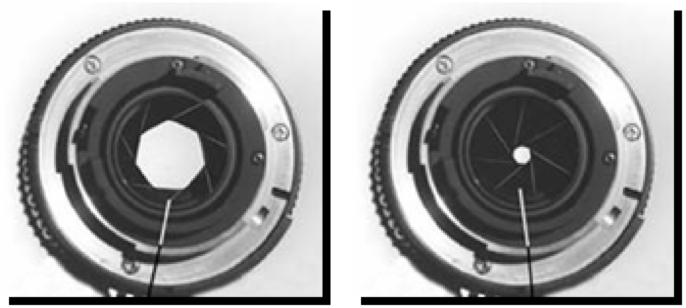
- Unknown, nonlinear response

  $\Rightarrow$ Not possible to convert pixel values to radiance

- Solution:
  - Recover response curve from multiple exposures, then reconstruct the ***radiance map***

Varying exposure

- Ways to change exposure
  - Shutter speed
  - Aperture
  - Neutral density filters



Shutter speed

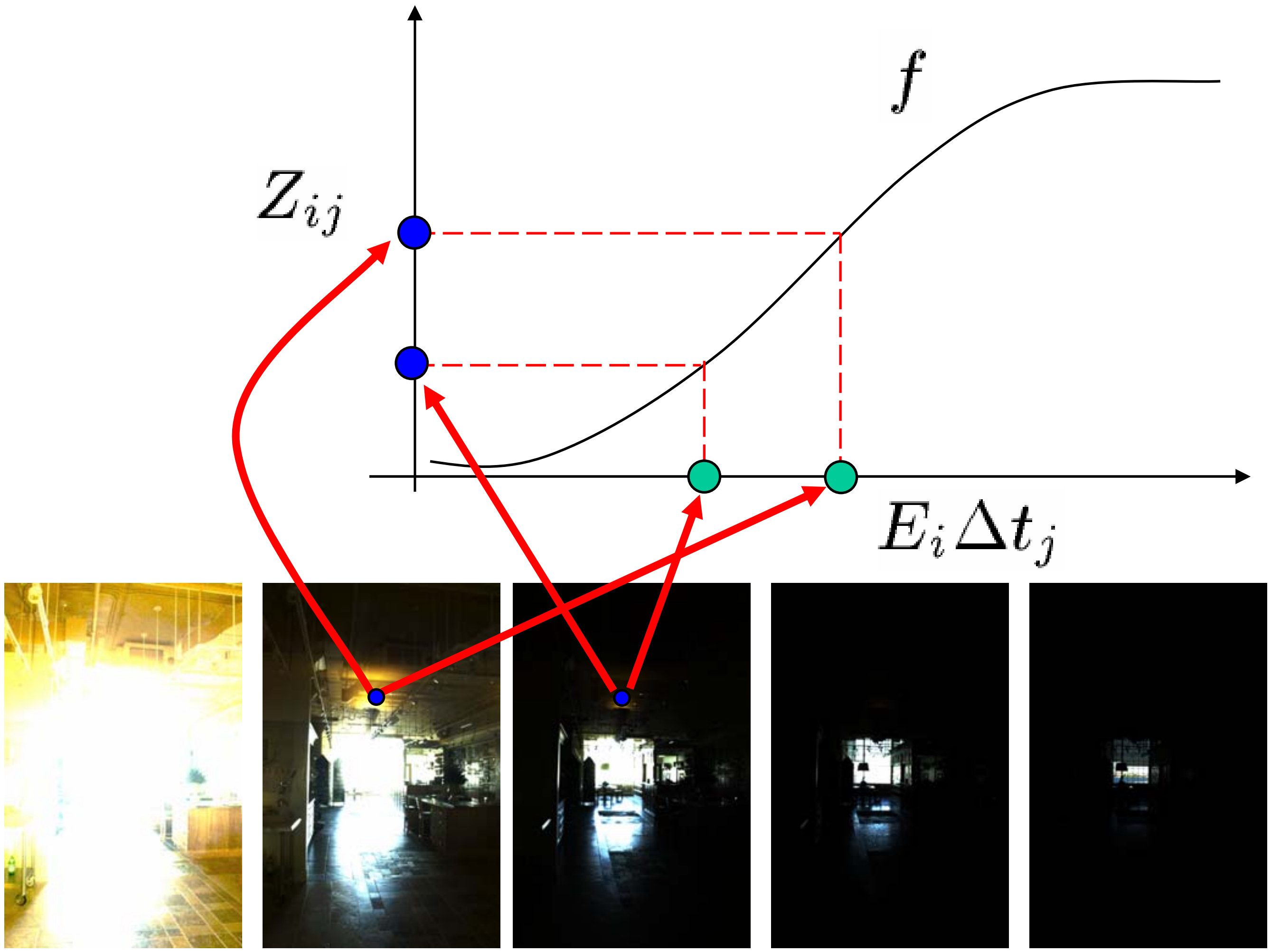
- Note: shutter times usually obey a power series – each “stop” is a factor of 2
- ¼, 1/8, 1/15, 1/30, 1/60, 1/125, 1/250, 1/500, 1/1000 sec

Usually really is:

