

# CSC320H: Intro to Visual Computing

Instructor: Fernando Flores-Mangas

Office: PT265C

Email: [mangas320@cs.toronto.edu](mailto:mangas320@cs.toronto.edu)

Office Hours: W 11-noon or by appt.

Course WWW (course information sheet available there):

<http://www.cs.toronto.edu/~mangas/teaching/320>

Textbooks:

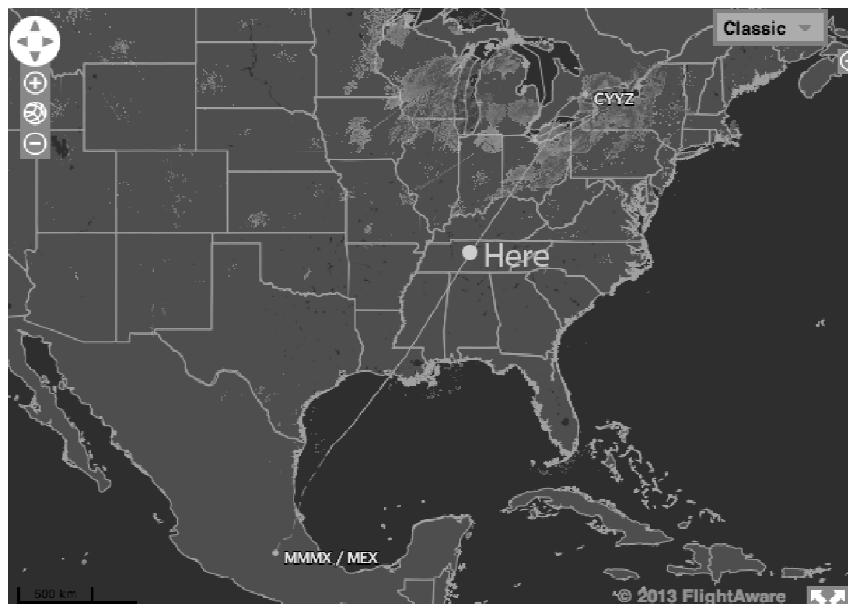
Digital Image Processing,

OpenGL Programming Guide

(both recommended but not required)

Tutorials: Wednesdays at 8 pm (first tutorial next week)

## Fernando is...



He apologizes and will be here next week.

# Today's Topics

0. Admin stuff

1. Intro to Visual Computing

2. Image formation process and high dynamic range photography

## Admin stuff

- Bulletin board (running!):

<https://piazza.com/utoronto.ca/winter2014/csc320/home>  
link available from course website

- Grading

40%: four assignments, worth 10% each, due at 11:59pm on due date.

20%: one midterm test, in tutorial

40%: one final exam

See calendar on course website for dates and other details.

# Admin stuff

- A1 out this Friday via course website
  - Due Jan 24!
  - A1 cannot be done last-minute. Things will go wrong.
    - Read course Info-Sheet for “late policy”
    - While at it, check other assignment-related policies (especially on academic honesty)

# Admin stuff

- Tutorials:
  - Math refreshers and visual programming.
  - Attendance is **STRONGLY** encouraged.
  - Visual programming or assignments will be discussed in the tutorials **ONLY**, not in class.
- First tutorial is next Wednesday (Jan 15)
  - Intro to programming toolkits (useful for A1)
    - fluid, fltk, VXL
  - Refresher on solving linear systems of equations
- Lecture notes: on course website, after class.

# Topic 0

- Intro to Visual Computing
  - What is visual computing?
  - Is this course for me?
  - Course topics

# Topic 0

- Intro to Visual Computing
  - What is visual computing?
  - Is this course for me?
  - Course topics



# What visual computing is NOT

- How do I use Adobe Photoshop© to...



From <http://hollywoodactressbeforephotoshop.blogspot.mx/>

# What visual computing is NOT

- The ultimate gaming video card is...

## Nvidia GeForce GTX 690

[Print/Email](#)

**\$1059.14**



### Pros

The GTX 690 is the fastest graphics card available. It has dual GPUs, extremely high memory bandwidth and great power efficiency.

### Cons

The card is very long, and because there is no native HDMI connection, you have to use a displayport dongle.

### The Verdict: 9.98/10

The GTX 690 gives unparalleled speed and quality to your gaming experience, easily running games on high and ultra settings without breaking a sweat.

From <http://graphics-cards-review.toptenreviews.com/nvidia-geforce-gtx-690-review.html>

# What visual computing actually is!

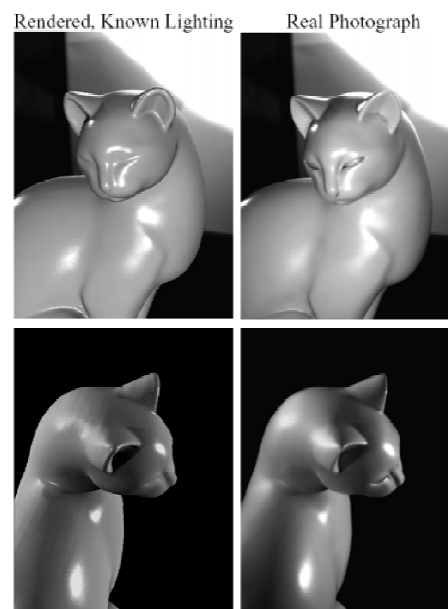
- To use (mainly) geometry- and physics-based models to generate images that mean something to people
- Other disciplines used (and abused)
  - Mathematics
  - Optics
  - Computer Science
  - Engineering
  - Psychophysics
  - Visual Arts



From <http://www.disneyanimation.com/technology/publications#>  
By Stomakhin, et al. at Disney Animation Studios

## Objective #1: Realistic Image Synthesis

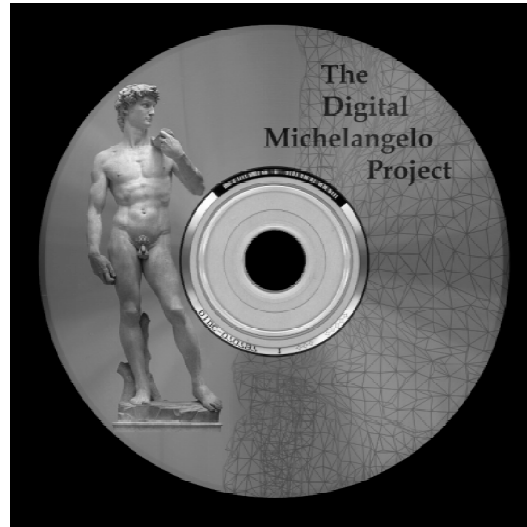
- Create pictures and videos that convey the illusion of reality



(Ramamoorthi et al, SIGGRAPH'01)

## Objective #2: Capturing Reality

- Example: Capturing real scenes in museums, etc



(Levoy et al, SIGGRAPH'00)  
<http://graphics.stanford.edu/projects/mich>

## Objective #3: Manipulating Photos and Videos

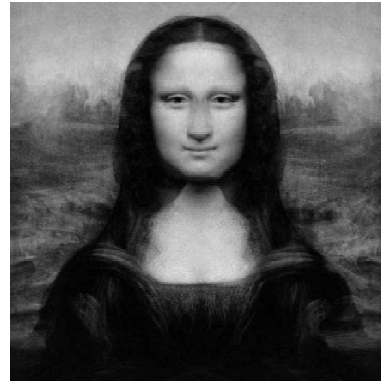
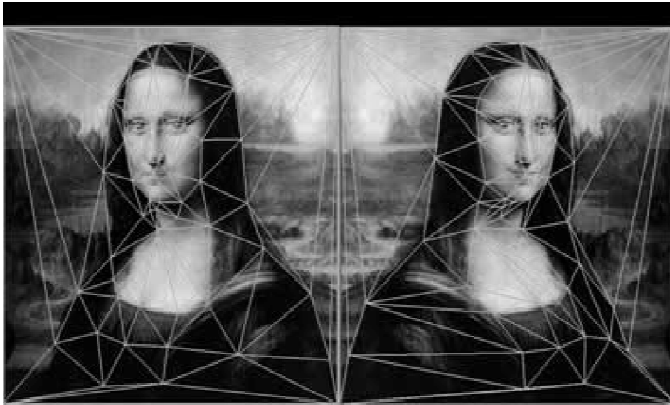
- Manipulate reality (e.g. for special effects)



