# Radiometric Calibration

CS635

Daniel G. Aliaga
Department of Computer Science
Purdue University

# Radiometric Calibration

- Since color is important, we want to emit and capture the colors we expect
- Example Goal:

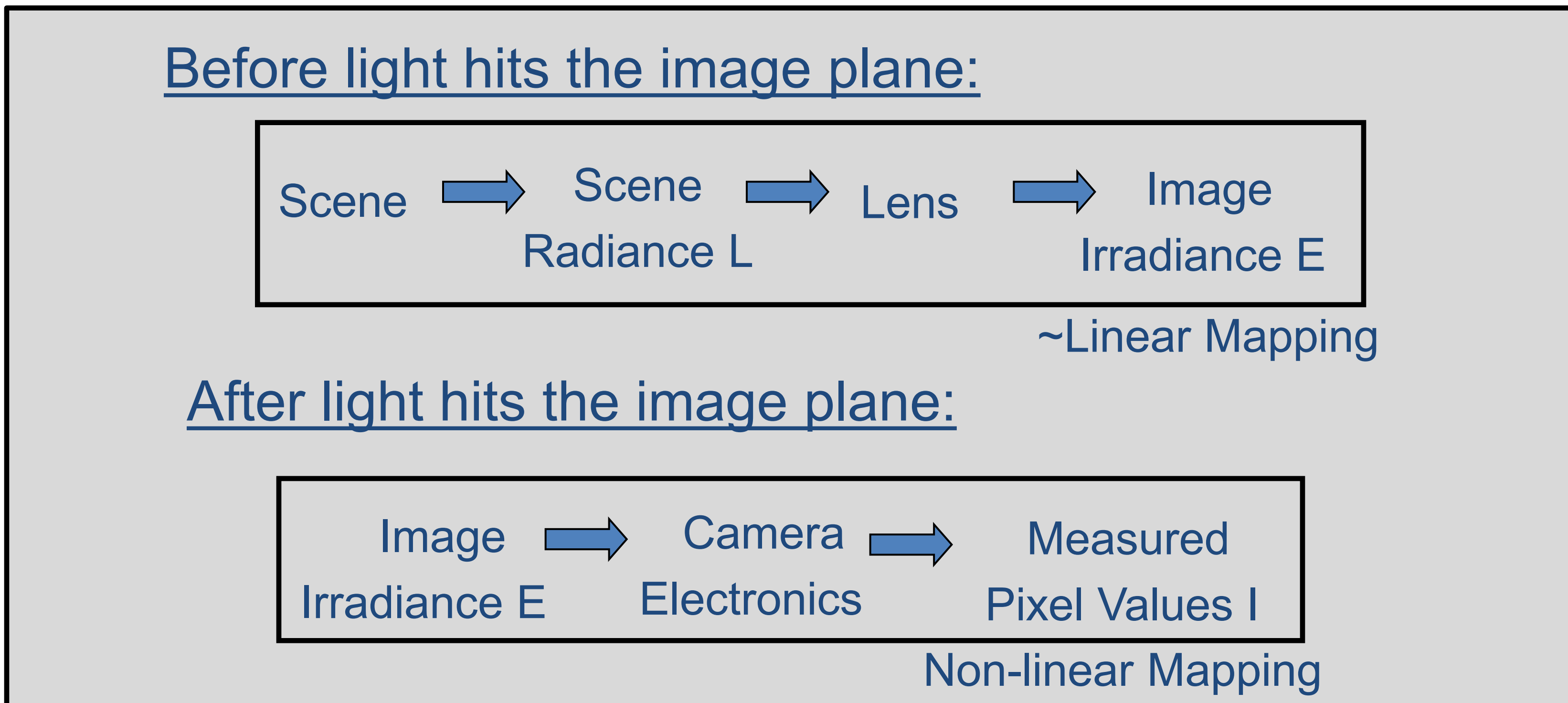
  "Given a desired color $\boldsymbol{x}$, instruct the projector to illuminate the color $\boldsymbol{y}$ such that what appears to the camera is $\boldsymbol{x}$"

  "Take a picture of $\boldsymbol{u}$ with two cameras and obtain the same pixel values $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$"

  …Ideally $\boldsymbol{x} = \boldsymbol{y}$ **and** $\boldsymbol{v_1} = \boldsymbol{v_2}$, but in practice this is not the case