¼, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, 1/512, 1/1024 sec

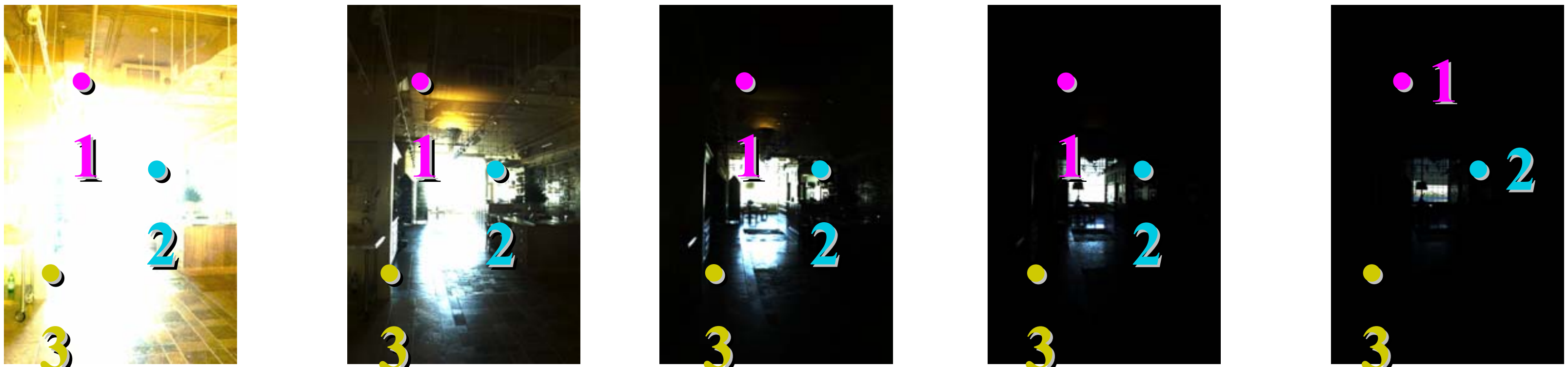
HDRI capturing from multiple exposures

- We want to obtain the response curve



HDRI capturing from multiple exposures

Image series



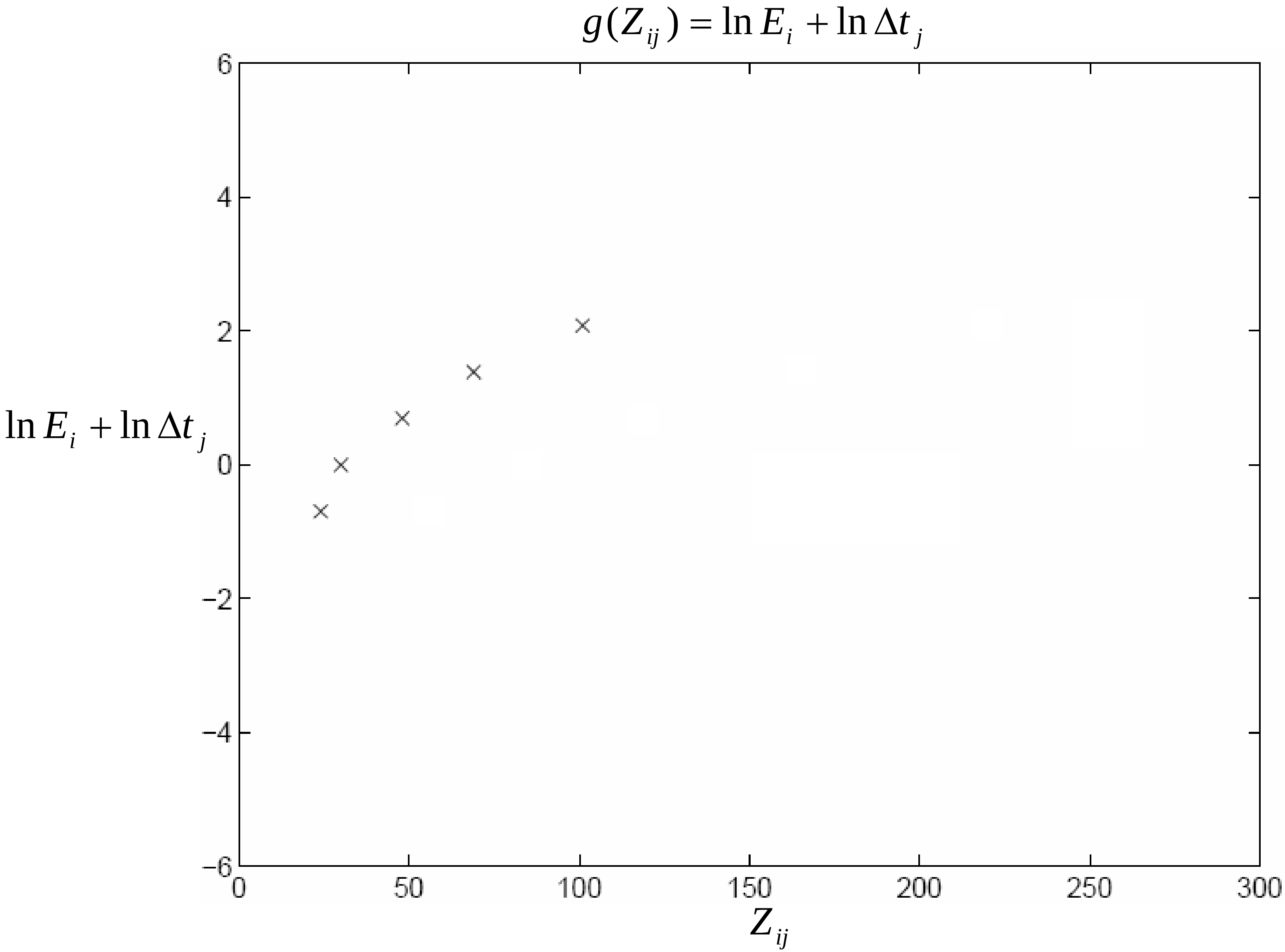
Δt = 2 sec	**Δt = 1 sec**	**Δt = 1/2 sec**	**Δt = 1/4 sec**	**Δt = 1/8 sec**