<http://www.thepixelart.com/>

## Objective #3: Manipulating Photos and Videos

- Manipulate reality (e.g. for special effects)



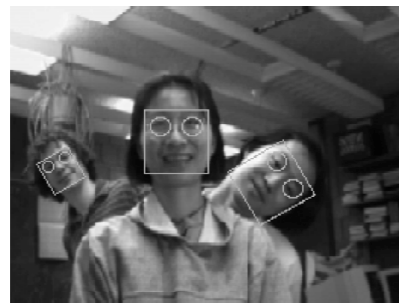
Seitz & Dyer, SIGGRAPH'96

## Objective #4: Photo and Video Interpretation

- Detection, tracking



Shu, Dehghan, Oreifej, Hand, Shah. CVPR 2012

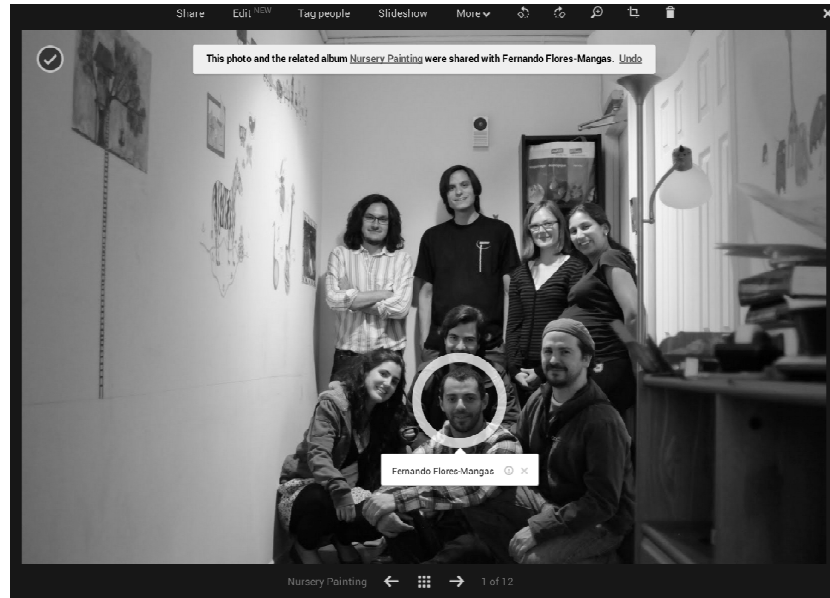


(Rowley et al, PAMI'98)



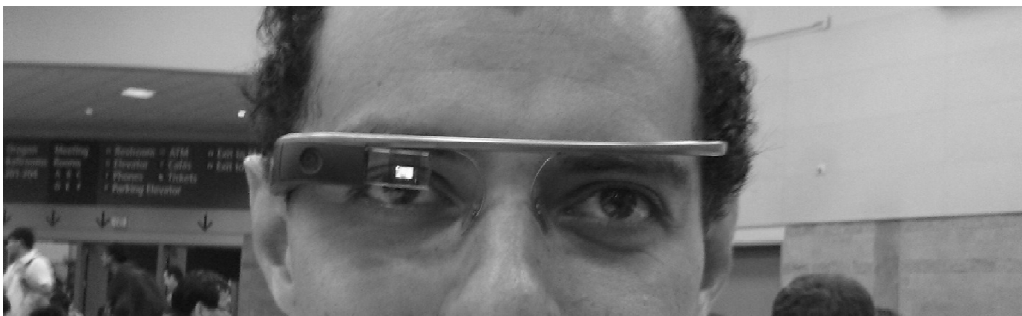
(El-Maraghi et al, CVPR'01)

## Objective #4: Photo and Video Interpretation

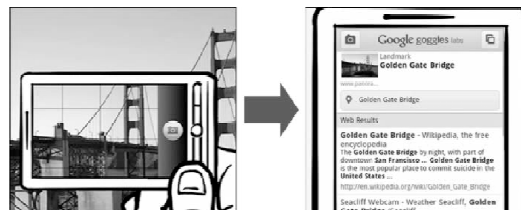
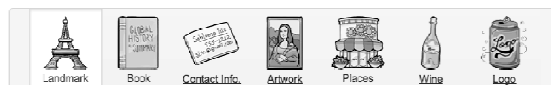


e.g. Face recognition (in Google's Picassa)

## Objective #4: Photo and Video Interpretation



e.g. Automatic object & location recognition with Google Goggles



# Topic 0

- Intro to Visual Computing
  - What is visual computing?
  - Is this course for me?
  - Course topics

## Is this course for me?

### Visual Computing

- Application of the rules of math and physics to generate images that mean something to people
- You must be comfortable with
  - Linear algebra
  - Elementary calculus
- You must be willing to code... a lot!



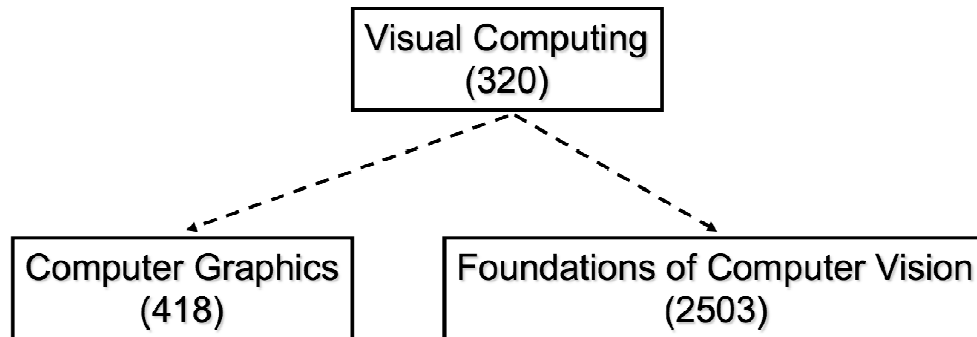
# Assignment 1: Demo



## What you will take away

1. Math drives CS
2. How to use math to create pictures
3. Basics on Image manipulation
4. Intro to coding interactive tools
5. How to read research papers

## Where does this course fit?



- 320 has some thematic overlap with 418 or 2503, BUT the underlying math is the same (i.e. taking 320 first will most definitely help!)

## Where does this course fit?

At the intersection of computer vision and computer graphics because often real photos are the input!

- Video examples
  - Slavenly et al, SIGGRAPH 06
    - Phototourism: Organizing photo collections in 3D



# Phototourism



## Topic 0

- Intro to Visual Computing
  - What is visual computing?
  - Is this course for me?
  - Course topics

# Course Topics

- Principles
  - Computational and mathematical principles for creating, capturing, analyzing and manipulating 2D photos
- Case Studies
  - Applying these principles to the design of specific image manipulation tools (mostly for special effects)
- Visual Programming and Numerical Computing (in tutorials)
  - Learning to use software tools and C++ libraries for graphical user interface design.
  - Implementing math operations on images

## Visual Computing Principles

- Imaging essentials (~2 weeks)
  - Understanding pixel intensity and color
- Image representation and transformation
  - Image  $\Leftrightarrow$  2D array of pixels
  - Image  $\Leftrightarrow$  continuous 2D function (4 weeks)
  - Image  $\Leftrightarrow$  N-Dimensional vector (2.5 weeks)
  - Hierarchical image representations (2 weeks)
  - Image matching and transformation (3 weeks)

