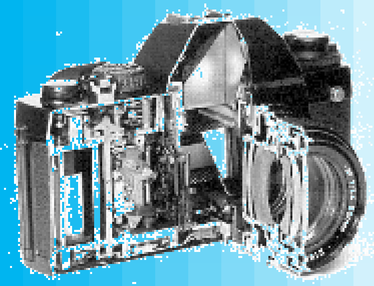
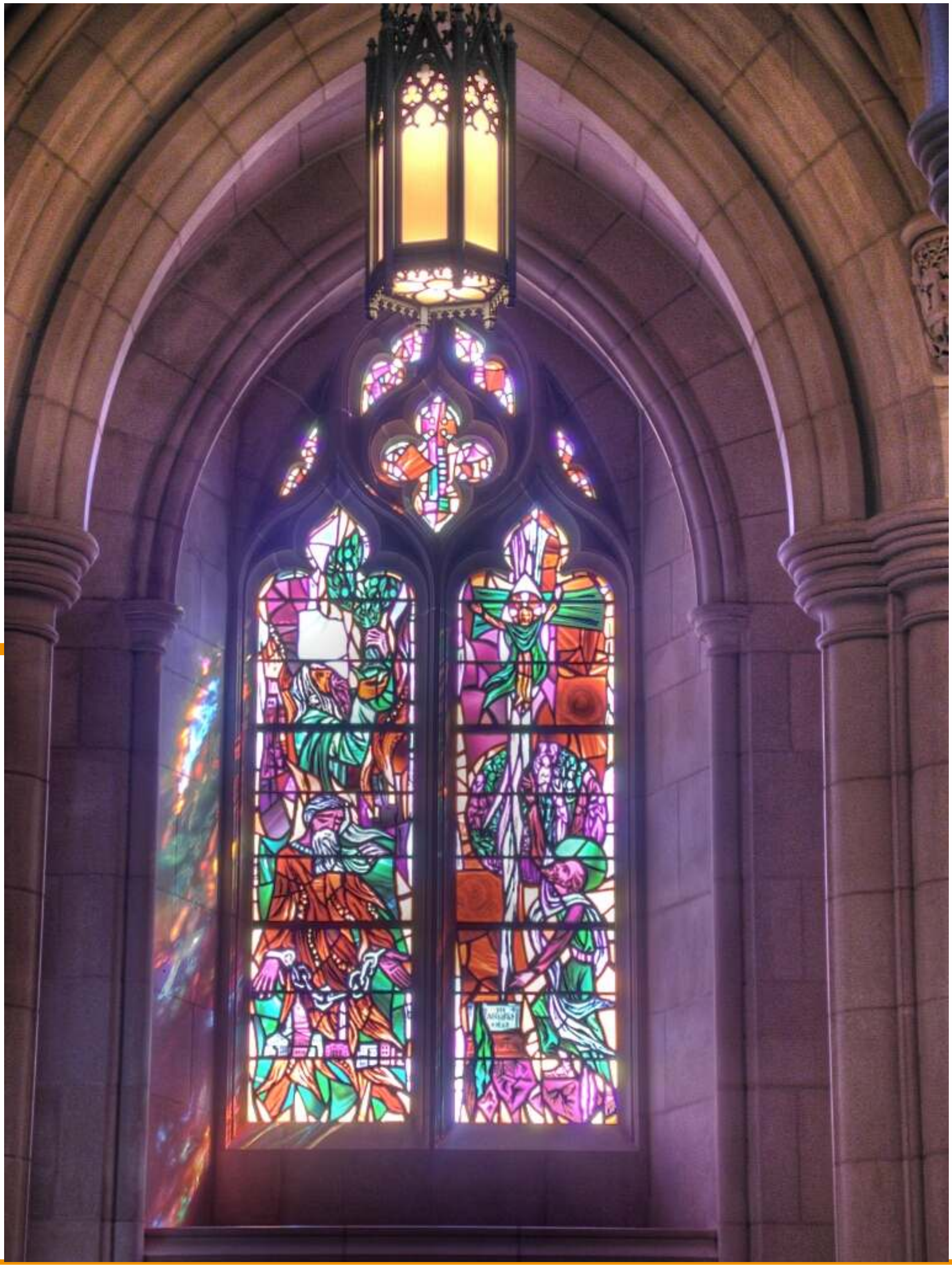
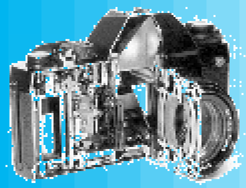


High Dynamic Range Imaging



UNIVERSITY OF DELAWARE

Department of Computer and Information Science

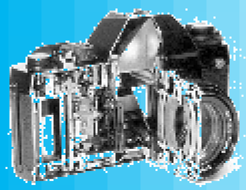


Introduction

- Problems with current imaging
- Basics of HDR images
- Different techniques
  - Tone mapping
  - HDR compression
  - Image encondings
- Applications of HDR images

8-bit problem

- **Typical images displayed on screen are 24-bits**
  - 8-bits per color component (RGB)
  - 256 different intensity levels
- **Real-world dynamic range is far greater than 256 intensity levels!**