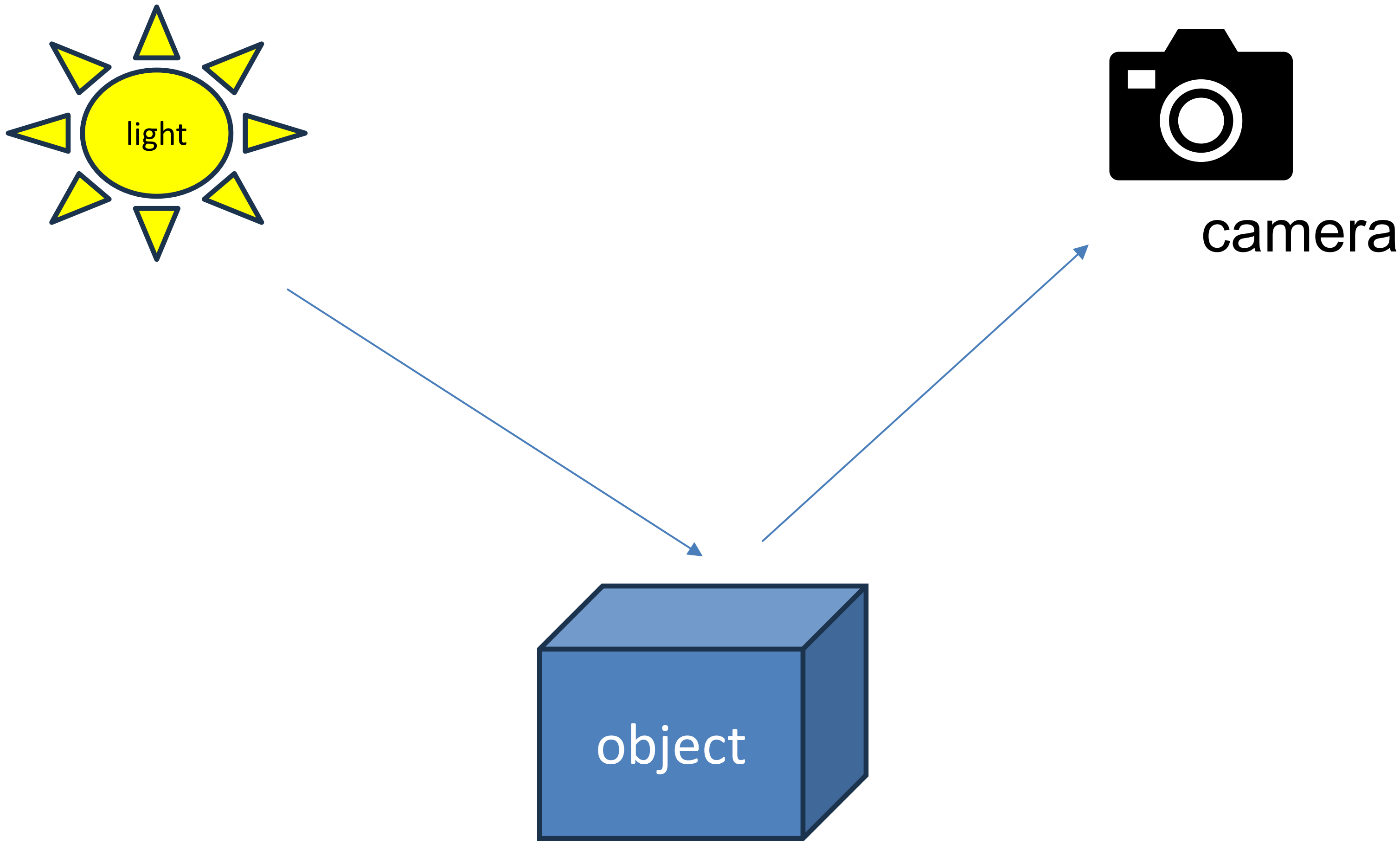
# Radiometric Calibration

- Coordinate the per-channel color intensities between multiple projectors/light-sources

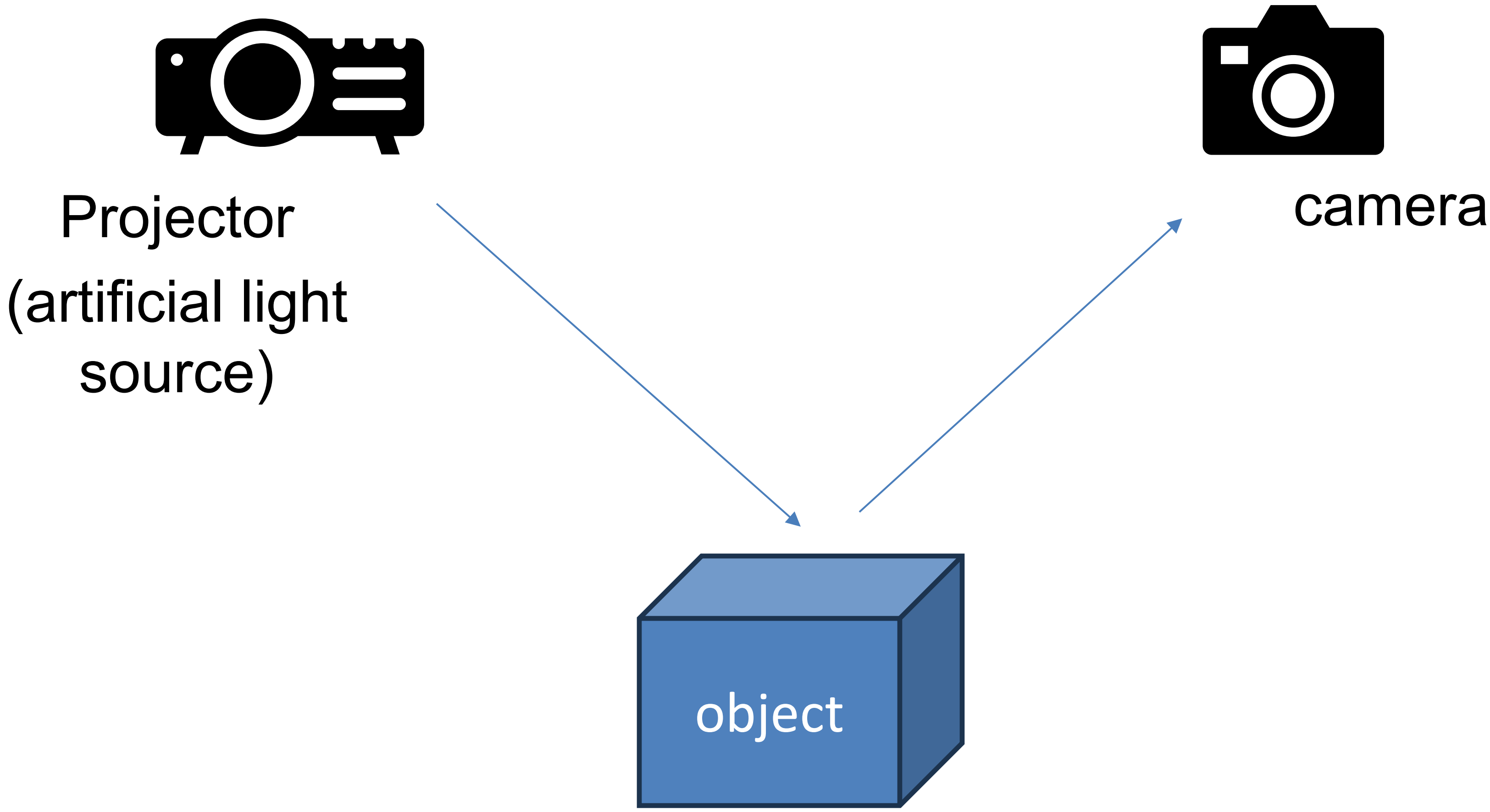



Can we go from measured pixel value, I, to scene radiance L?