# Visual Programming

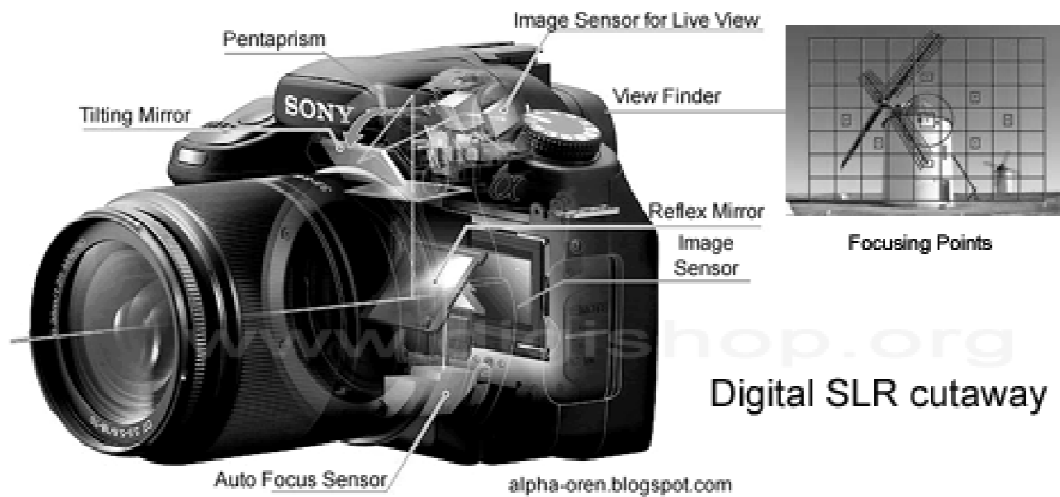
- Basic tools we will use
  - FLUID and fltk GUI toolkit (by Digital Domain)
  - VXL library for image analysis and numerical computing (by a major consortium of computer vision researchers)
- All these tools are under GNU license and are completely portable (Linux/Windows/OSX)

## Topic 1

### Image Formation and High Dynamic Range (HDR) Photography

- Imaging sensors (in grayscale)
- High Dynamic Range photography
- Digital image formation
- The camera response function
- Computing the camera response function

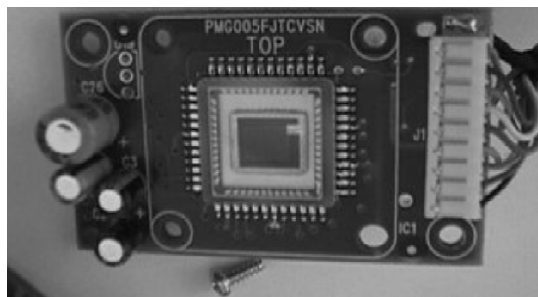
# The Digital Single-Lens Reflex Camera



From <http://www.digishop.org/>

## The Imaging Sensor

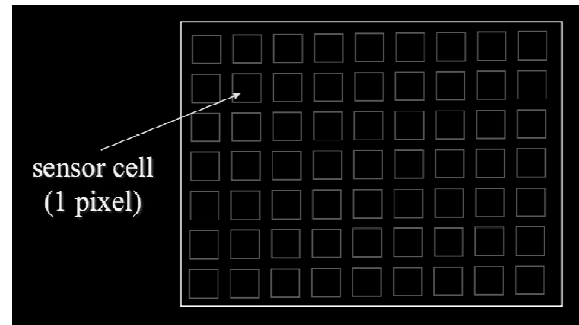
- An array of photo-sensitive cells (usually 2-dimensional), each corresponding to one pixel (picture element)



- Light falling onto a cell induces voltage that depends on the intensity of the incident light
- Voltage is converted to a digital signal within a sensor specific range (in a 8-, 10- or 14-bit number)

# The Imaging Sensor

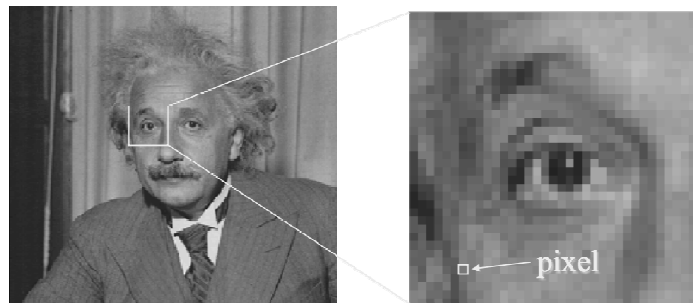
- An array of photo-sensible cells (usually 2-dimensional), each corresponding to one pixel (picture element)



- Light falling onto a cell induces voltage that depends on the intensity of the incident light
- Voltage is converted to a digital signal within a sensor specific range (in a 8-, 10- or 14-bit number)

## What does the value of a pixel mean?

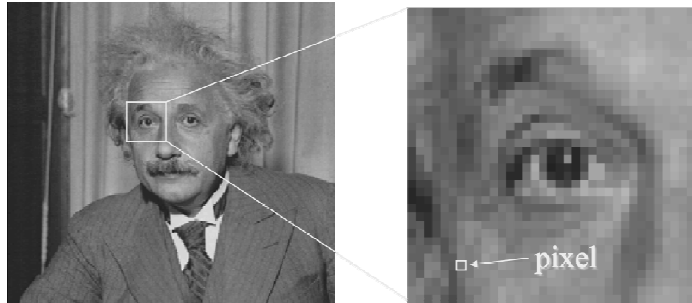
- Pixel values in an image are clearly related to the amount of incoming light, but how exactly?



- And crucially, why do we care about this relation?

# The Imaging Sensor

- An array of photo-sensible cells (usually 2-dimensional), each corresponding to one pixel (picture element)



- Light falling onto a cell induces voltage that depends on the intensity of the incident light
- Voltage is converted to a digital signal within a sensor specific range (in a 8-, 10- or 14-bit number)

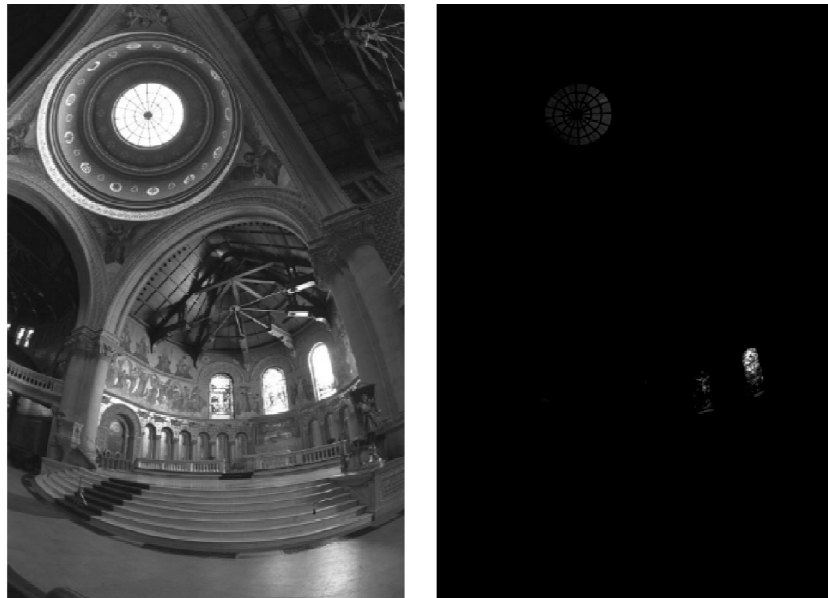
## Topic 1

### Image Formation and High Dynamic Range (HDR) Photography

- Imaging sensors (in grayscale)
- High Dynamic Range photography
- Digital image formation
- The camera response function
- Computing the camera response function

# One major difficulty in photography

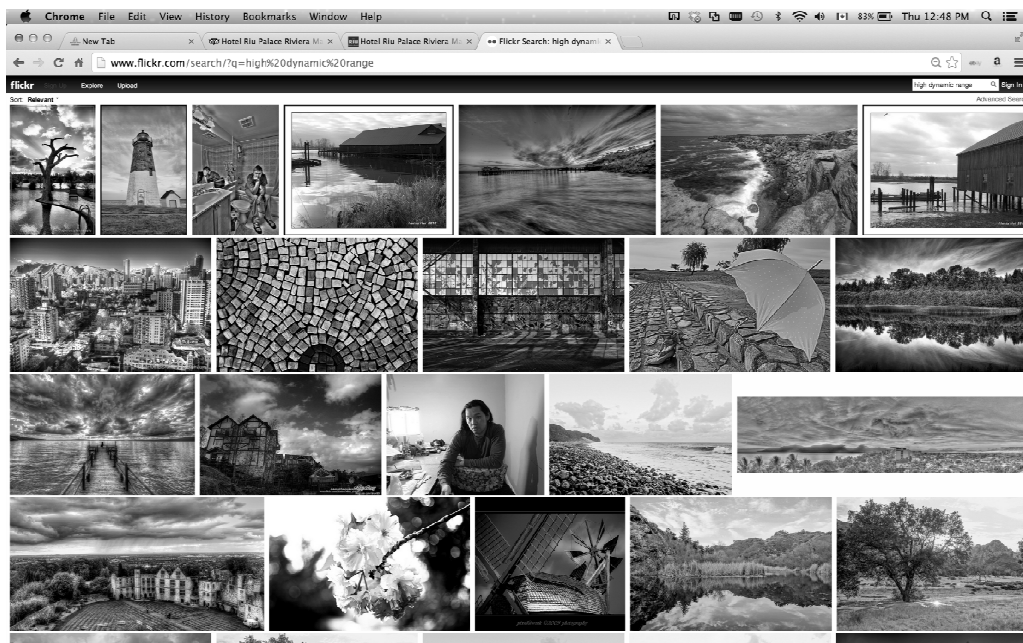
Scenes with very bright and very dark areas are hard to capture