Range of luminance

100 000 000:1

in the natural world

10 000:1

that the eye can accommodate in a single view

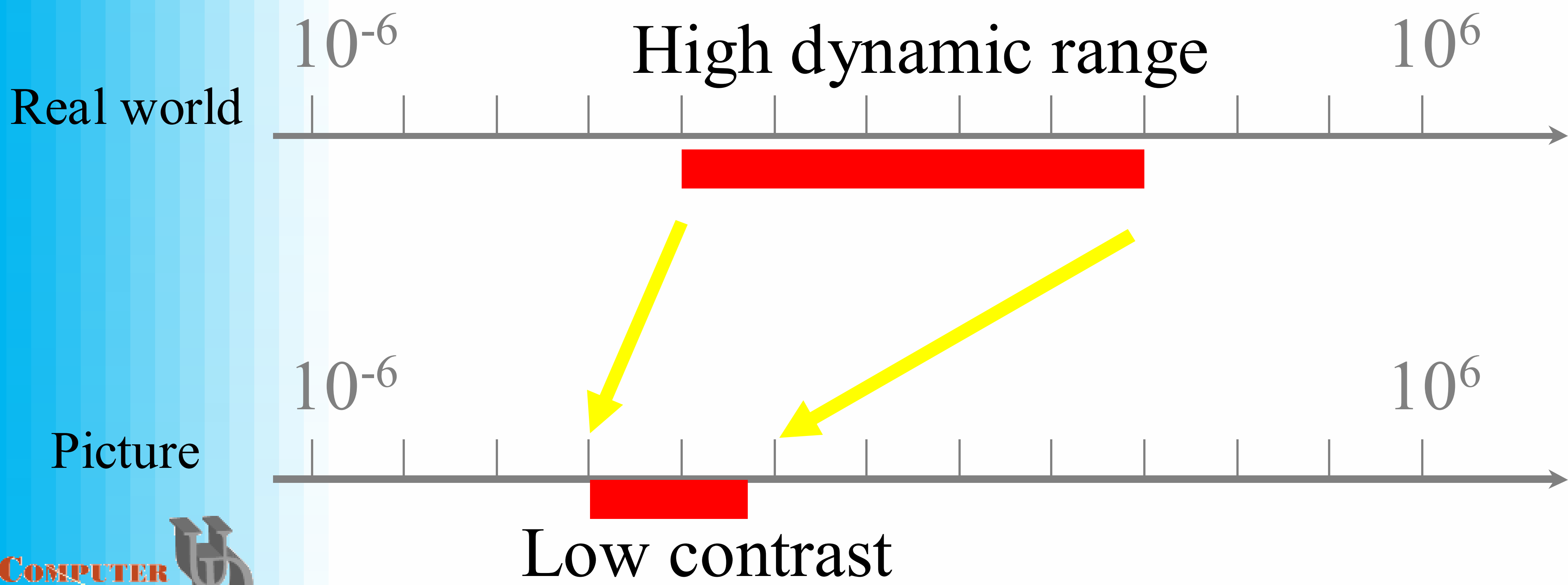
100:1

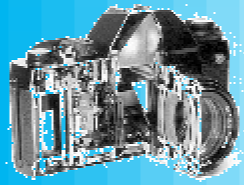
that a typical CRT/LCD monitor can display

(Source: A review of tone reproduction techniques; Devlin, 2002)

Contrast reduction

- **Match limited contrast of the medium**
- **Preserve details**





What is a High Dynamic Range image?

- HDR image is an image that has a greater dynamic range that can be
  - shown on a display device
  - captured with a camera with just a single exposure for each image
  - of course, more than 8 bit per channel
- The "dynamic range" of a scene is the contrast ratio between its brightest and darkest parts

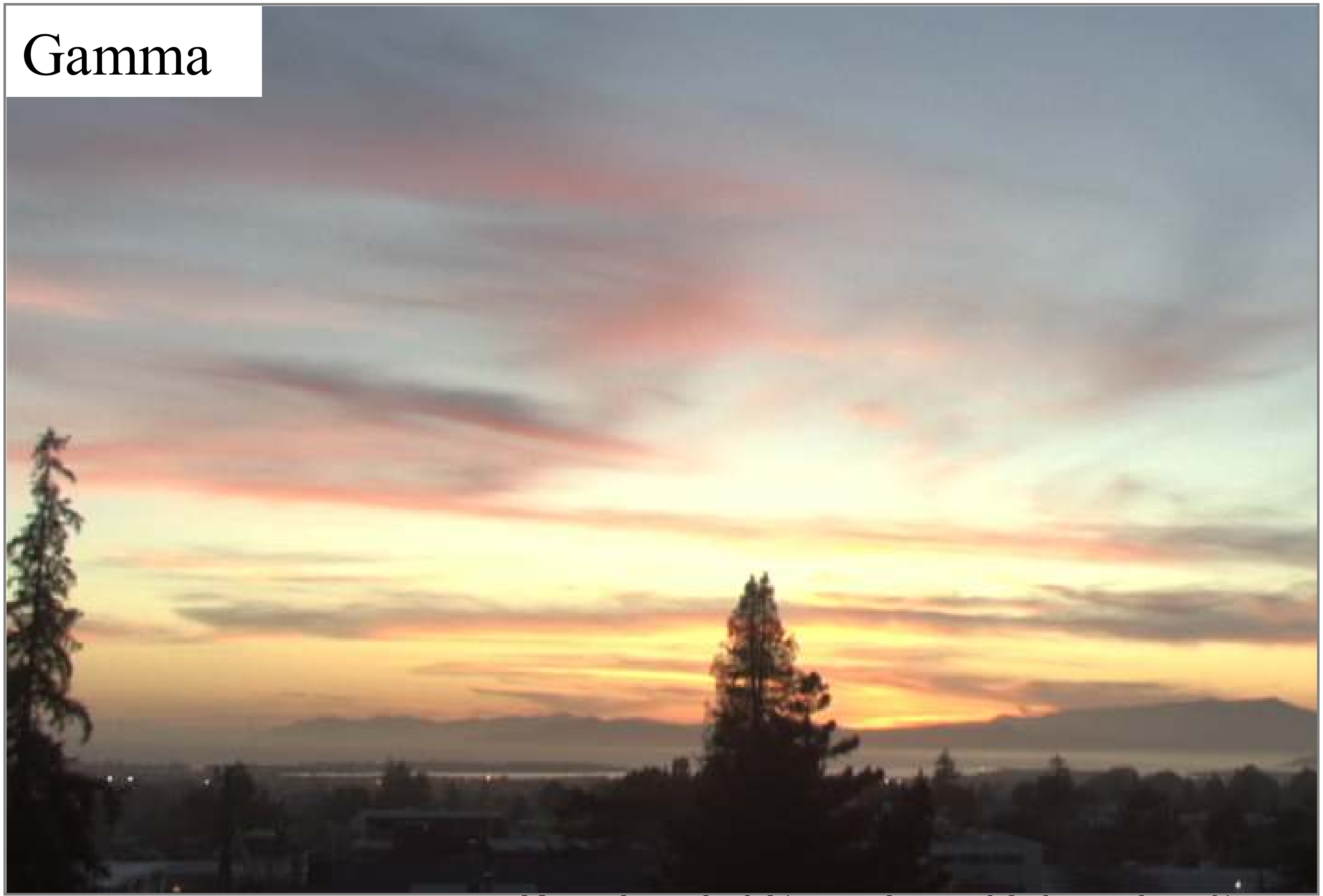
(Source: http://www.ict.usc.edu/graphics/HDRShop/)