# Basic Setup (A)



# Basic Setup (B)

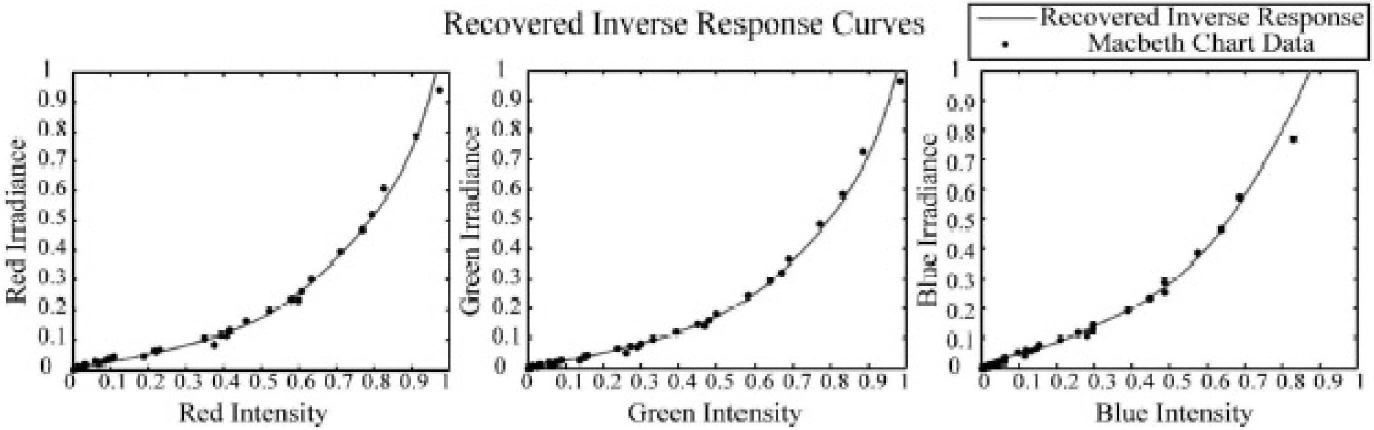


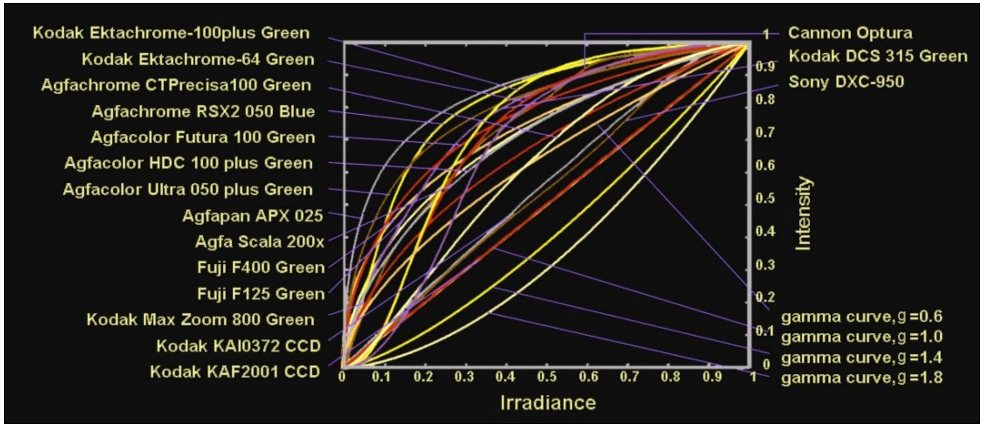
# Grayscale Calibration

- Assuming n-bits of gray
- Illuminate with $x$, the camera sees $y$
- What is a simple calibration option?