Long- and short-exposure photos (Debevec et al, SIGGRAPH'97)

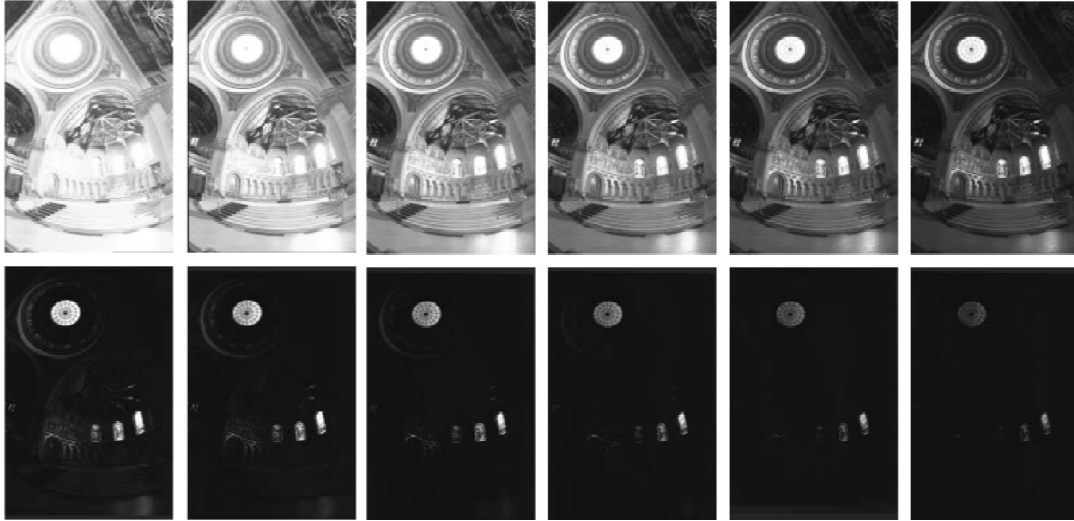
## High-Dynamic Range Photography

- See flickr.com for many examples



# High-Dynamic Range Photography

The 8-bit values at each pixel in all photos are converted to a single floating point value



# High-Dynamic Range Photography

The 8-bit values at each pixel in all photos are converted to a single “HDR” floating point value



(Fattal et al, SIGGRAPH'03)

To do this, we must know how the sensor cells convert light to 8-bit values. Why?

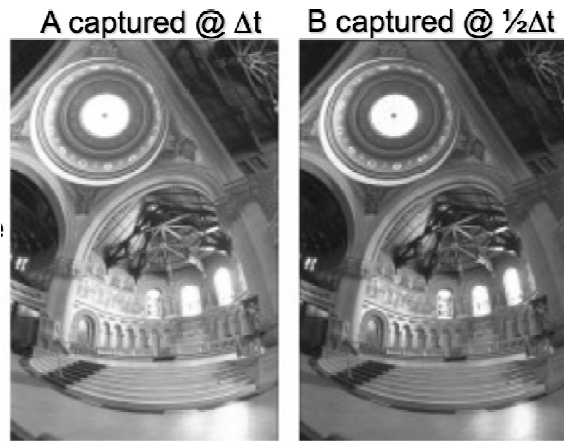


# High-Dynamic Range Photography

- Question:

– Suppose we take two photos A and B with exposure intervals  $\Delta t$  and  $\frac{1}{2}\Delta t$ . The intensity at pixel  $(x,y)$  will satisfy:

- a)  $A(x,y) = 2 B(x,y)$
- b)  $A(x,y) = \frac{1}{2} B(x,y)$
- c)  $A(x,y) = B(x,y)$
- d) none of the above

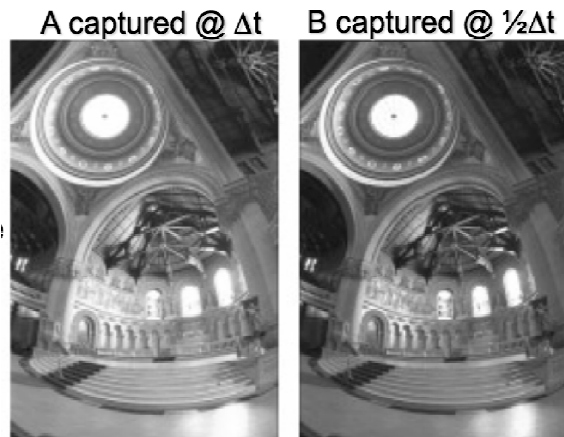


# High-Dynamic Range Photography

- Question:

– Suppose we take two photos A and B with exposure intervals  $\Delta t$  and  $\frac{1}{2}\Delta t$ . The intensity at pixel  $(x,y)$  will satisfy:

- a)  $A(x,y) = 2 B(x,y)$
- b)  $A(x,y) = \frac{1}{2} B(x,y)$
- c)  $A(x,y) = B(x,y)$
- d) none of the above  
(most likely!)

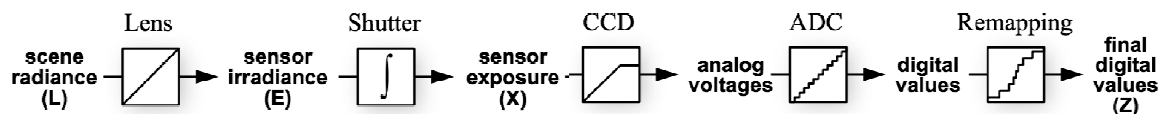


# Topic 1

## Image Formation and High Dynamic Range (HDR) Photography

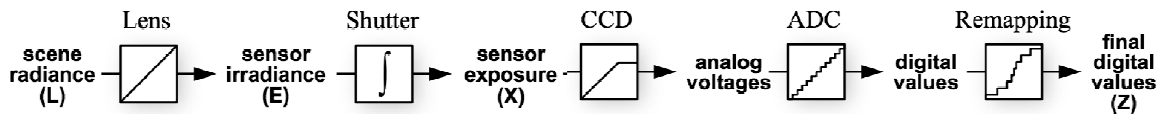
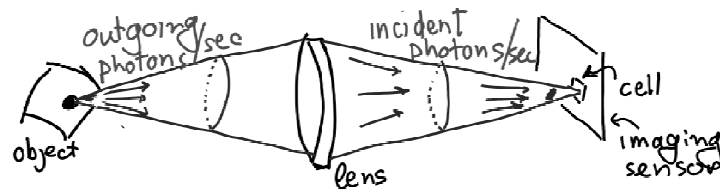
- Imaging sensors (in grayscale)
- High Dynamic Range photography
- Digital image formation
- The camera response function
- Computing the camera response function