A typical photo

- Sun is overexposed
- Foreground is underexposed

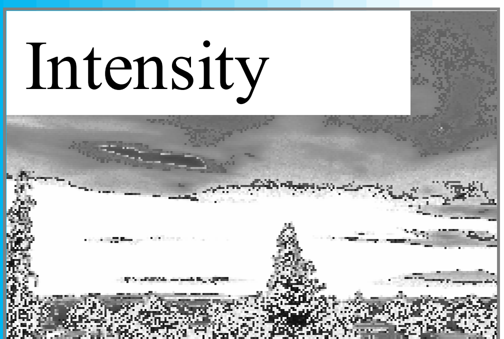
Gamma compression

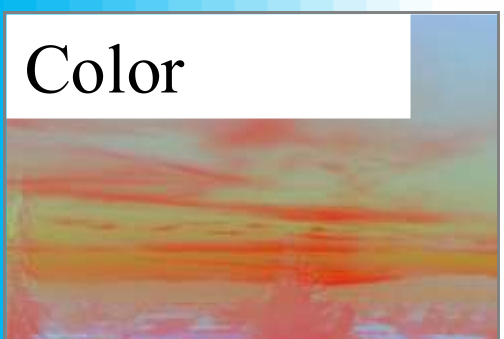
- $x \rightarrow x^{\gamma}$
- **Colors are washed-out**

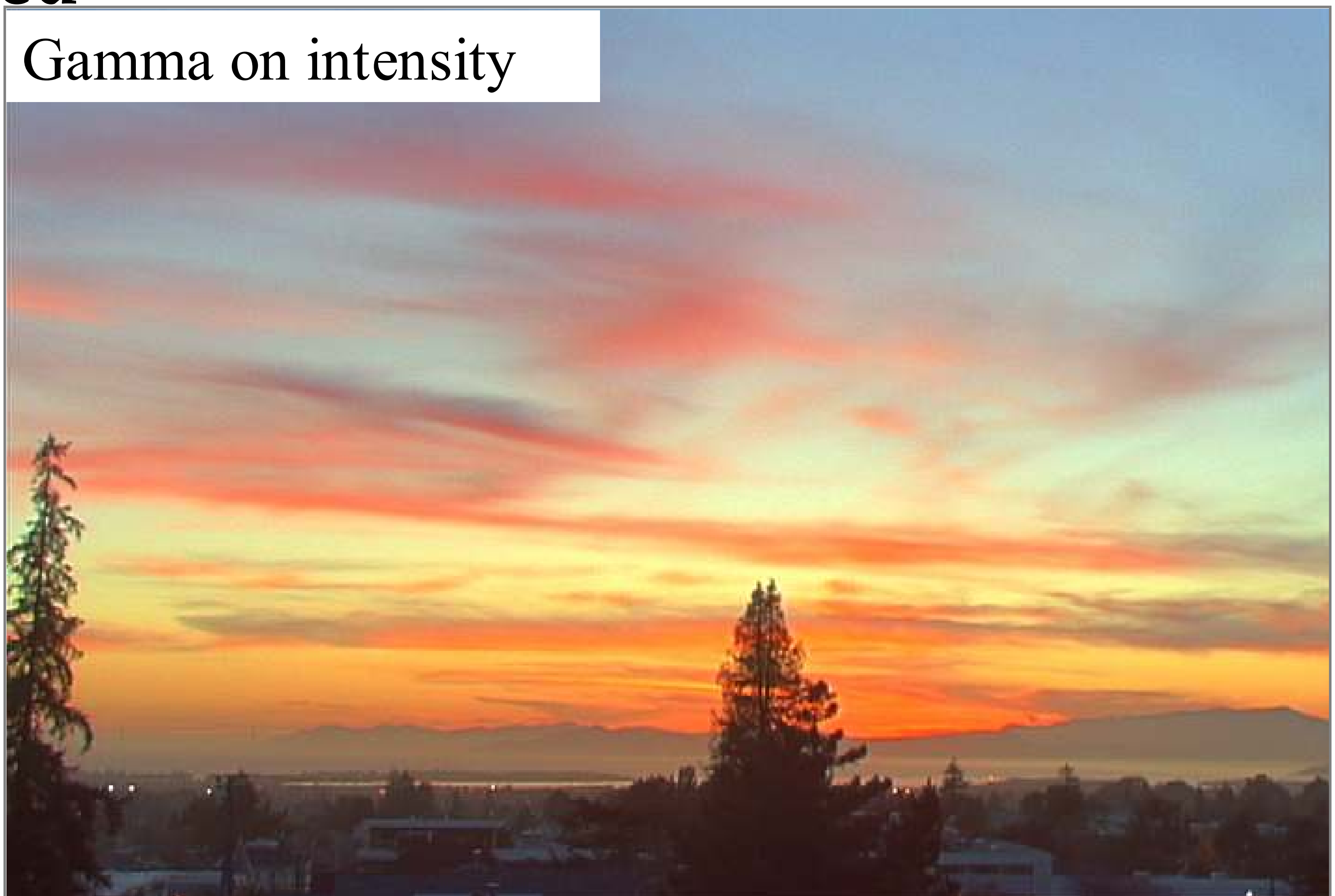


Gamma compression on intensity

- **Colors are OK,**
  but details (intensity high-frequency) are blurred



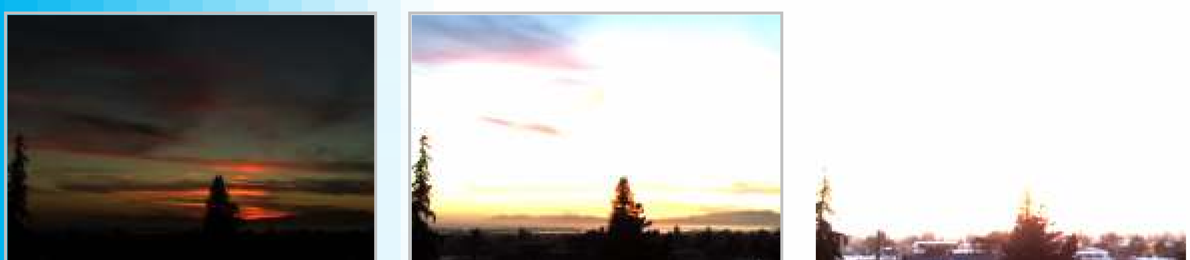




High-dynamic-range (HDR) images

- **CG Images**

- **Multiple exposure photo** **[Debevec & Malik 1997]**



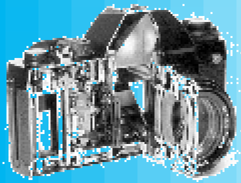
Recover response curve → HDR value for each pixel