$$Z_{ij} = f(E_i \Delta t_j)$$

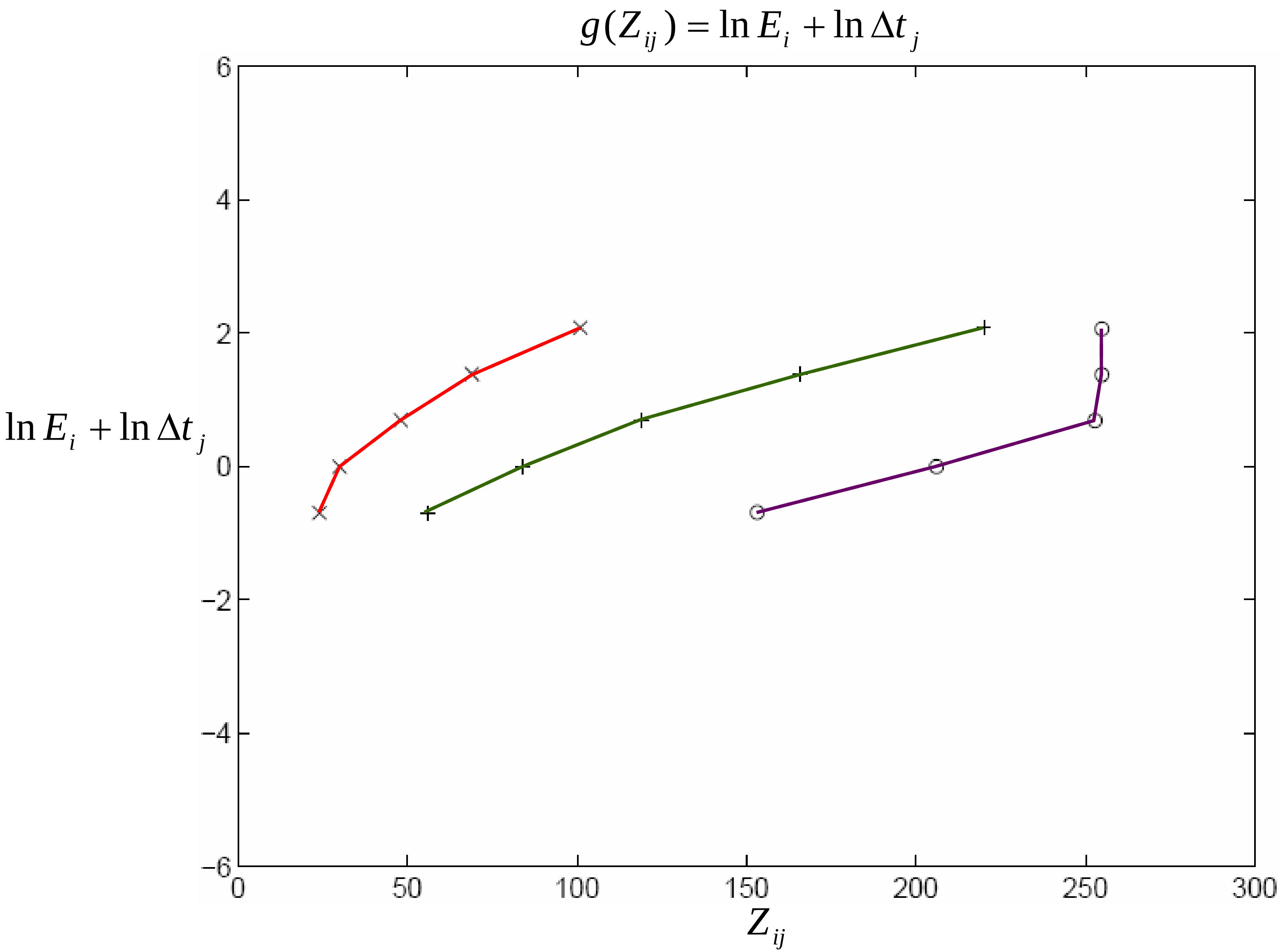
$$f^{-1}(Z_{ij}) = E_i \Delta t_j$$

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j, \text{ where } g = \ln f^{-1}$$

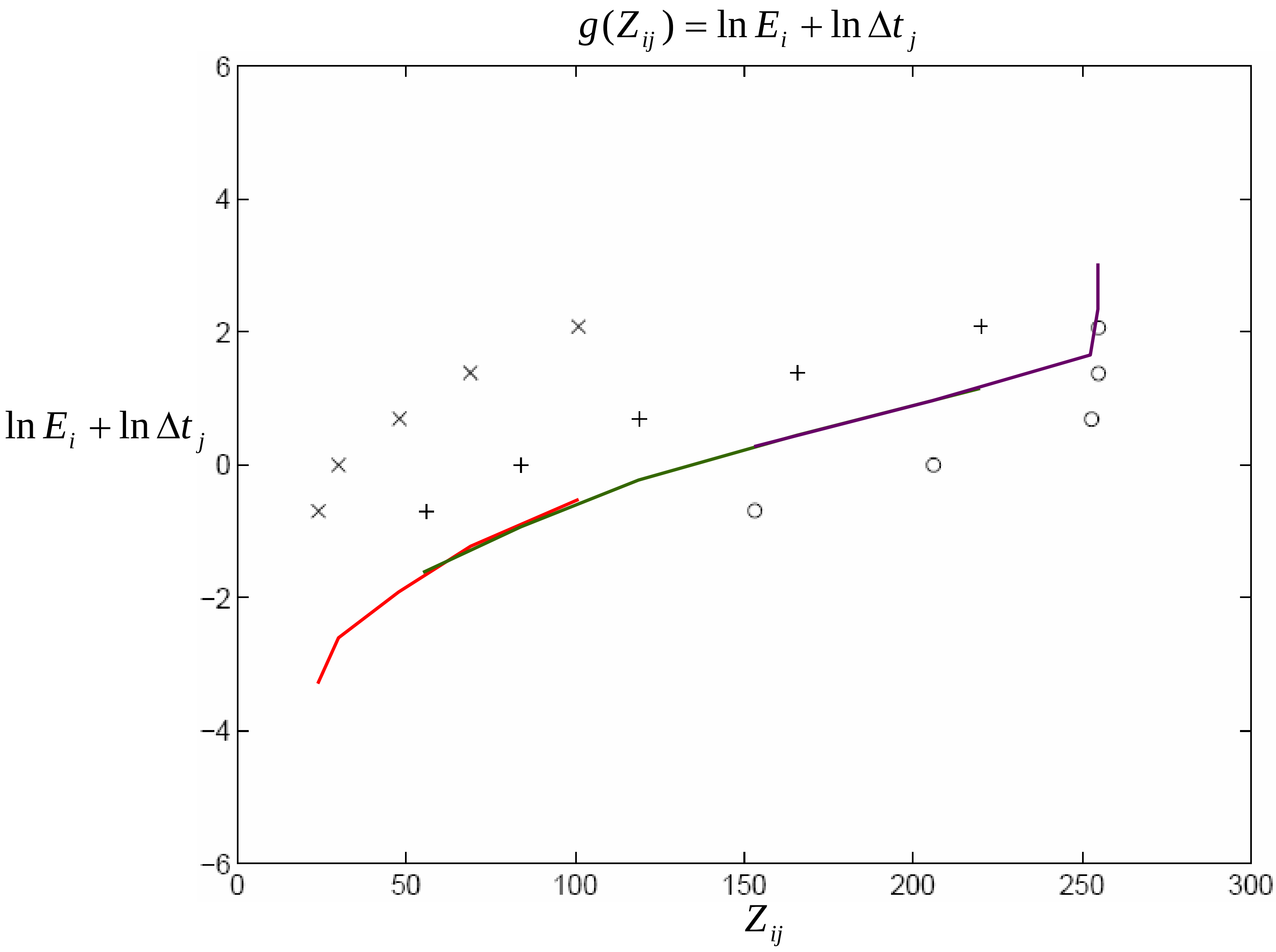
Idea behind the math



Idea behind the math



Idea behind the math



Math for recovering response curve

$$Z_{ij} = f(E_i \Delta t_j)$$

f is monotonic, it is invertible

$$\ln f^{-1}(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

let us define function $g = \ln f^{-1}$

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j$$

minimize the following

$$\mathcal{O} = \sum_{i=1}^{N} \sum_{j=1}^{P} [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]^2 + \lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} g''(z)^2$$

$$g''(z) = g(z-1) - 2g(z) + g(z+1)$$

Recovering response curve

- The solution can be only up to a scale, add a constraint

$$g(Z_{mid}) = 0, \text{ where } Z_{mid} = \frac{1}{2}(Z_{min} + Z_{max})$$

- Add a hat weighting function

$$w(z) = \begin{cases} z - Z_{min} & \text{for } z \le \frac{1}{2}(Z_{min} + Z_{max}) \\ Z_{max} - z & \text{for } z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases}$$

$$\mathcal{O} = \sum_{i=1}^{N} \sum_{j=1}^{P} \{w(Z_{ij}) \left[g(Z_{ij}) - \ln E_i - \ln \Delta t_j\right]\}^2 +$$

$$\lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} \left[w(z) g''(z)\right]^2$$

Recovering response curve

$$N(P-1) > (Z_{max} - Z_{min})$$

- We want

  If P=11, N~25 (typically 50 is used)
- We prefer that selected pixels are well distributed and sampled from constant regions. They picked points by hand.
- It is an overdetermined system of linear equations and can be solved using SVD

How to optimize?