### Digital Image Formation: Basic Steps



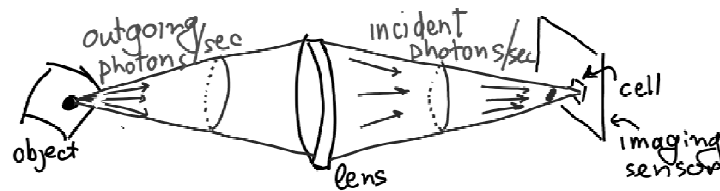
Adapted from Debevec et al, SIGGRAPH 97

# Digital Image Formation: Basic Steps

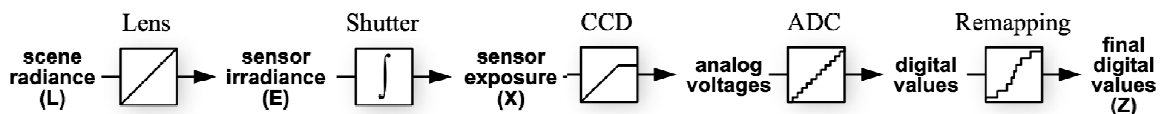


Adapted from Debevec et al, SIGGRAPH 97

# Digital Image Formation: Basic Steps

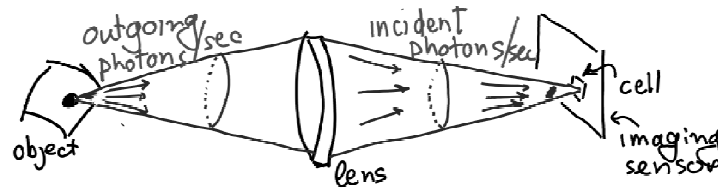


Photons/sec  
received at  
the cell



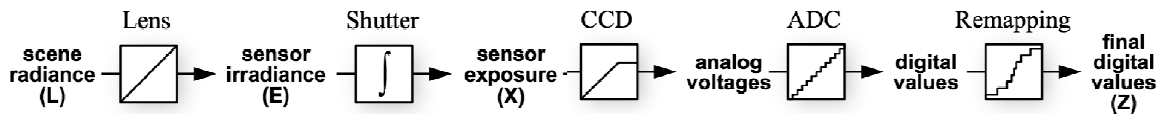
Adapted from Debevec et al, SIGGRAPH 97

# Digital Image Formation: Basic Steps



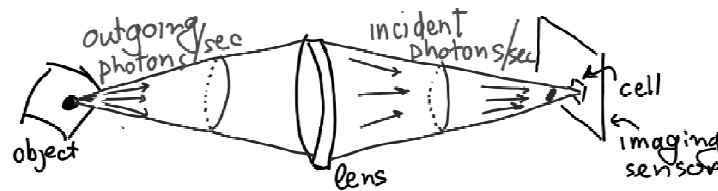
Photons/sec received at the cell

Total photos received at cell during exposure time ( $\Delta t$ )



Adapted from Debevec et al, SIGGRAPH 97

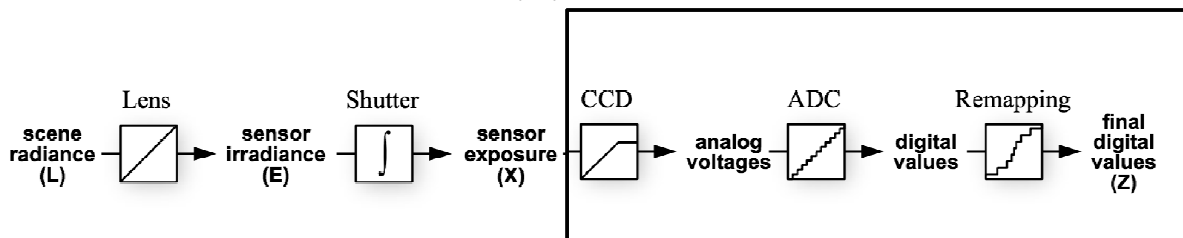
# Digital Image Formation: Basic Steps



Photons/sec received at the cell

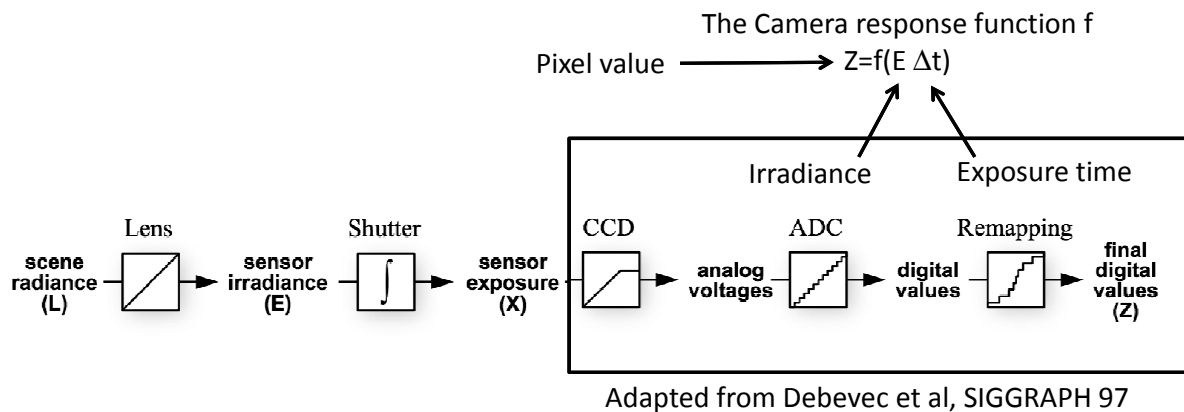
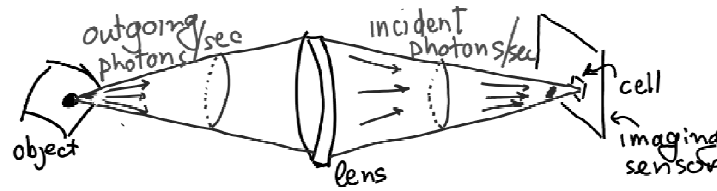
Total photos received at cell during exposure time ( $\Delta t$ )