- **HDR sensors**

Department of Computer and Information Science

Fundamental Problems

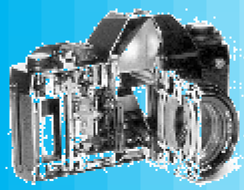
- How to generate a high dynamic range image
  - Debevec & Malik 97 method
- How to display a high dynamic image
  - Tone mapping
  - Bilateral filter
  - Gradient domain fusion
  - Laplacian pyramids



Generating HDR Image

- Using an HDR camera (used to be a dream), but still very expensive
- Image with a series of images combined into a single high dynamic range image (also called a "radiance map")
- They are useful for representing true illumination values in image-based rendering applications
- And for recording incident illumination and using it to illuminate CG objects for realistic composition

(Source:http://www.debevec.org/Research/HDR/)



Ways to Vary Exposure

- Shutter speed
  - Most commonly used
- F/stop (aperture, iris)
  - Problems?
- Neutral Density (ND) Filters
- Gain /ISO / Film Speed

Courtesy of Paul Debevec

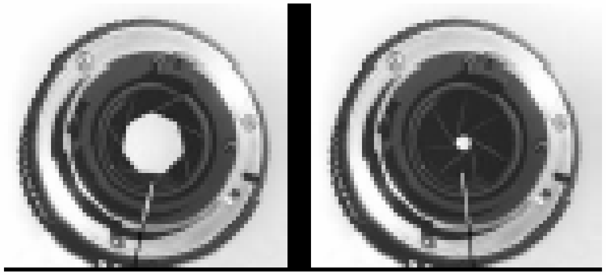
Shutter Speed

- Ranges: vary from camera to camera
  - Canon D30 30 ~ 1/4,000 sec

- Cons
  - Digital: noise in long exposures
  - Film: Reciprocity failure at > ~5 sec

- Pros
  - Directly varies the exposure
  - Usually accurate and repeatable

Aperture

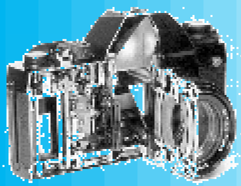


- Ranges:
  - f/2.8 to f/22
- Standard f-numbers: 2, 2.8, 4, 5.6, 8, 11, 16, 22
- Exposure is proportional to the inverse square of the f-number
  - f/22 is 1/64 the light of f/2.8
- Pros
  - Can use aperture when you run out of shutter speed variation

Problem with Aperture

- Changes depth of field
- Not very repeatable
- Limited range of exposure variation
- Not recommended for HDR


Neutral Density (ND) Filters
TIFFEN
Ranges: 0.1 to 4.0 density
(0.3, 0.6, 0.9 density = 1, 2, 3 stops common)
Log base 10 scale:
Density of 0.3 = ½ the light (1 stop)
Density of 1.0 = 1/10 the light
Density of 4.0 = 1/10,000 the light

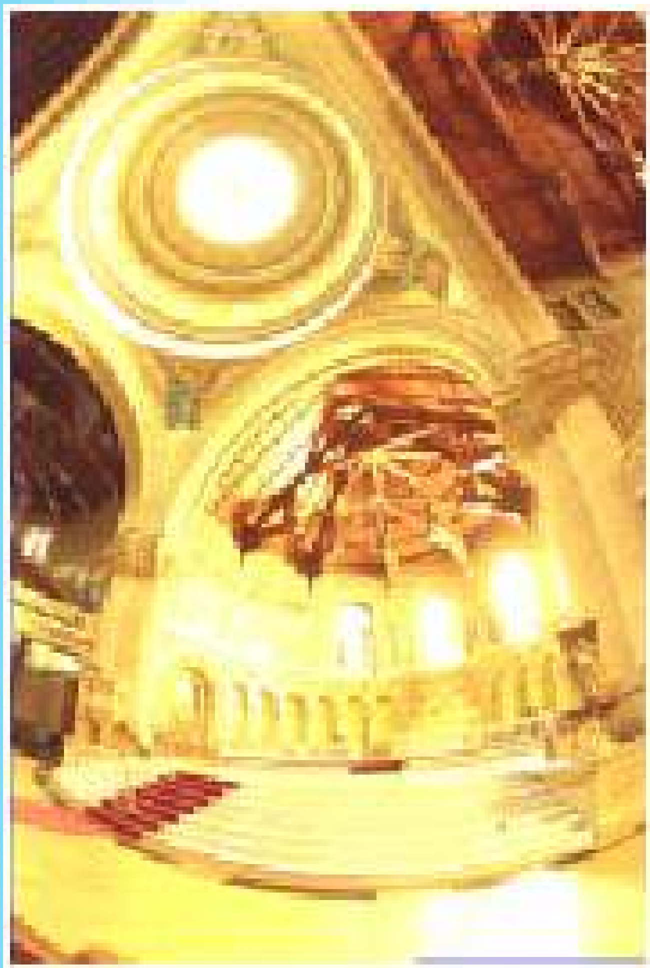


Recovering High Dynamic Range Radiance Maps from Photographs

- Acquiring series of differently exposed photographs
  - Example: sixteen photographs taken at 1-stop increments from 30 seconds to 1/1000 seconds exposure
- Then combine the photos by using HDR Shop

(Source: Recovering High Dynamic Range Radiance Maps from Photographs; Debevec & Malik, 1997)

Limited Dynamic Range



saturated

underexposed

The Main Idea

- How can we cover a wide dynamic range?
- What are we recovering?
- Combine many photographs taken with different exposures!

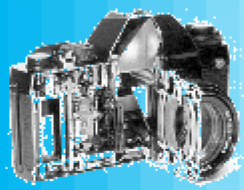
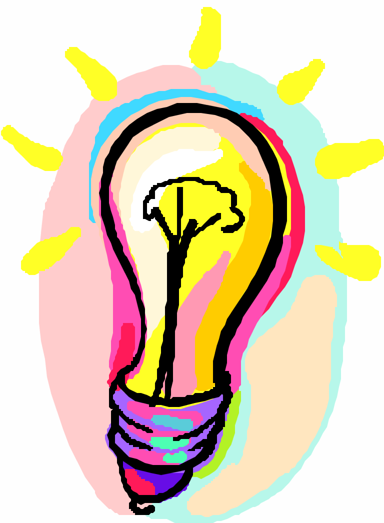
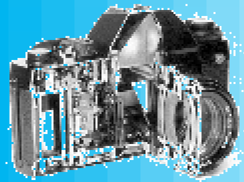