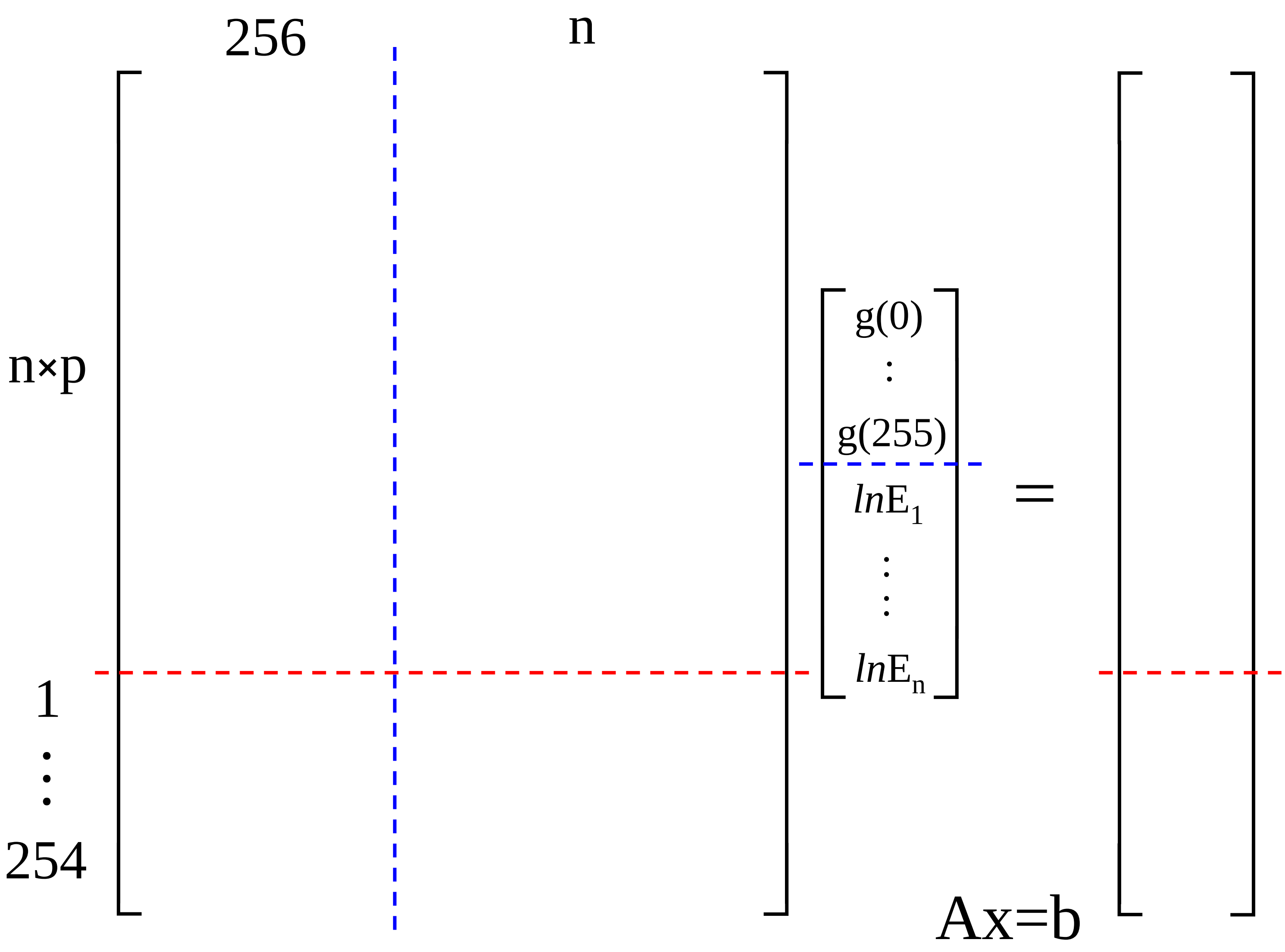
$$\mathcal{O} = \sum_{i=1}^{N} \sum_{j=1}^{P} \{w(Z_{ij}) [g(Z_{ij}) - \ln E_i - \ln \Delta t_j]\}^2 +$$

$$\lambda \sum_{z=Z_{min}+1}^{Z_{max}-1} [w(z) g''(z)]^2$$

1. Set partial derivatives zero
2.

$$\min \sum_{i=1}^{N} (\mathbf{a_i x} - \mathbf{b_i})^2 \rightarrow \text{least-square solution of } \begin{bmatrix} \mathbf{a_1} \\ \mathbf{a_2} \\ \vdots \\ \mathbf{a_N} \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \\ \vdots \\ \mathbf{b_N} \end{bmatrix}$$

Sparse linear system



Questions

- Will g(127)=0 always be satisfied? Why and why not?
- How to find the least-square solution for an over-determined system?

Least-square solution for a linear system

$$\underset{m \times n}{\mathbf{A}}\ \underset{n}{\mathbf{x}} = \underset{m}{\mathbf{b}}$$

$$m > n$$

The are often mutually incompatible. We instead find $\mathbf{x}$ to minimize the norm $\|\mathbf{Ax} - \mathbf{b}\|$ of the residual vector $\mathbf{Ax} - \mathbf{b}$. If there are multiple solutions, we prefer the one with the minimal length $\|\mathbf{x}\|$.

Least-square solution for a linear system

If we perform SVD on $\mathbf{A}$ and rewrite it as

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}V^{\mathbf{T}}$$

then $\hat{\mathbf{x}} = \boxed{\mathbf{V}\mathbf{\Sigma}^{+}\mathbf{U}^{\mathbf{T}}}\mathbf{b}$ is the least-square solution.

pseudo inverse

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \\ 0 & & & & & 0 \\ \vdots & & & & & \vdots \\ 0 & & & & & 0 \end{bmatrix}$$

$$\mathbf{\Sigma}^{+} = \begin{bmatrix} 1/\sigma_1 & & & & & 0 & \cdots & 0 \\ & \ddots & & & & & & \\ & & 1/\sigma_r & & & \vdots & & \vdots \\ & & & 0 & & & & \\ & & & & \ddots & & & \\ & & & & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Libraries for SVD