# Main Issue

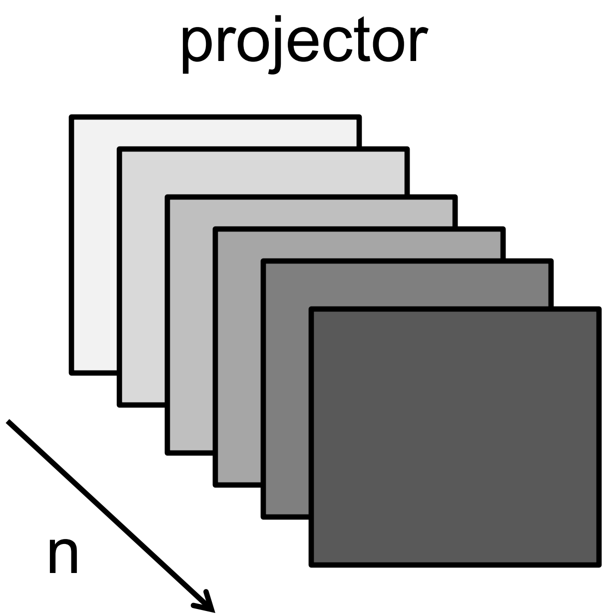
- Linear vs. Nonlinear Response

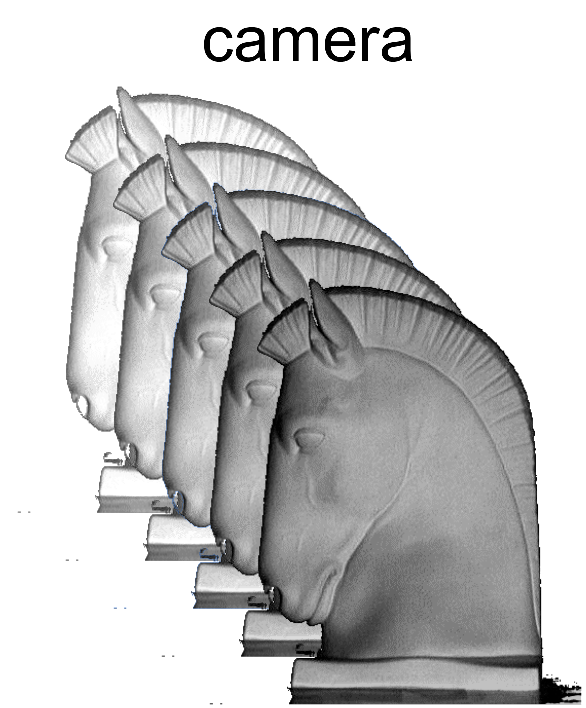


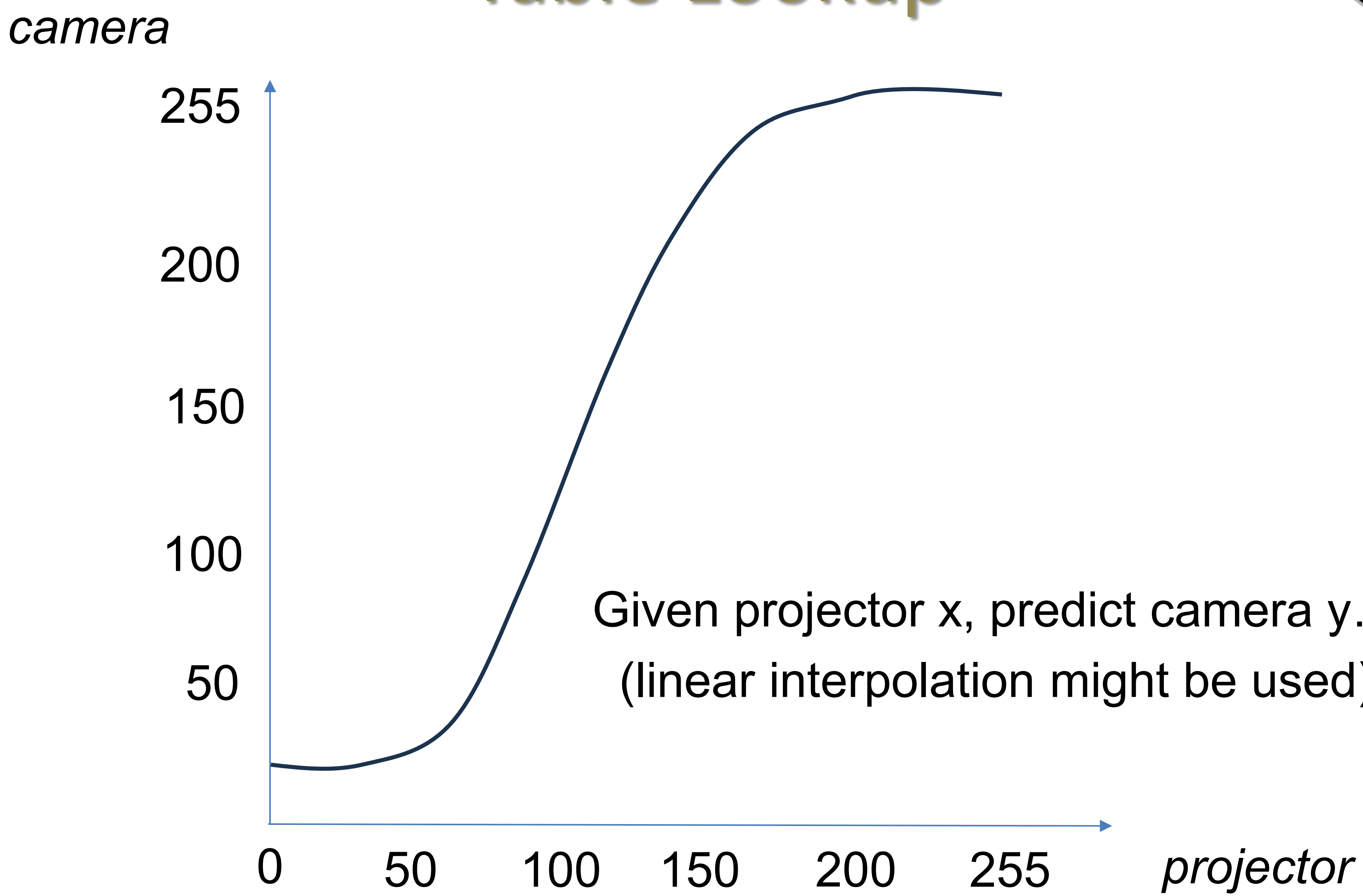
# Grayscale Calibration

- Assuming n-bit channels, project 2^n “gray-level images” onto the scene and capture the appearance 2^n times



# Table Lookup

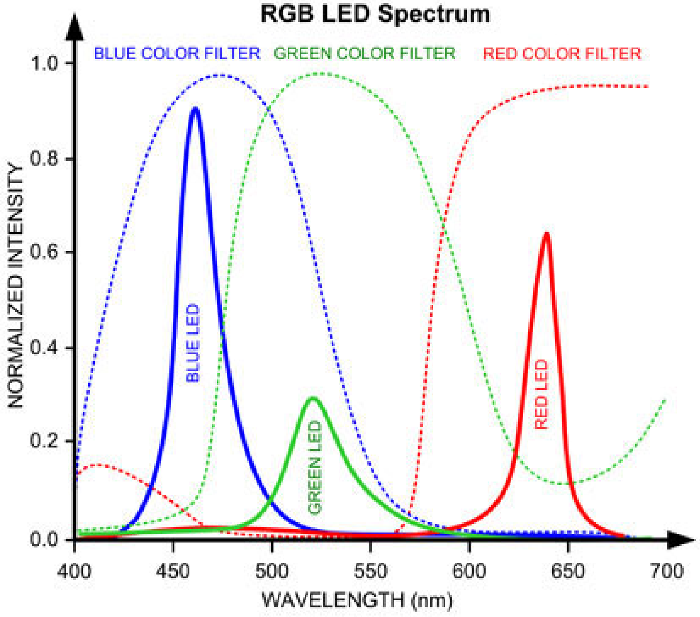



Given projector x, predict camera y…
(linear interpolation might be used)