Image Acquisition

- **Pipeline**

  physical scene radiance (L) →
  sensor irradiance (E) →
  sensor exposure (X) →
  { development → scanning → }
  digitization →
  re-mapping digital values →
  final pixel values (Z)





High Dynamic Range Imaging

(Debevec and Malik, SIGGRAPH 1997)

- Problem: Limited dynamic range of film or CCDs makes it impossible to capture high dynamic range in a single image
- Solution: Take multiple images at different exposures
- Problem: How do the pieces get put back together to form a single, composite image
  - Made difficult because mapping from incoming radiance to pixel values is non-linear and poorly documents
- Solution: this paper
  - Very influential for such a simple idea – used in lots of other papers
  - Code is available

Quantities

- The output you see – pixel values – from a scanned film or digital camera, is some function of the scene irradiance:

$$Z = f(X)$$

- *X* is the product of irradiance and exposure time: $X = E\Delta t$
  - Assuming the “principle of reciprocity”: double exposure and halving irradiance gives the same output, and vice versa
- Aim: recover *f* to allow inversion from observed values to scene irradiances
  - Assumption: *f* is monotonic (surely true, or it’s a useless imaging device)

Input

- **A set of images, indexed by *j*, with known exposure times: Δt_j**
- **Call the observed value in image *j* at pixel *i* Z_{ij}**
- **Doing some math gives us an equation involving *f* and E_i:**

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

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

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

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

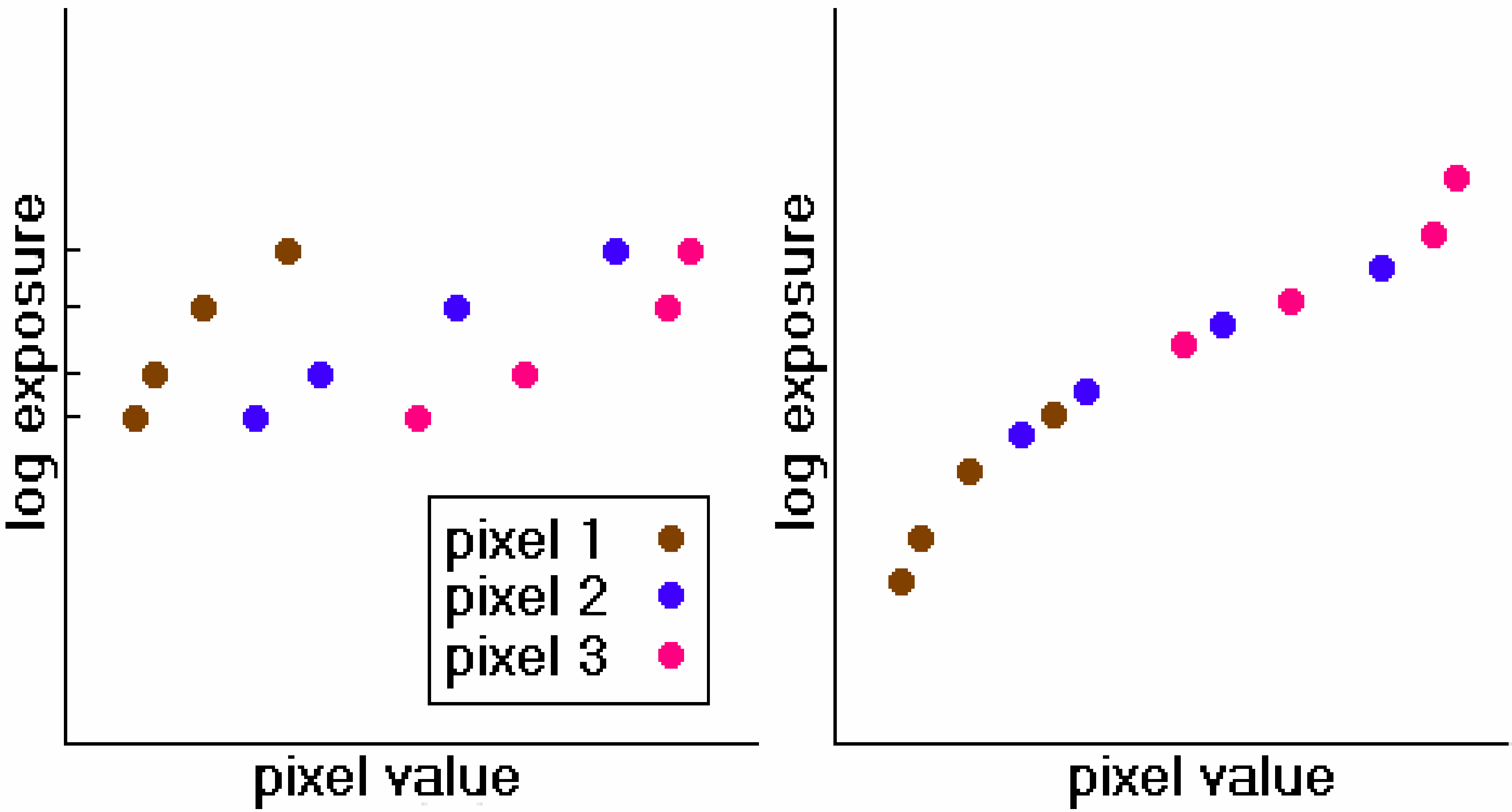
- **We want the *g* and E_i that best represent the given data (the images)**

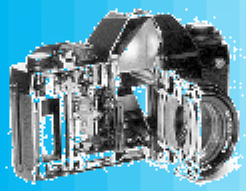
Formulae

- Given $g(Z_{ij}) = \ln E_i + \ln \Delta t_j$
- Find the
  - N values of $\ln E_i$
  - $(Z_{max} - Z_{min} + 1)$ values of $g(z)$
- That minimizes the objective function