- Matlab
- GSL
- Boost
- LAPACK (recommended)
- ATLAS

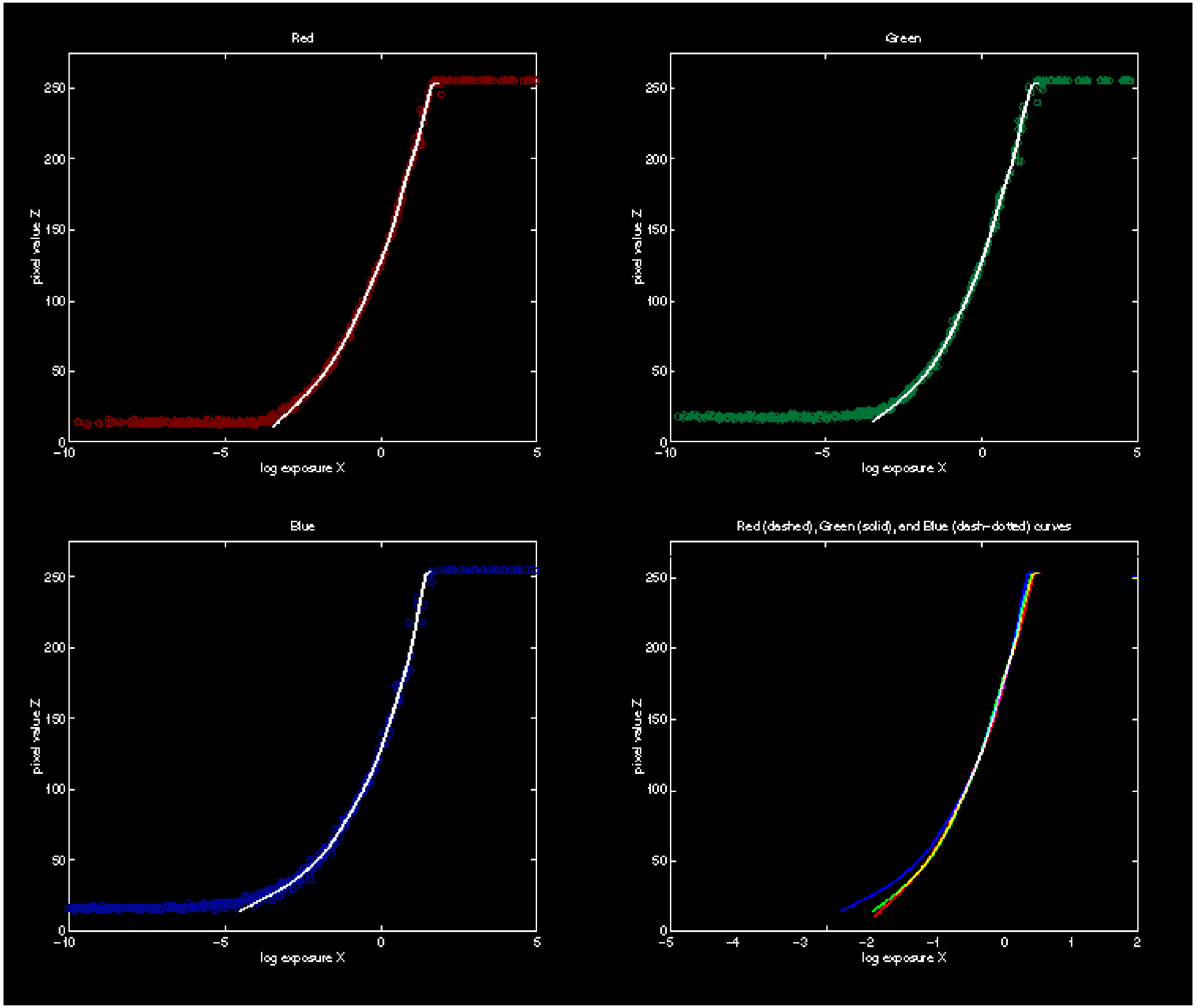
Matlab code

```
%
% gsolve.m - Solve for imaging system response function
%
% Given a set of pixel values observed for several pixels in several
% images with different exposure times, this function returns the
% imaging system's response function g as well as the log film irradiance
% values for the observed pixels.
%
% Assumes:
%
%  Zmin = 0
%  Zmax = 255
%
% Arguments:
%
%  Z(i,j) is the pixel values of pixel location number i in image j
%  B(j)   is the log delta t, or log shutter speed, for image j
%  l      is lamdba, the constant that determines the amount of smoothness
%  w(z)   is the weighting function value for pixel value z
%
% Returns:
%
%  g(z)   is the log exposure corresponding to pixel value z
%  lE(i)  is the log film irradiance at pixel location i
%
```

Matlab code

```
function [g,lE]=gsolve(Z,B,l,w)

n = 256;
A = zeros(size(Z,1)*size(Z,2)+n+1,n+size(Z,1));
b = zeros(size(A,1),1);

k = 1;                  %% Include the data-fitting equations
for i=1:size(Z,1)
  for j=1:size(Z,2)
    wij = w(Z(i,j)+1);
    A(k,Z(i,j)+1) = wij; A(k,n+i) = -wij; b(k,1) = wij * B(i,j);
    k=k+1;
  end
end

A(k,129) = 1;           %% Fix the curve by setting its middle value to 0
k=k+1;

for i=1:n-2             %% Include the smoothness equations
  A(k,i)=l*w(i+1); A(k,i+1)=-2*l*w(i+1); A(k,i+2)=l*w(i+1);
  k=k+1;
end

x = A\b;                %% Solve the system using SVD

g = x(1:n);
lE = x(n+1:size(x,1));
```