# How to extend to color?

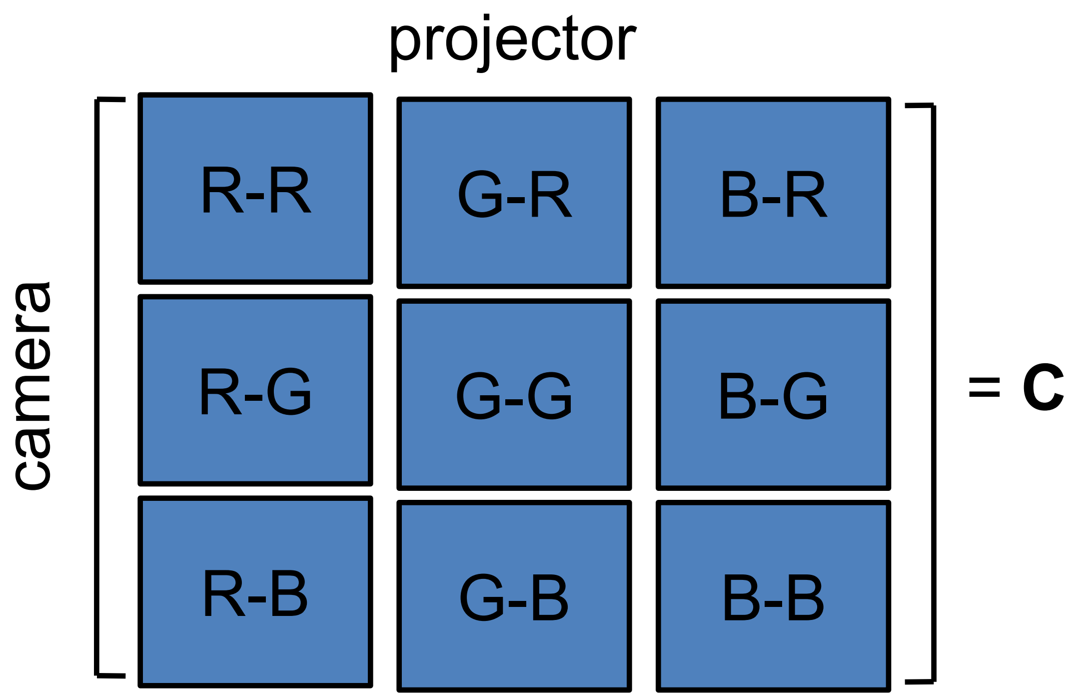
# How to extend to color?

- Color channels are overlapping, thus the color mixture must be taken into account



# Color Calibration

- Assuming n-bit RGB channels, project 3 x 2^n "single-color images" onto the scene and capture the appearance 3 x 2^n times



# Color Calibration Option A

- **x** = projector color, **y** = camera color
- **3D Table Lookup:**
  - Forward: given x, predict y
  - Inverse: given y,  find x
  - (trilinear interpolation could be used for subsample precision)

# Color Calibration Option B

- **x** = projector color, **y** = camera color
- **C** = linear color mixture matrix for all intensities
- Thus

$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

$$\mathbf{x} = \mathbf{C}^{-1}\mathbf{y}$$

# Lets Look More Closely…

- Radiometric response function:

$$E = L\frac{\pi}{4}\left(\frac{d}{h}\right)^2 \cos^4\phi$$

Where $E$ = image irradiance, $L$ = scene radiance, $h$ = focal length, $d$ = aperture diameter, $\phi$ = angle subtended by principal ray from optical axis

# Lets Look More Closely…

- In an ideal world, this becomes:

$$I = Et, or\ I = Lke$$

Where $k = 1/\mathrm{h}^2 \cos^4 \phi$ and $e = \left(\frac{\pi d^2}{4}\right) t$, and $e$ is the exposure of the image (depends on $d$ and $t$)

# Lets Look More Closely…

- Sources are nonlinearity are various, but all together we know measured brightness $M$ is:

$$M = g(I)$$
$$I = f(M)$$

  for some functions $f$ and $g$.

- **Recovering $f$ is the radiometric calibration problem**

# MacBeth Chart

- Observe this pre-calibrated chart with uniform illumination
- Useful to calibrate a camera, but not a projector-camera system

# Can you recover with calibrated charts?

- One observation:
  - measured brightness values change with exposure, scene radiance values L remain constant
  - Capture images at different exposures (diameter or time), and fit back to the function

# Debevec and Malik

- For camera radiometric calibration, take photographs of same scene and illumination conditions but under different exposure times
  - Exposures $\Delta t_j$
  - Pixel irradiances $E_i$ are constant
  - Pixel values $Z_{ij}$

# Debevec and Malik

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

Unknowns are $E_i$ and $g$ and note $Z_{ij}$ is in finite range (e.g., $Z_{min}$ to $Z_{max}$)

# Debevec and Malik

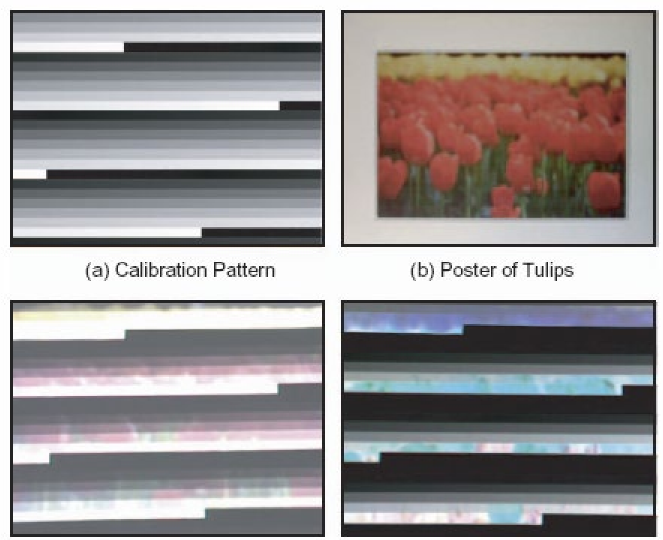
- Minimize with SVD:

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

(e.g., smooth $g''(z) = g(z-1) + 2g(z) + g(z+1)$)

# Grossberg et al.

- For camera+projector calibration, instead of 256x3x3 = 6912 images, maybe 6 is enough:


(a) Calibration Pattern (b) Poster of Tulips
(c) Pattern Projected on Tulips (d) Computed Invariant

- Assumes pixels behavior similarly

- Does not handle effect of “inter-reflections”…

# Combining Lights/Projectors

- Objective – find the weights for each pixel of each projector such that all object points are under a bounded amount of light $E_{bound}$
  - $E_{bound}$ limits the light shined on the object,
  - Per-pixel weights compute an efficient use of light

# Option 1: Maximally Efficient Compensation

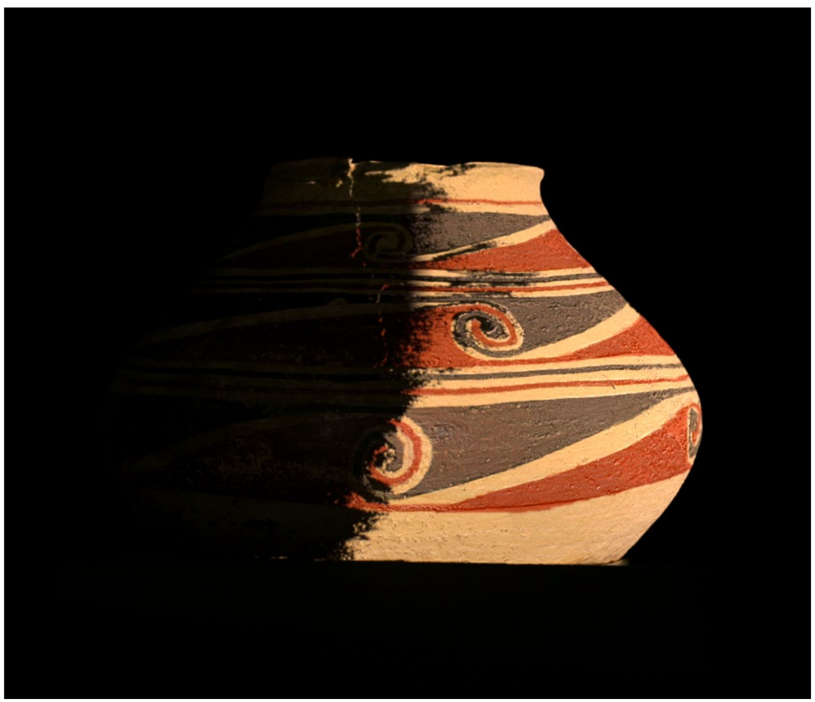
- For each object point, choose single projector which maximally illuminates it

# Option 1: Maximally Efficient Compensation

- For each object point, choose single projector which maximally illuminates it

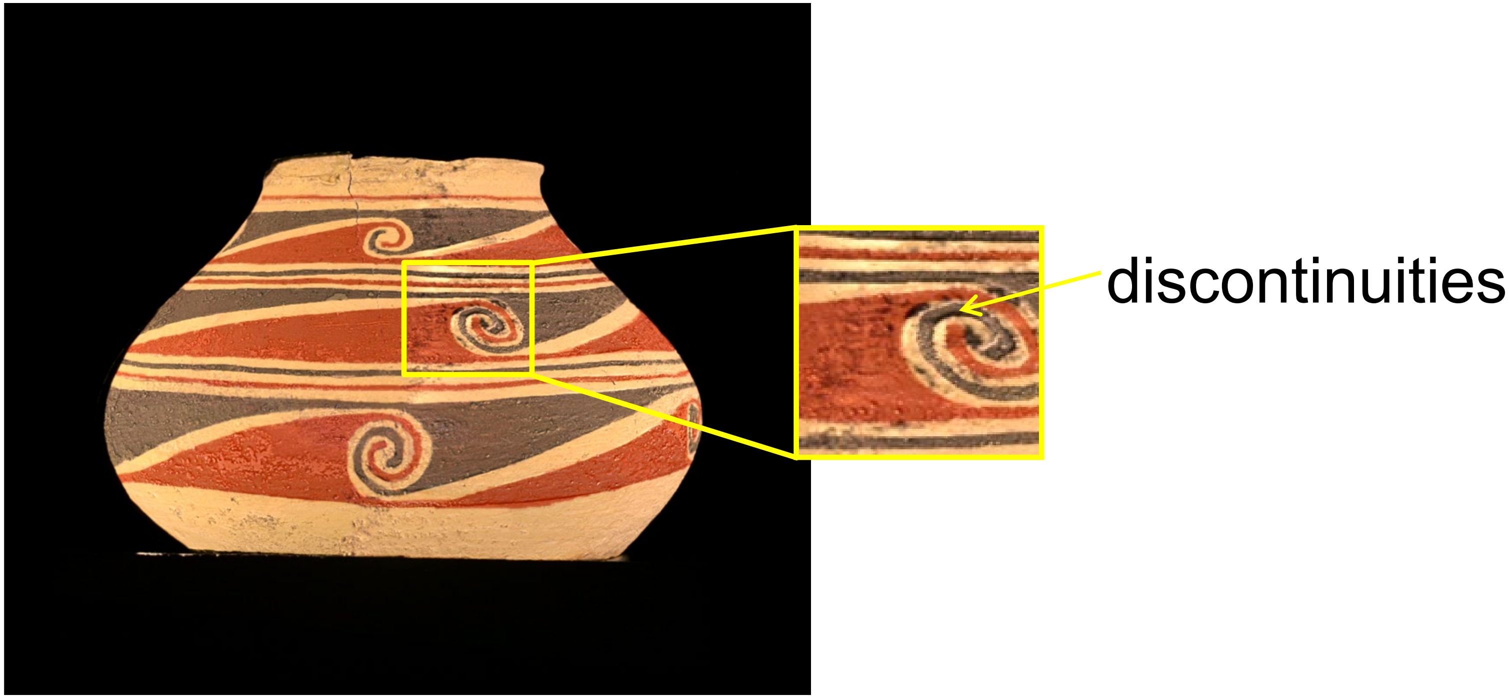
# Option 1: Maximally Efficient Compensation