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

Picture of the Algorithm





Solution Strategy

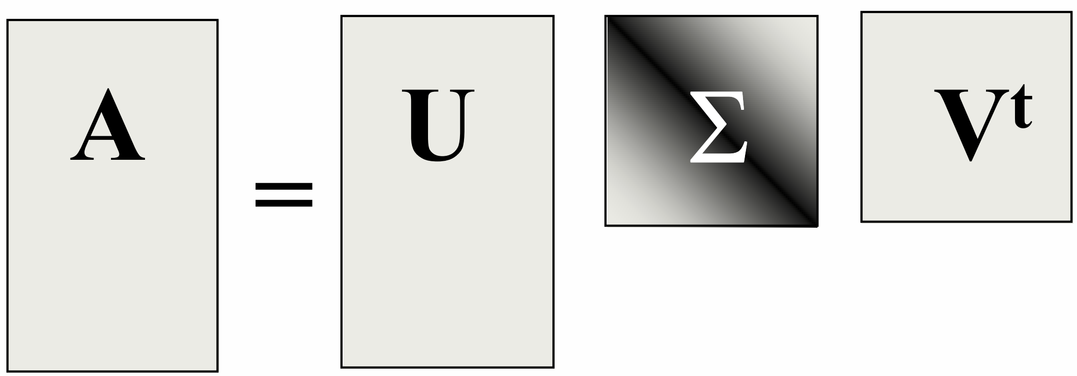
- Minimize
  - Least-squared error
  - Smoothness term
- Exploit discrete, finite world
  - N pixel locations
  - Domain of Z is finite = $(Z_{max} - Z_{min} + 1)$
- Linear least-squares problem (SVD)

What is SVD

- **Let** A **denote an** $m \times n$ **matrix of real data** ($m \geq n$)
- **Then, SVD of** A

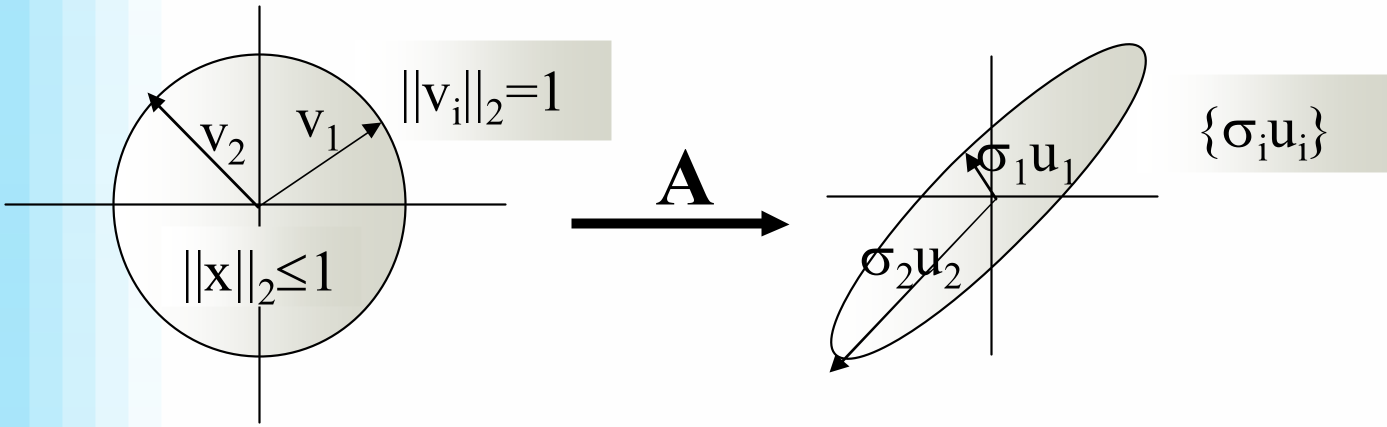
$$\mathbf{A} = \mathbf{U\Sigma} V^T$$

  - U **=** $m \times n$ **matrix; column of** U **= left singular vector,** $\{u_k\}$**, orthonormal**
  - Σ **=** $n \times n$ **matrix; singular values, nonzero on the diagonal**
  - V^T **=** $n \times n$ **matrix; row of** V^T **= right singular vectors,** $\{v_k\}$**, orthonormal**

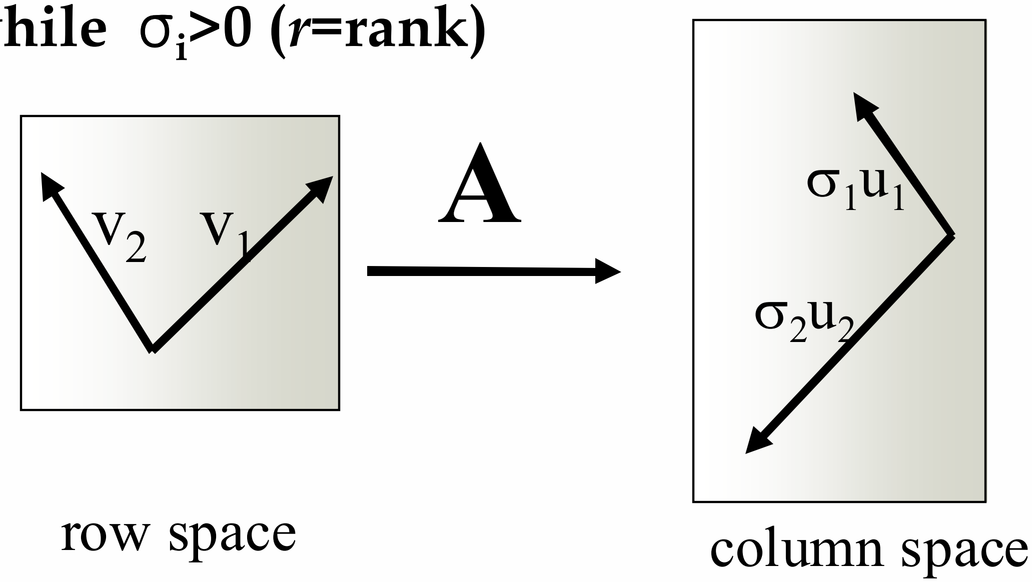


SVD, II

- **Any matrix** A(*m x n*) **maps the unit sphere to a "hyperellipse"**



- **Theorem: for any matrix** A **there are** ***n*-dimensional orthonormal bases** $v_1, \ldots v_r$ **and** ***m*-dimensional** $u_1, \ldots u_r$**, such that** $Av_i = \sigma_i u_i$, **while** $\sigma_i > 0$ **(*r*=rank)**



SVD, III

- **First we find the matrix V:**

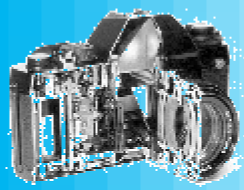
$A^TA=(U\Sigma V^T)^T(U\Sigma V^T)= V\Sigma^T U^TU\Sigma V^T$

$= V\Sigma^T\Sigma V^T$

- **This is an ordinary eigen decomposition of a symmetric matrix**
  - V **is built of eigenvectors of** A^TA**.**
  - **The eigenvectors of** A^TA **are rows of** V^T**.**
- **Find U and** Σ **using use the equations:**