Recovered response function



Constructing HDR radiance map

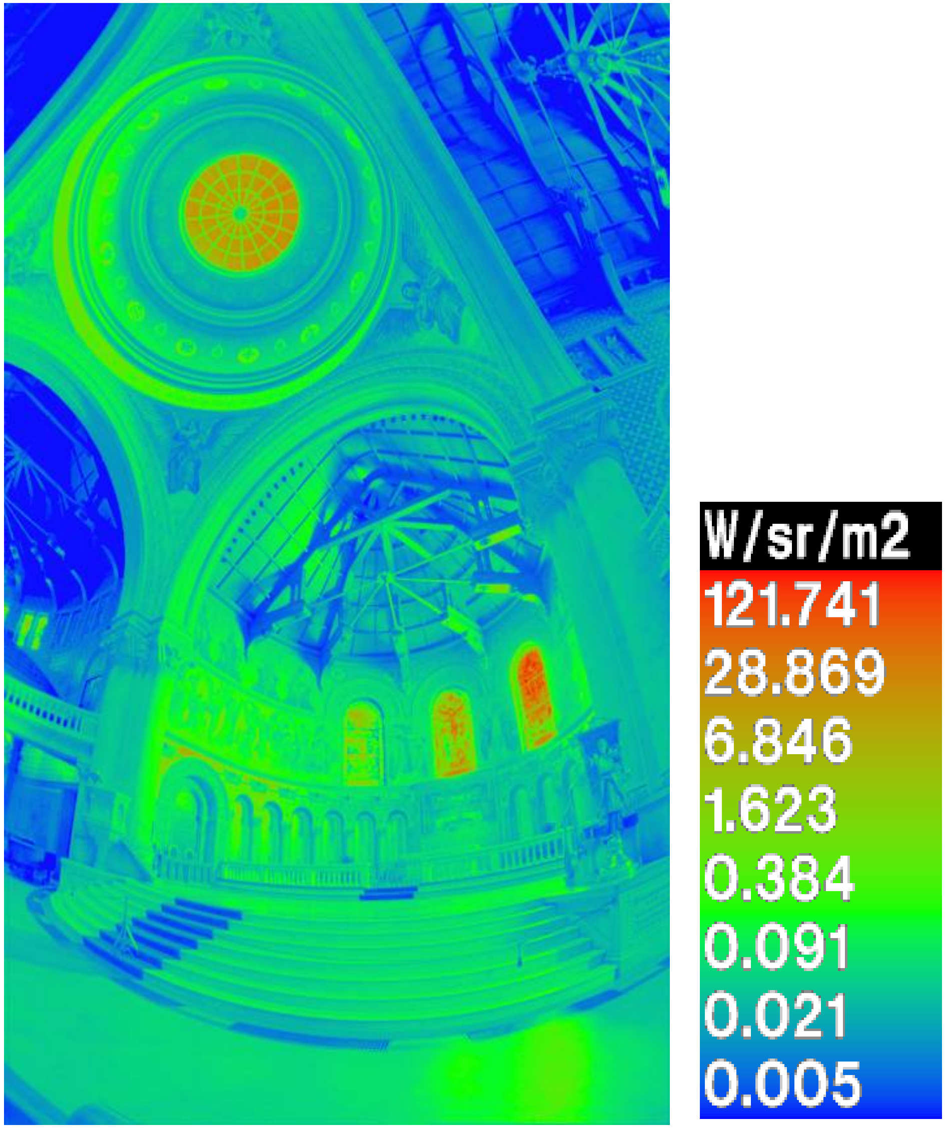
$$\ln E_i = g(Z_{ij}) - \ln \Delta t_j$$

combine pixels to reduce noise and obtain a more reliable estimation

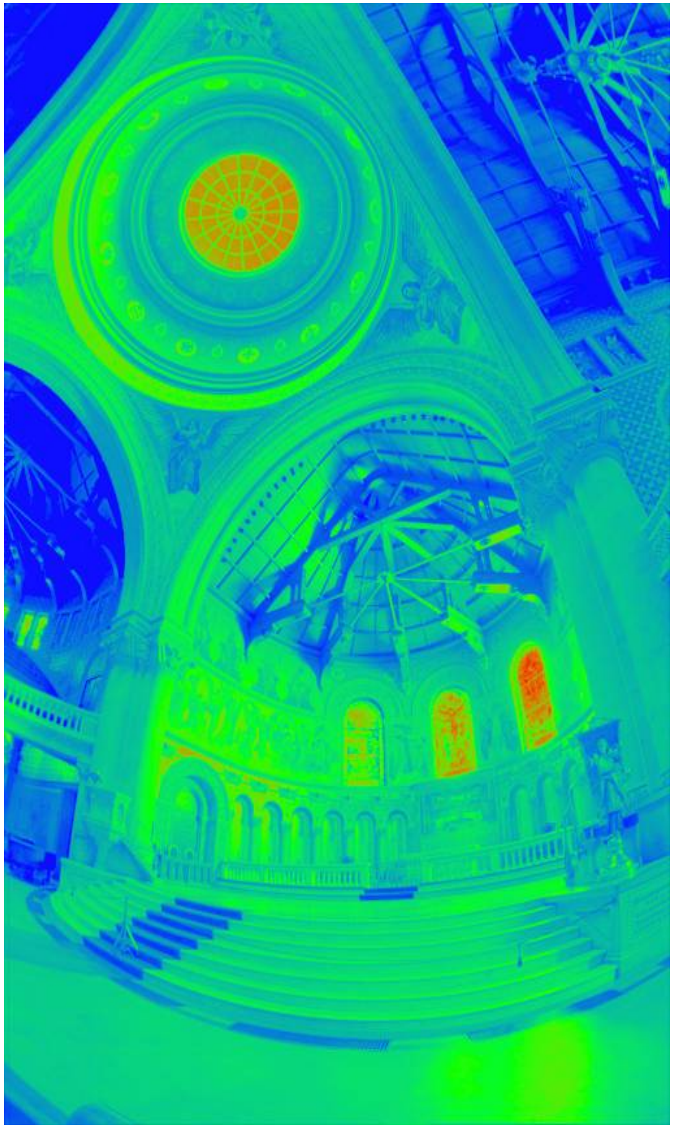
$$\ln E_i = \frac{\sum_{j=1}^{P} w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^{P} w(Z_{ij})}$$

Varying shutter speeds

Reconstructed radiance map



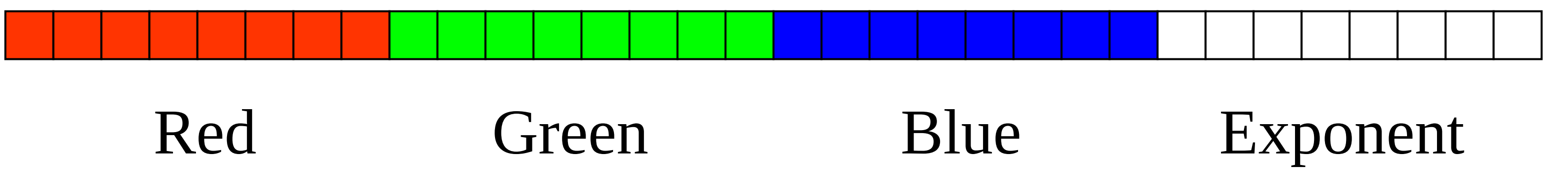
What is this for?