The Camera response function  $Z=f(E \Delta t)$



Adapted from Debevec et al, SIGGRAPH 97

# Digital Image Formation: Basic Steps

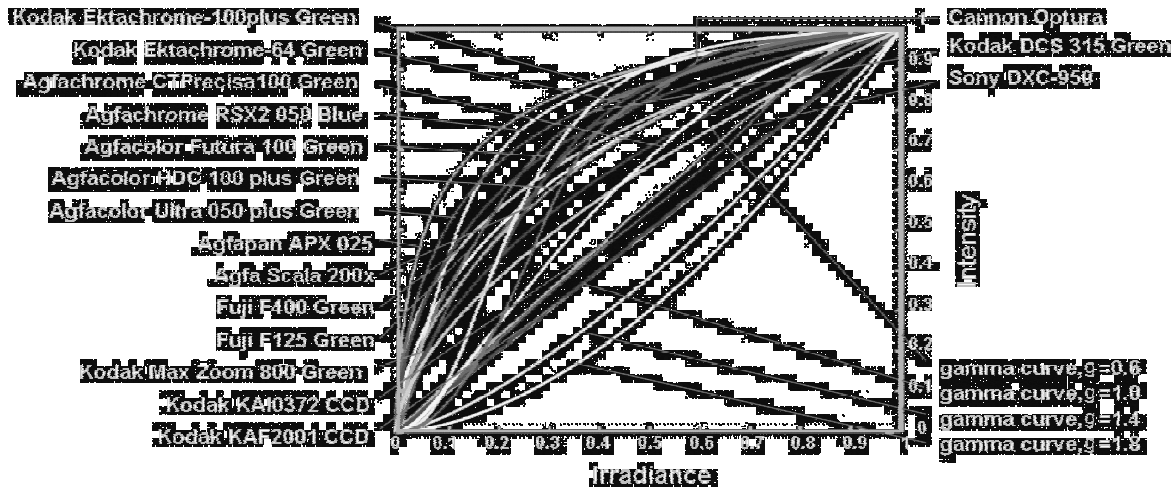


## Topic 1

# Image Formation and High Dynamic Range (HDR) Photography

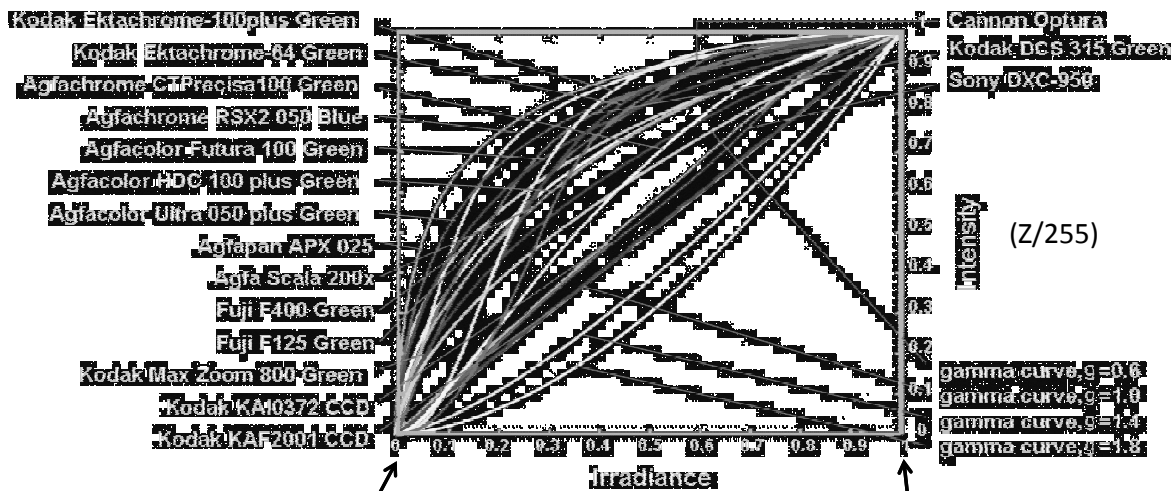
- Imaging sensors (in grayscale)
- High Dynamic Range photography
- Digital image formation
- The camera response function
- Computing the camera response function

# Example Camera Response Functions



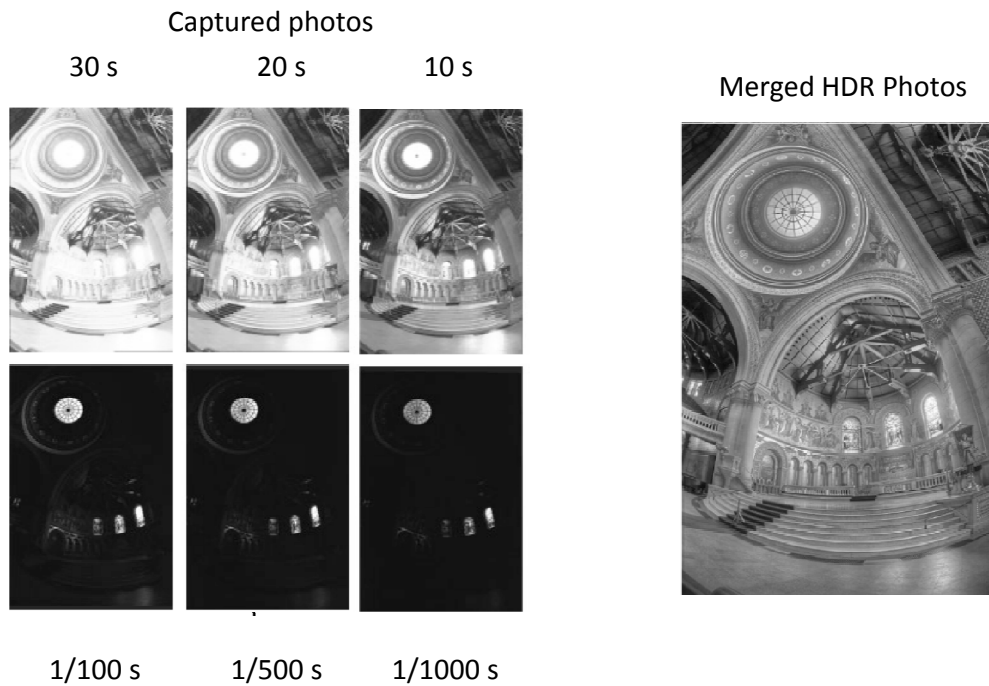
From Grossberg & Nayar, CVPR 2003

# Example Camera Response Functions

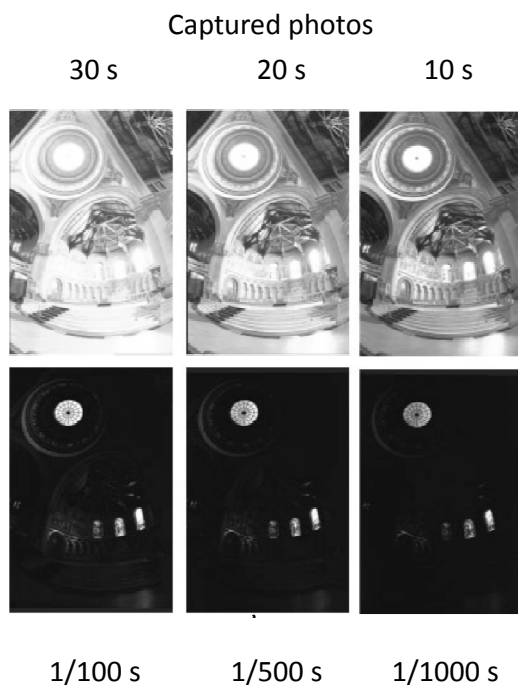


From Grossberg & Nayar, CVPR 2003

# Application: High-Dynamic Range Photography



# Application: High-Dynamic Range Photography

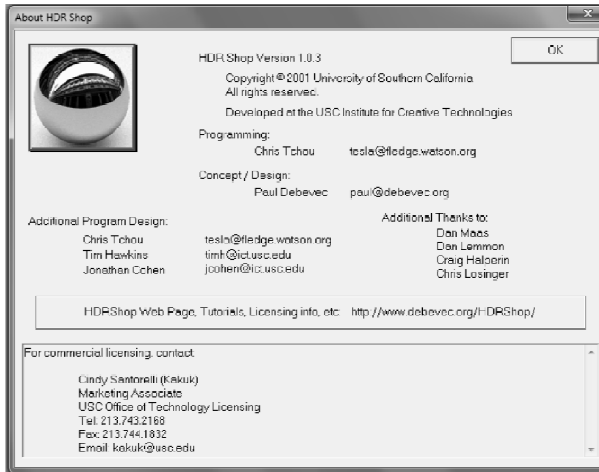


Basic merging procedure:

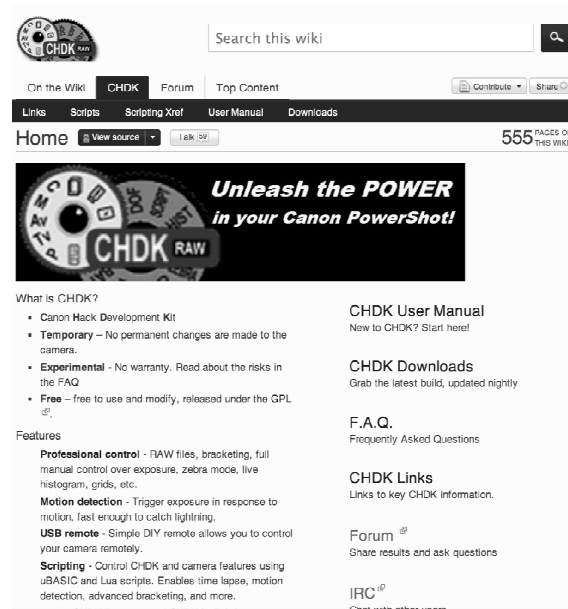
For each pixel  $(x,y)$   
For each photo  $j$   
if  $(x,y)$  is not saturated in  $j$   
Convert pixel intensity  
 $Z_j(x,y)$  to linear  
irradiance  
measurement  $E_j(x,y)$

Merge  $E_1(x,y), E_2(x,y), \dots$  into one  
floating point value

# High-Dynamic Range Photography



Computing Response functions and HDR photos



Hacked camera firmware for Canon Powershots

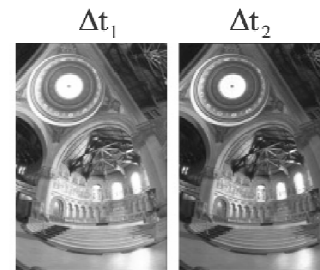
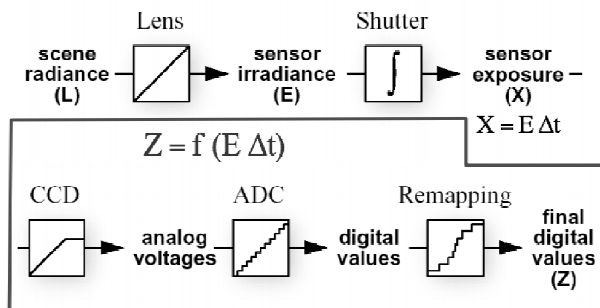
## Topic 1