- For each object point, choose single projector which maximally illuminates it



# Option 1: Maximally Efficient Compensation

- For each object point, choose single projector which maximally illuminates it
  - Most light efficient scheme yields brightest compensation
  - Sharp change from projector to projector causes discontinuities

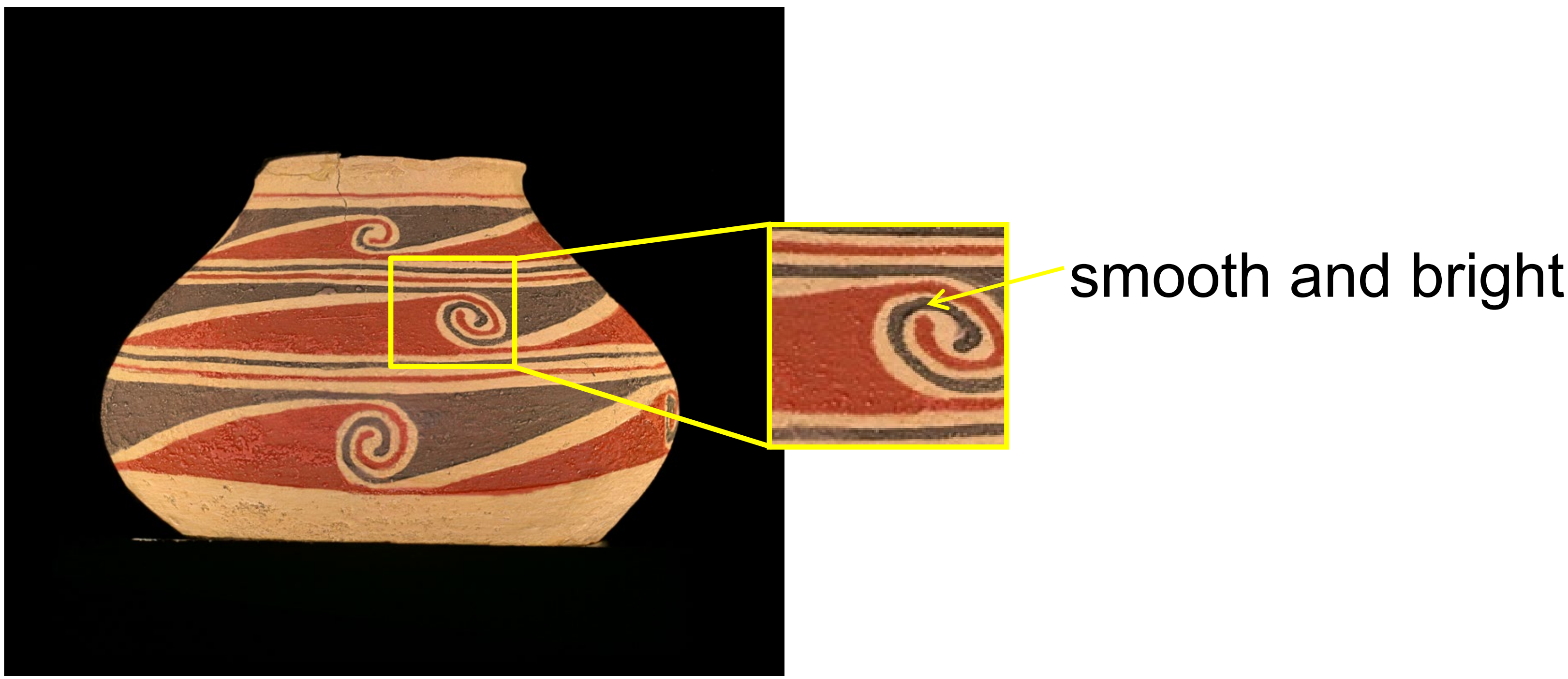


# Option 2: Linear Compensation

- For each object point, linearly weigh contribution of each projector with respect to incident angle
  - Removes discontinuity issue
  - Results in dimmer compensations

# Non-Linear Compensation

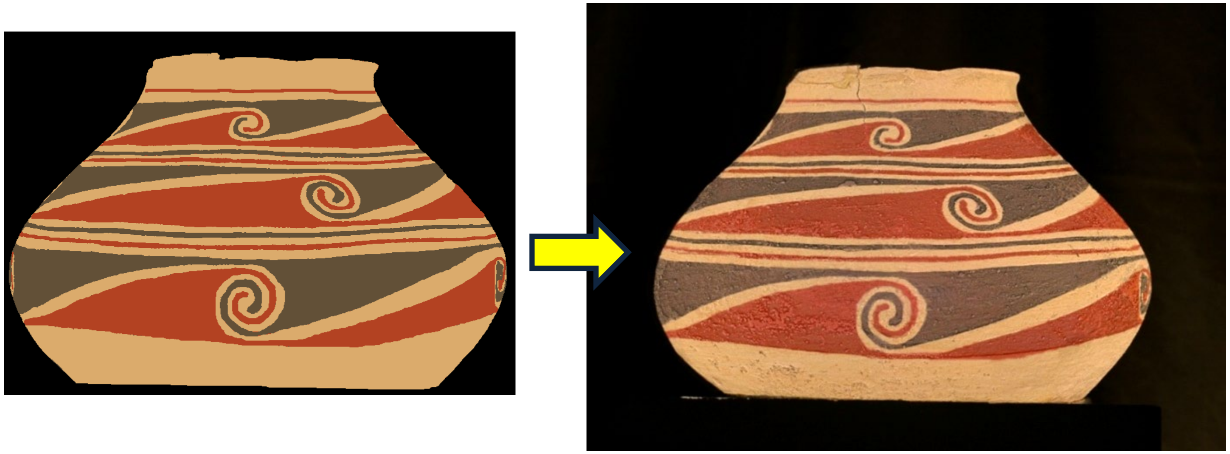
- Generalize to a nonlinear model that keeps light energy within $E_{bound}$
  - Yields smooth and bright compensations