$Av_i=\sigma_i u_i$

Department of Computer and Information Science



SVD, IV

- **Example:** $A = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix}$

- **eigenvectors of** $A^T A$**:** $A^T A = \begin{bmatrix} 2 & -1 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 5 & 3 \\ 3 & 5 \end{bmatrix}$

$(5-\lambda)^2 - 9 = 0;$

$\lambda^2 - 10\lambda + 16 = 0$

$\lambda_1 = 2, \ \lambda_2 = 8$

$$V_1 = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \quad V_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

- **Now, we obtain the** U **and** Σ **:**

$$Av_1 = \sigma_1 u_1 = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0 \\ \sqrt{2} \end{bmatrix} \quad u_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \sigma_1 = \sqrt{2}; \quad Av_2 = \sigma_2 u_2 = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 2\sqrt{2} \\ 0 \end{bmatrix} \quad u_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \sigma_1 = 2\sqrt{2};$$

- $A = U\Sigma V^T$**:** $\begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \sqrt{2} & 0 \\ 0 & 2\sqrt{2} \end{bmatrix}\begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$

SVD, VI

- **Consider the matrix:**

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 2 \end{bmatrix} = \begin{pmatrix} 0 & 2/\sqrt{22} \\ 1/\sqrt{2} & 3/\sqrt{22} \\ -1/\sqrt{2} & 3/\sqrt{22} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{11} \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix}$$

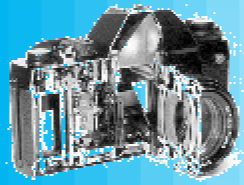
- **and "overdetermined" linear systems:**

$$\mathbf{AX} = \mathbf{b}: \quad \begin{pmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 2.5 \\ 3.5 \end{pmatrix}$$

SVD, VII

- **One way: minimize the error:**
  Least square fitting
  $r=b-Ax \Rightarrow r^2=r^Tr=(b-Ax)^T(b-Ax)=b^Tb-2x^TA^Tb+x^TA^TAx$
  $\therefore A^TAx = A^Tb;$ – **Normal Equations**

- **by SVD (more general)**
  $Ax=b;\ A=U\Sigma V^T \Rightarrow U\Sigma V^T x=b;$
  $\Rightarrow x= V\Sigma^{-1}U^T b$



Getting a Better Fit

- Anticipate the basic shape
  - g(z) is steep and fits poorly at extremes
  - Introduce a weighting function w(z) to emphasize the middle areas
- Define $Z_{mid} = \frac{1}{2}(Z_{min} + Z_{max})$
- Suggested w(z) =

  $z - Z_{min}$ for $z \le Z_{mid}$

  $Z_{max} - z$ for $z > Z_{mid}$

Revised Formulae

- **Given** $g(Z_{ij}) = \ln E_i + \ln \Delta t_j$
- **Minimize the objective function**