# Image Formation and High Dynamic Range (HDR) Photography

- Imaging sensors (in grayscale)
- High Dynamic Range photography
- Digital image formation
- The camera response function
- Computing the camera response function



# Computing The Camera Response Function

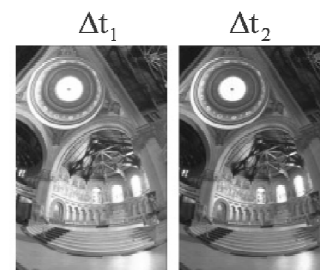
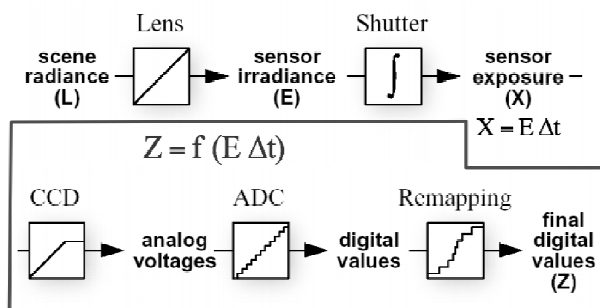


General Procedure:

1. Collect photos for several exposure intervals  $\Delta t_1, \Delta t_2, \dots$  without moving the camera
2. Process photos to compute  $f$

Problem:  
For a given photo we know  $\Delta t_j$  and  $Z$  but we have no way of measuring  $E$

# Computing The Camera Response Function



Idea #1: Invert and use logs

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

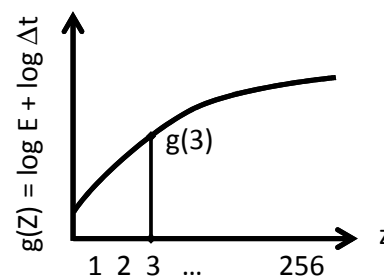
$$f^{-1}(Z) = E \Delta t$$

$$\log f^{-1}(Z) = \log(E) + \log \Delta t$$

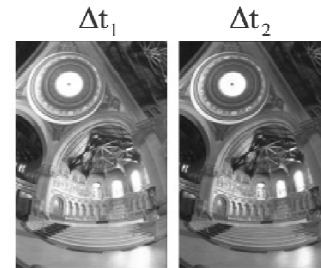
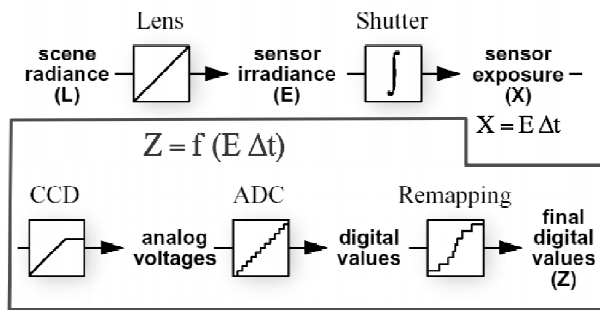
Log-Inverse response fn:

$$g(Z) = \log(E) + \log(\Delta t)$$

The log-inverse response function

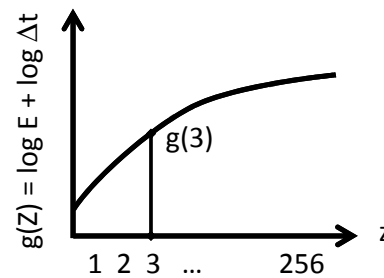


# Computing The Camera Response Function

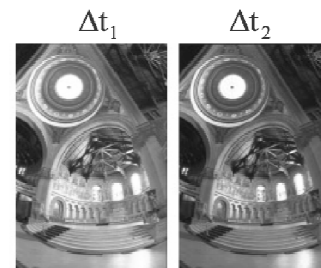
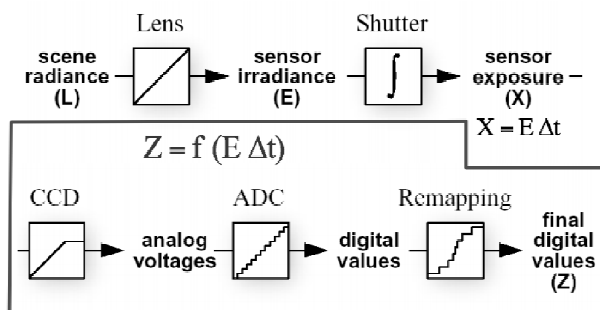


Idea #2: How many quantities must we observe to fully determine  $g()$ ?

The log-inverse response function

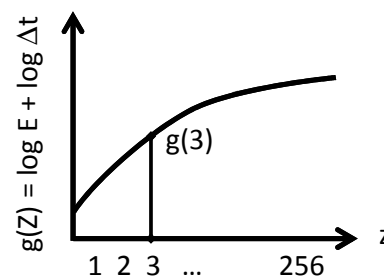


# Computing The Camera Response Function



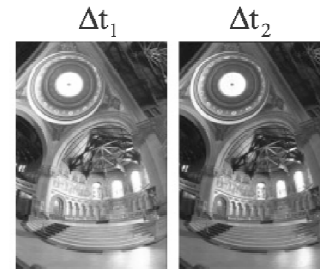
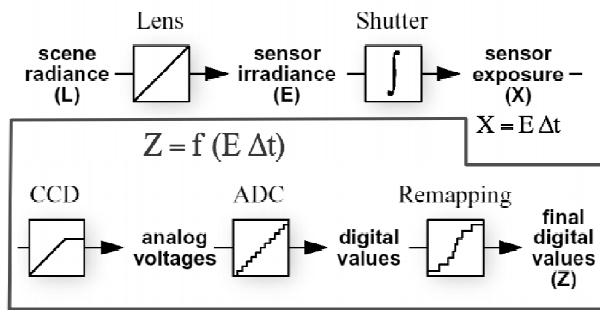
Idea #2: How many quantities must we observe to fully determine  $g()$ ?

The log-inverse response function

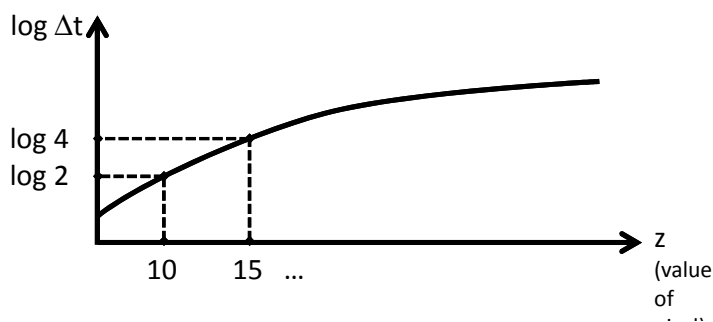


Ans: 256 ( $g(0), g(1), \dots, g(256)$ )

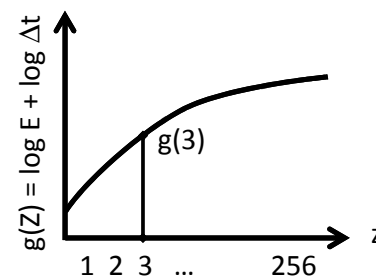
# Computing The Camera Response Function



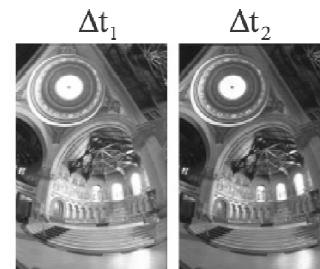
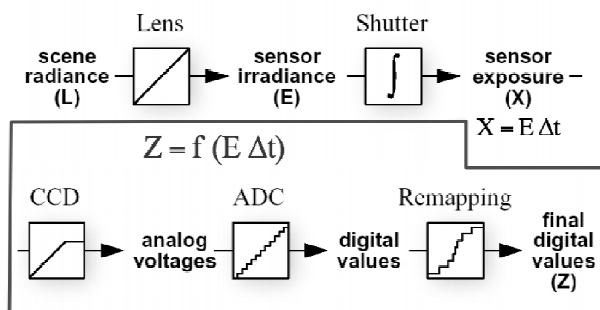
## Approach #1: One pixel, many images