# Multi-Projector Compensation Algorithm

1. Use surface radiance model of object
2. Compute weights $w_{ik}$ for each object point $i$ from projector $k$
3. Calculate compensation image given weights $w_{ik}$
4. Project light onto object's surface

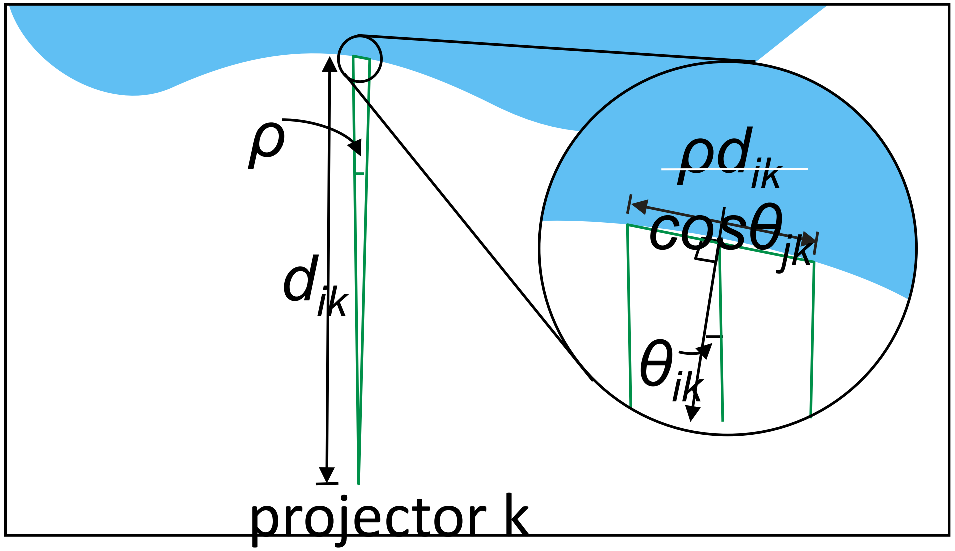
$$E_{bound} \geq E_i = \sum_{k=1}^{N_p} w_{ik} e_{ik} L_{ik}$$



# Surface Radiance Model

- Assume surface is diffuse
- $L_{ik}$ measures energy from projector *k* onto object point *i*'s area ($A_{ik}$)

$$L_{ik} = \frac{1}{A_{ik}} \approx \left(\frac{\cos\theta_{ik}}{\rho d_{ik}}\right)^2$$



# Non-linear Pixel Weighting Scheme

- Strike balance between smooth and bright compensations

$$w_{ik} = \frac{{L_{ik}}^m}{\sum_{j=1}^{N_p} {L_{ij}}^m}$$

- Generalization of different weighting techniques
  - m = 1 → Linear compensation

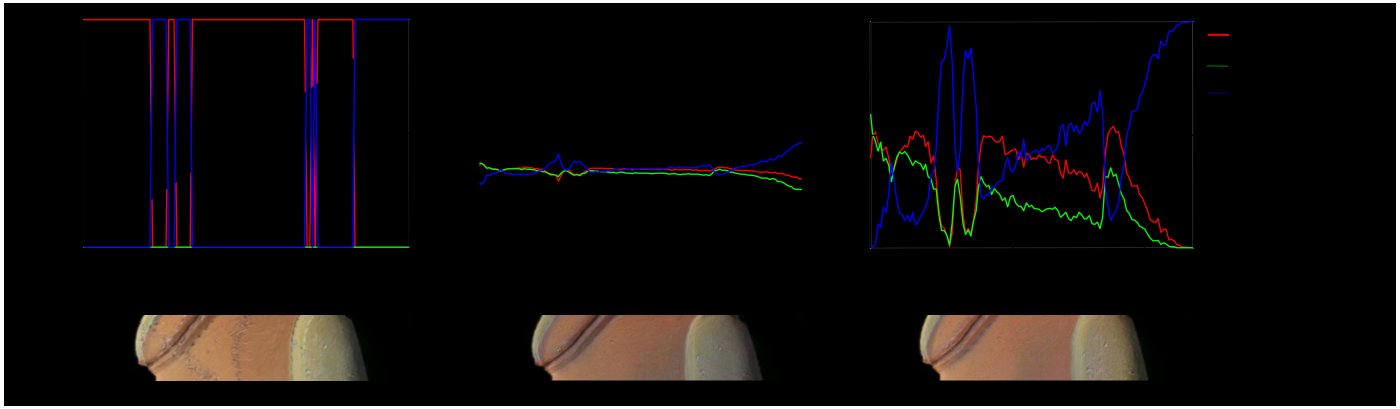
$$w_{ik} = \frac{L_{ik}}{\sum_{j=1}^{N_p} L_{ij}}$$

  - m = ∞ → Maximally efficient compensation

$$w_{ik} = \begin{cases} 1 & L_{ik} = \max_j L_{ij} \\ 0 & otherwise \end{cases}$$

# Compensation Comparison

- Combinations of projectors under different weighting schemes



maximally-bright | linear | our non-linear approach