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

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

Results – Store Mapping

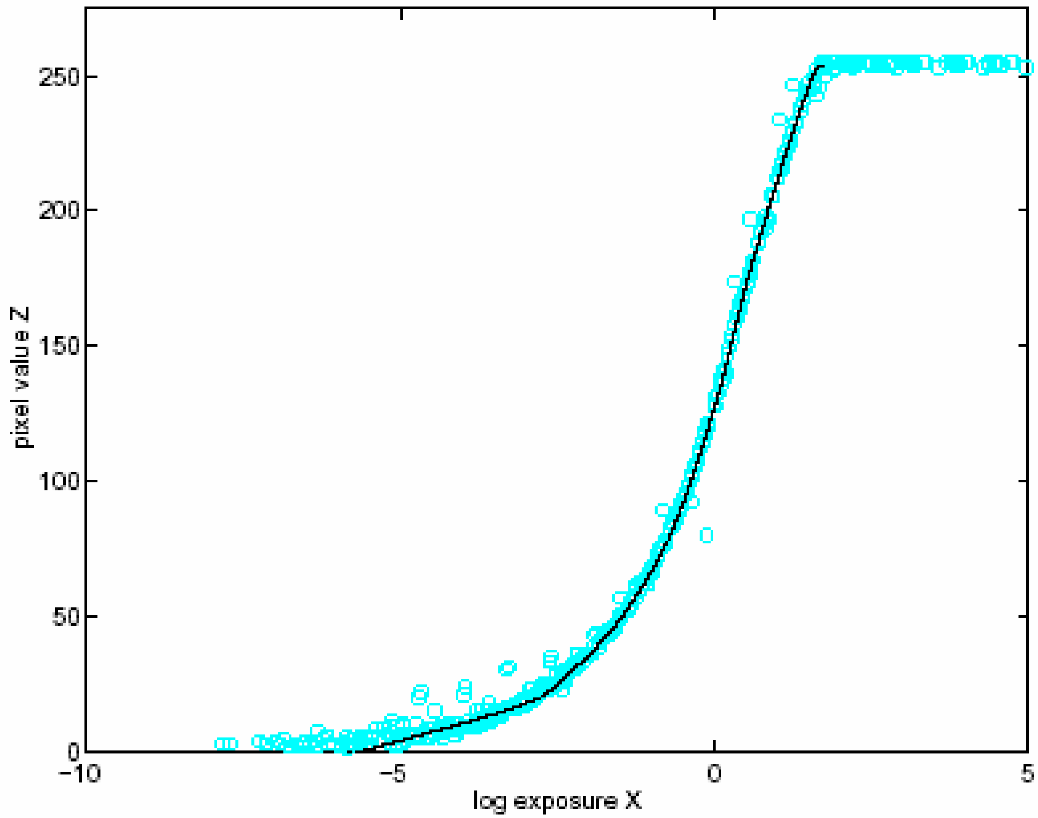


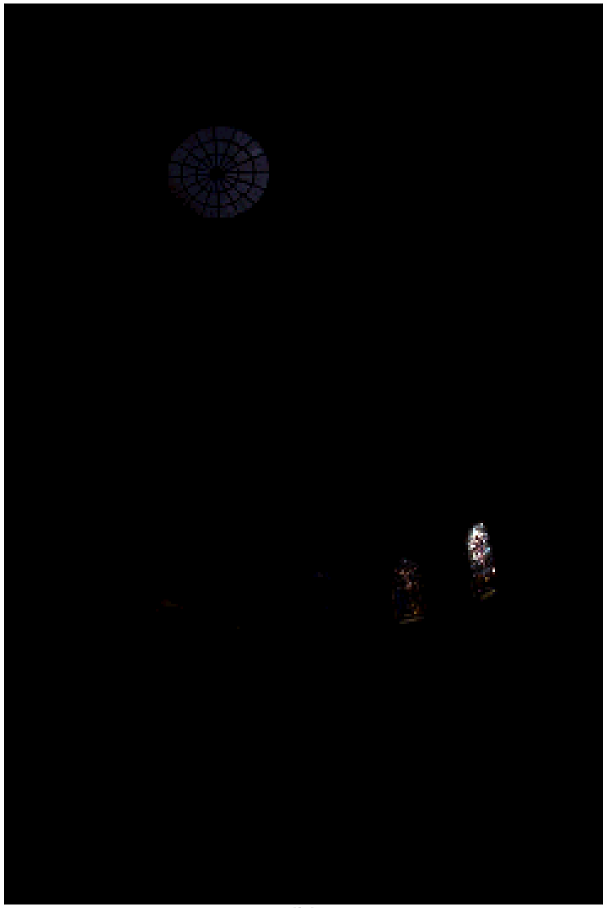
Figure 4: *The response function of the DCS460 recovered by our algorithm, with the underlying* $(E_i \Delta t_j, Z_{ij})$ *data shown as light circles. The logarithm is base* e.

Technicalities

- Only good to some scale factor (logarithms!)
  - Add the extra constraint $Z_{mid} = 0$
  - Or calibrate to a standard luminaire
- Sample a small number of pixels
  - Perhaps N=50
  - Should be evenly distributed from Z
- Smoothness term
  - Approximate g´´ with divided differences
  - Not explicitly enforced that g is monotonic

Results 1

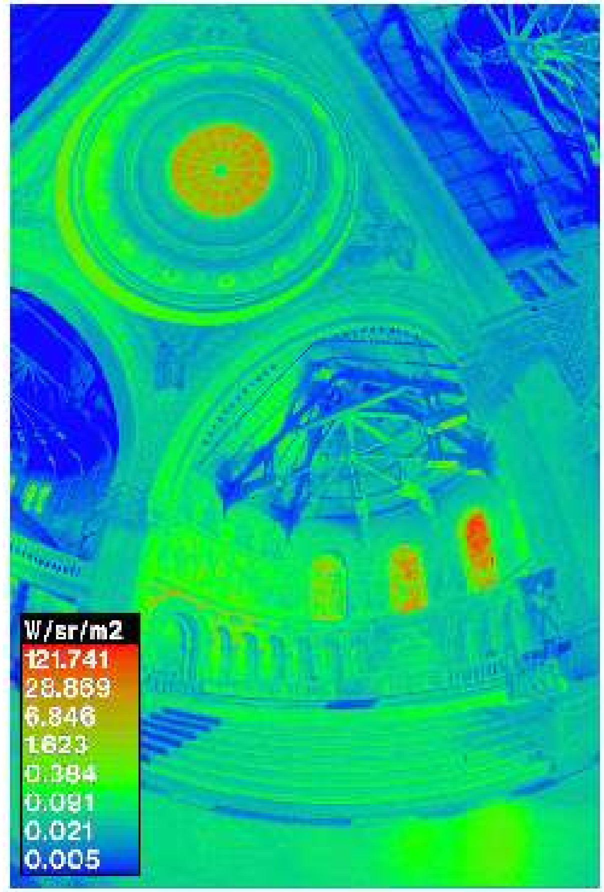
actual photograph
($\Delta t = 2$ s)



radiance map
displayed linearly

Results 2

lower 0.1% of the radiance map (linear)



false color (log) radiance map

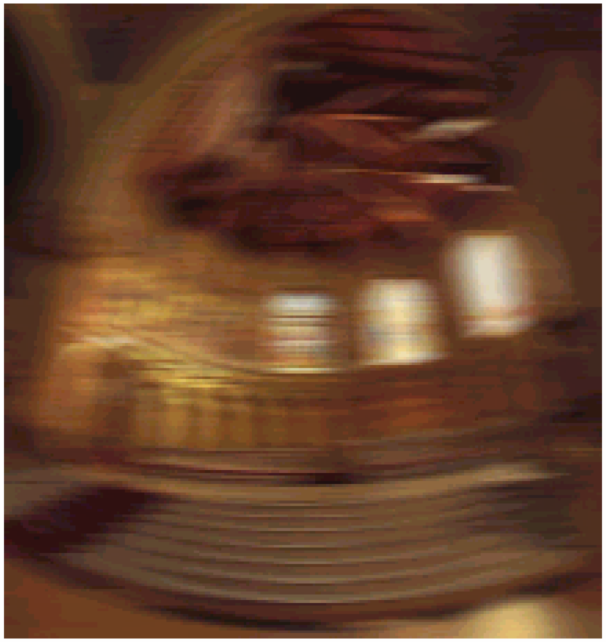
Results 3

histogram compression

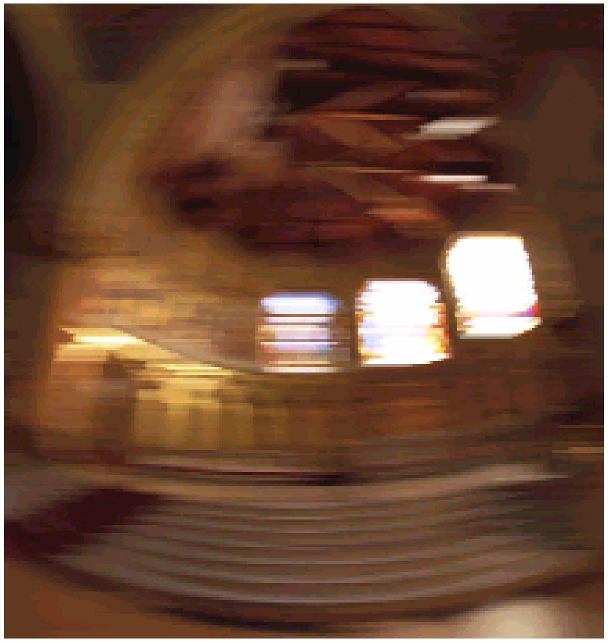
…plus a human perceptual model

Motion Blur

actual blurred photograph



synthetically blurred digital image



synthetically blurred radiance map

[Video]

- **FiatLux (SIGGRAPH'99)**
- **Better image compositing using high dynamic range reflectance maps**