## The log-inverse response function



# Computing The Camera Response Function

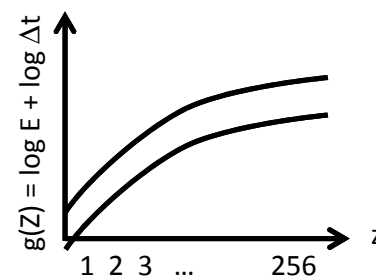


## Approach #1: One pixel, many images

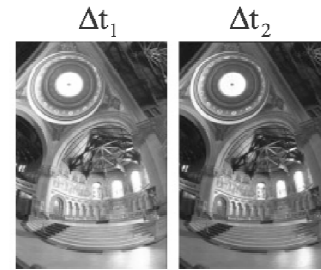
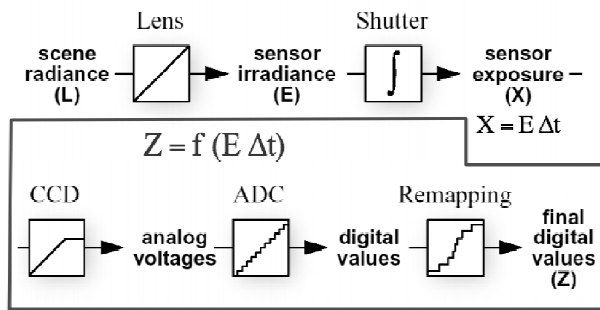
1. Finely adjust  $\Delta t$  in range (1/1000s, 30 s)
2. Plot  $\log \Delta t$  as a function of the pixels' observed intensity.

But  $\log E$  is unknown! Does it matter?

## The log-inverse response function



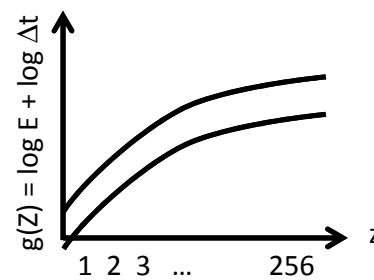
# Computing The Camera Response Function



## Approach #1: One pixel, many images

- What are the problems with this approach?
  - Need lots of images
  - Wasted pixels (we use only one out of thousands in the photo)
  - Possible  $\Delta t$  values are often determined by the camera

## The log-inverse response function



# Computing The Camera Response Function

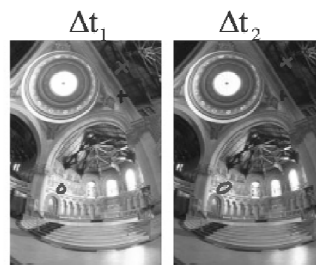
## Approach #2: Few pixels, few images?

different for each pixel

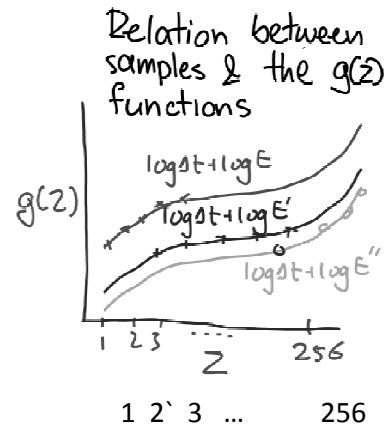
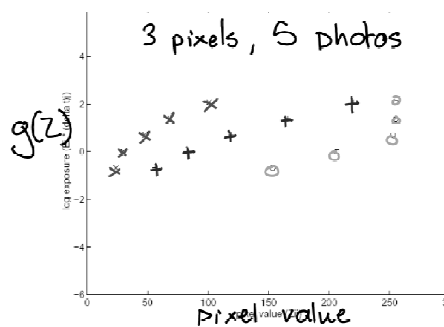
$$g(Z) = \log E + \log \Delta t$$

Same for all pixels

Same for all pixels in a photo

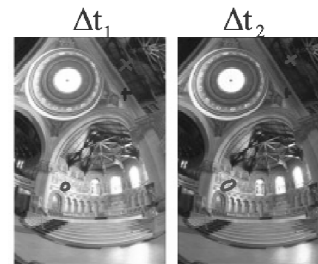


Samples of  $g(Z)$  for multiple pixels and photos

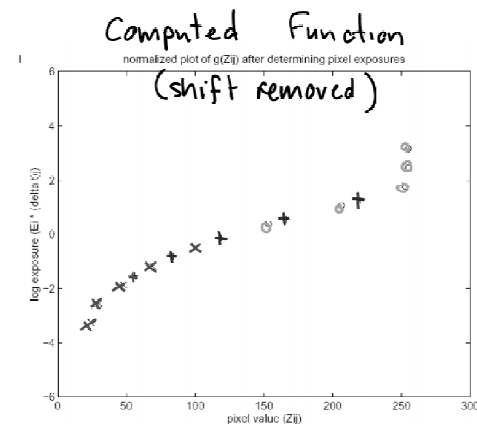
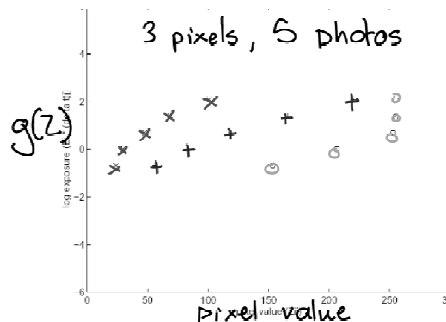


# Computing The Camera Response Function

Approach #2: Few pixels, few images?  
 To compute the complete function we must estimate the relative vertical shift of the  $g(Z)$  function from individual pixels



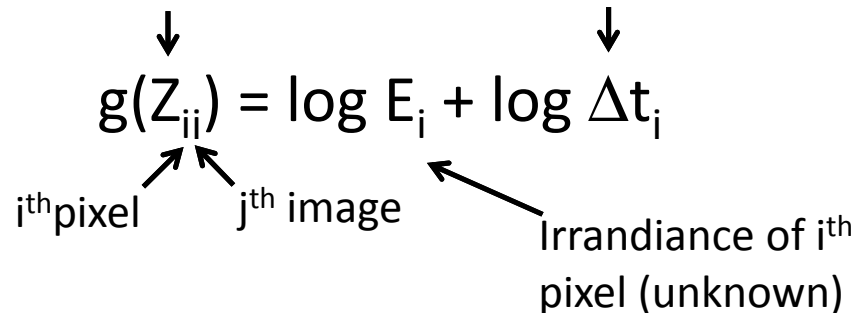
Samples of  $g(Z)$  for multiple pixels and photos



# Computing The Camera Response Function

Approach #2: Few pixels, few images?

observed intensity (known)      exposure interval (known)



Goal:

Compute  $g(0), g(1), \dots, g(255)$  and  $\log E_i$

Given  $N$  pixel intensities in  $P$  images with known  $\Delta t_j$

## Computing The Camera Response Function

Simplifying the notation:  $g(Z_{ij}) = \log E_i + \log \Delta t_j$

becomes  $g_{Z_{ij}} - e_i = \delta_j$

Example: Pixel 100 in the 5<sup>th</sup> photo has intensity 125, denoted as  $Z_{100,5}=125$ . The associated equation is:

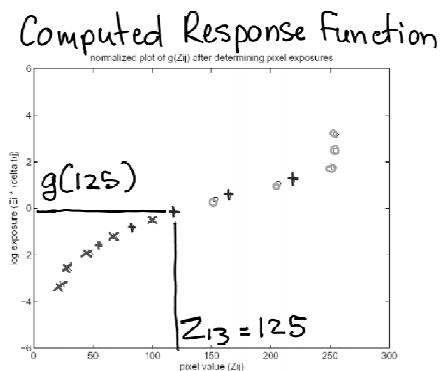
$$g_{125} - e_{100} = \delta_5$$

## Computing The Camera Response Function

Approach #2: Few pixels, few images?

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

We know the above equation is true for all pixels and for all images.



This means we have  $N \cdot P$  equations with  $N+256$  unknowns (one equation per pixel per photo)