- Human perception
- Vision/graphics applications

Radiance format (.pic, .hdr, .rad)

32 bits/pixel



(145, 215, 87, 149) =

(145, 215, 87) * 2^(149-128) =

1190000 1760000 713000

(145, 215, 87, 103) =

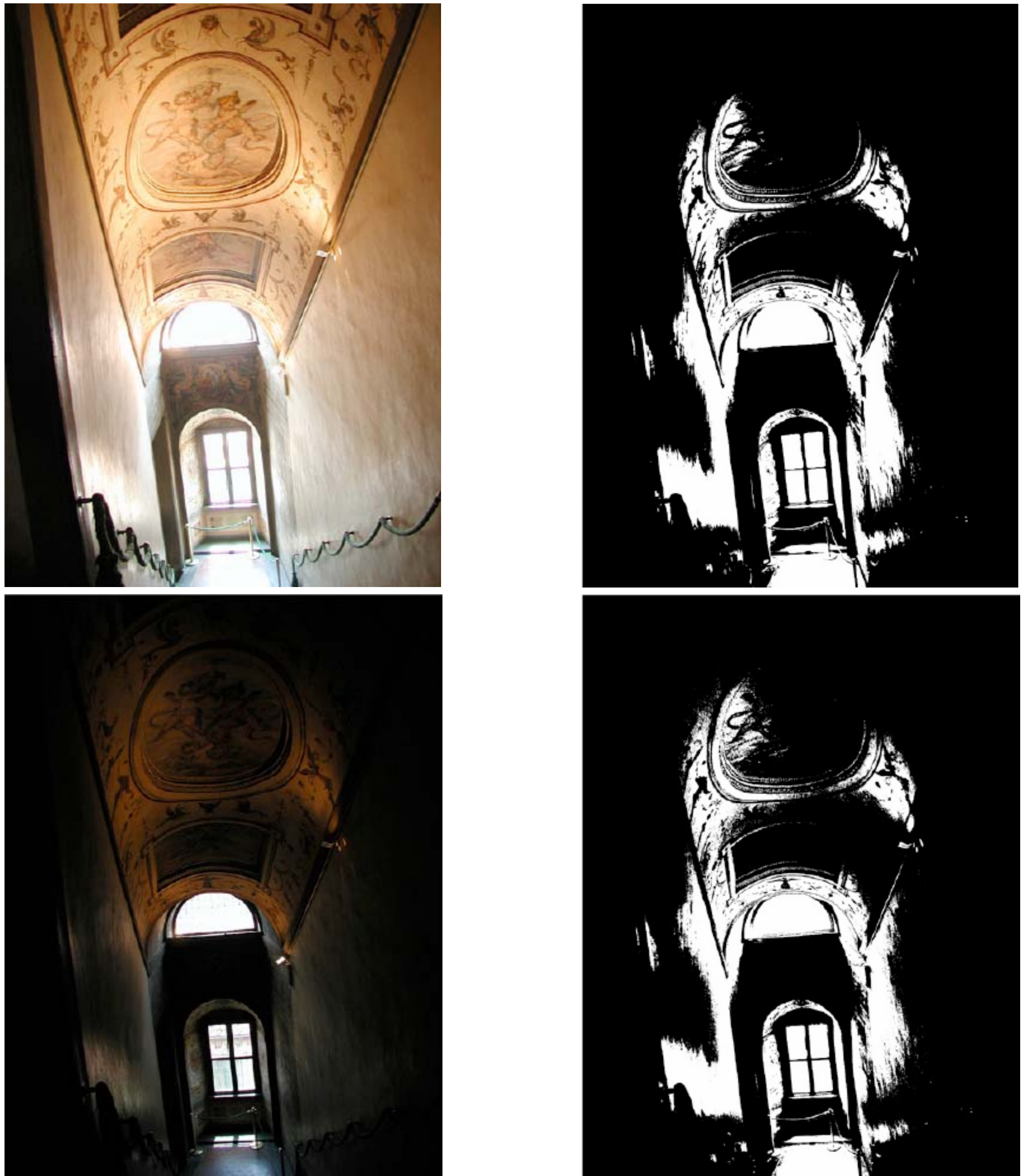
(145, 215, 87) * 2^(103-128) =

0.00000432 0.00000641 0.00000259

Ward, Greg. "Real Pixels," in Graphics Gems IV, edited by James Arvo, Academic Press, 1994

Demo

Image alignment



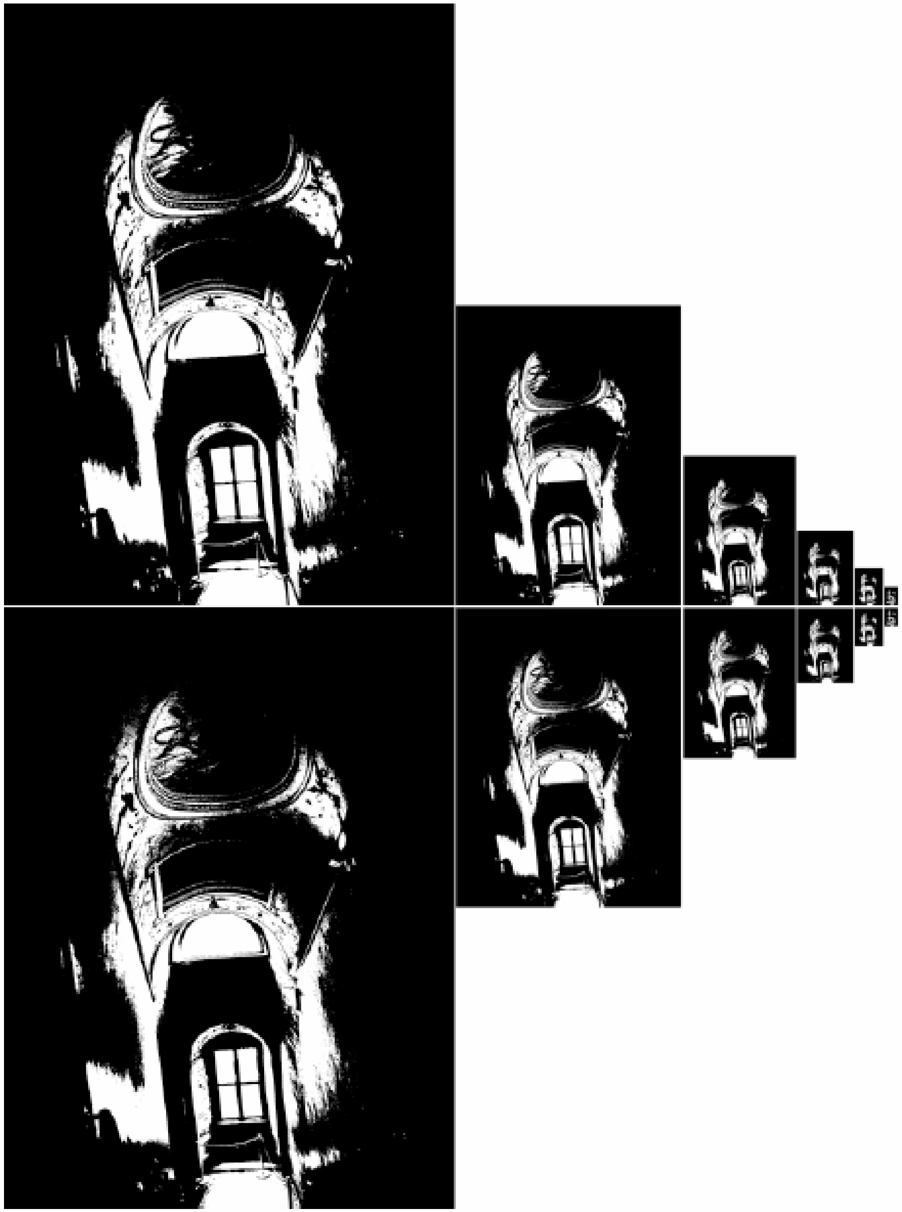
Median Threshold Bitmap (MTB) alignment

- Consider only integral translations. It is enough empirically.
- The inputs are N grayscale images. (You can either use the green channel or convert into grayscale by Y=(54R+183G+19B)/256)
- MTB is a binary image formed by thresholding the input image using the median of intensities.

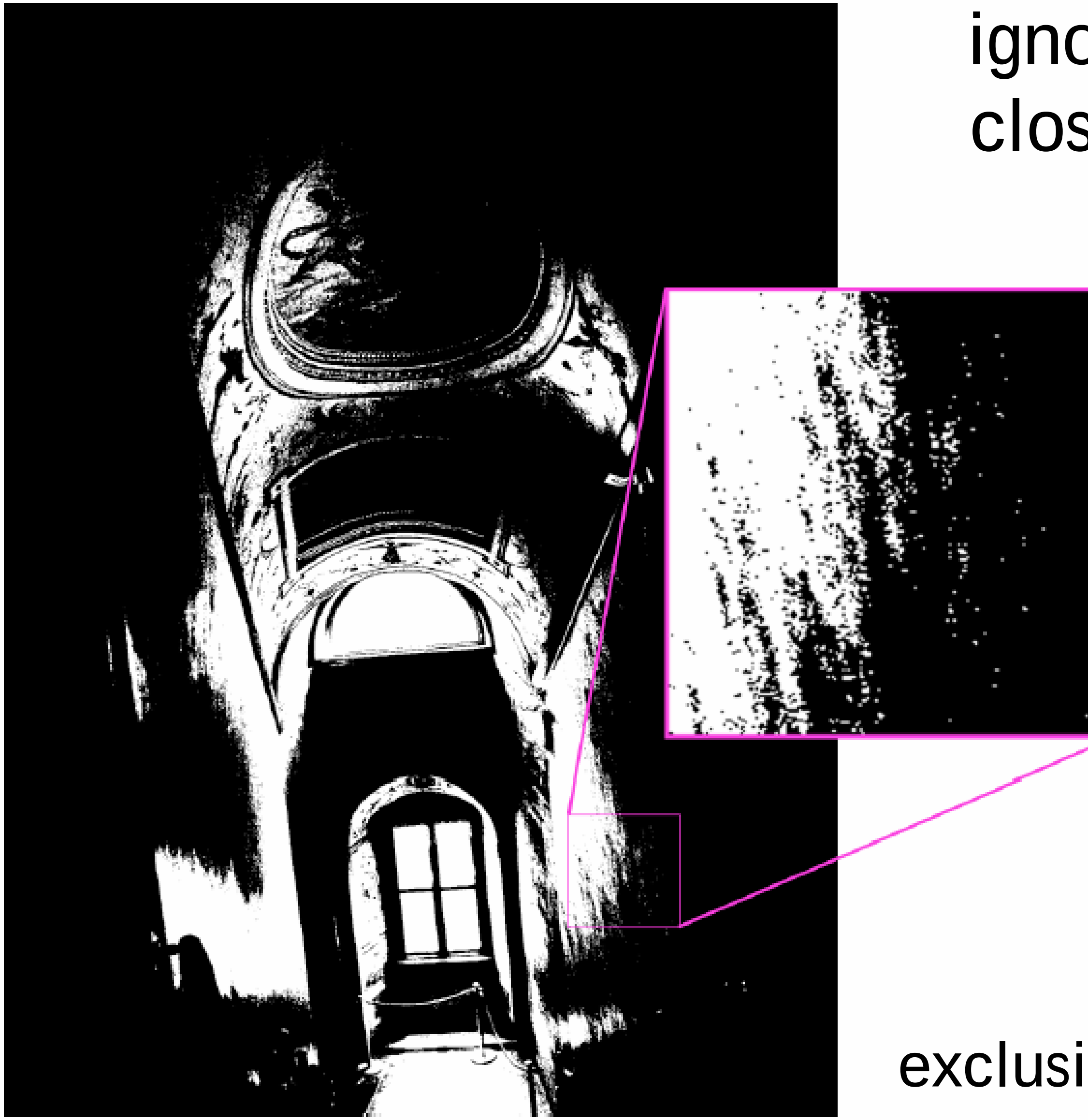
Search for the optimal offset

- Try all possible offsets.
- Gradient descent
- Multiscale technique

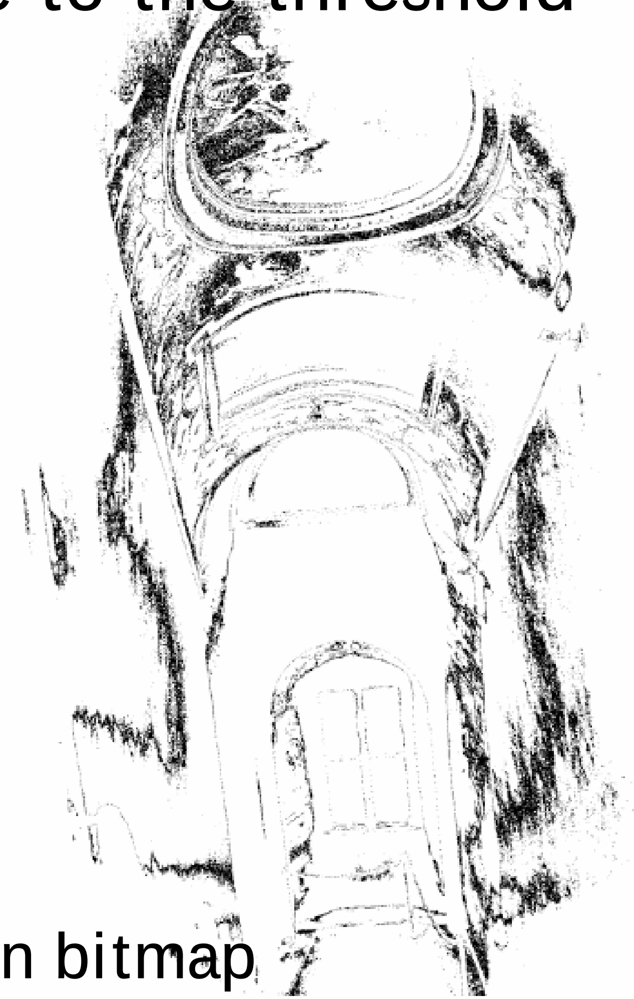
- log(max_offset) levels
- Try 9 possibilities for the top level
- Scale by 2 when passing down; try its 9 neighbors



Threshold noise



ignore pixels that are close to the threshold



exclusion bitmap

Results

Success rate = 84%. 10% failure due to rotation. 3% for excessive motion and 3% for too much high-frequency content.

Equipment

We provide 3 sets:

Contact TA for checkout.