
# Recovering High Dynamic Range Multispectral Radiance Maps

Paul Rosen

Department of Computer Science

# Motivation


- Existing high dynamic range radiance recovery techniques recalculate radiance maps for every channel of color
- Use the known scaling factor between color channels, the spectral response, to calculate a per pixel spectral function

# Review – Radiance Maps


- Cameras have limited dynamic range when taking photographs
- Radiance maps are generated by taking multiple photographs of a scene with increasing exposure times
- [Debevec96] Generated radiance maps from photographs on a per color channel basis

# Image Acquisition Pipeline

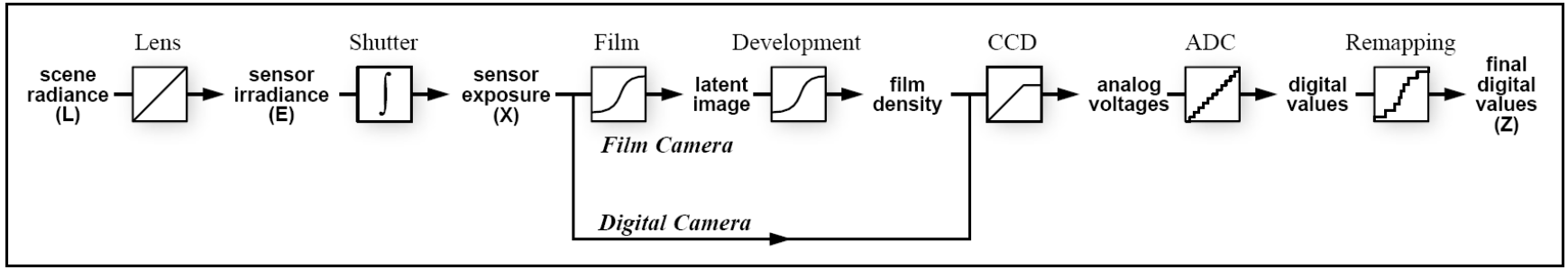



Image Courtesy of Paul Debevec

- Unknown nonlinear mapping in exposure, development, scanning, digitization, and remapping.
- Limited dynamic range in processing

# Review – Color Spectrum

- In graphics we think in primary colors: red, green, and blue
- The visible spectrum ranges in wavelength from ~400nm - ~700nm

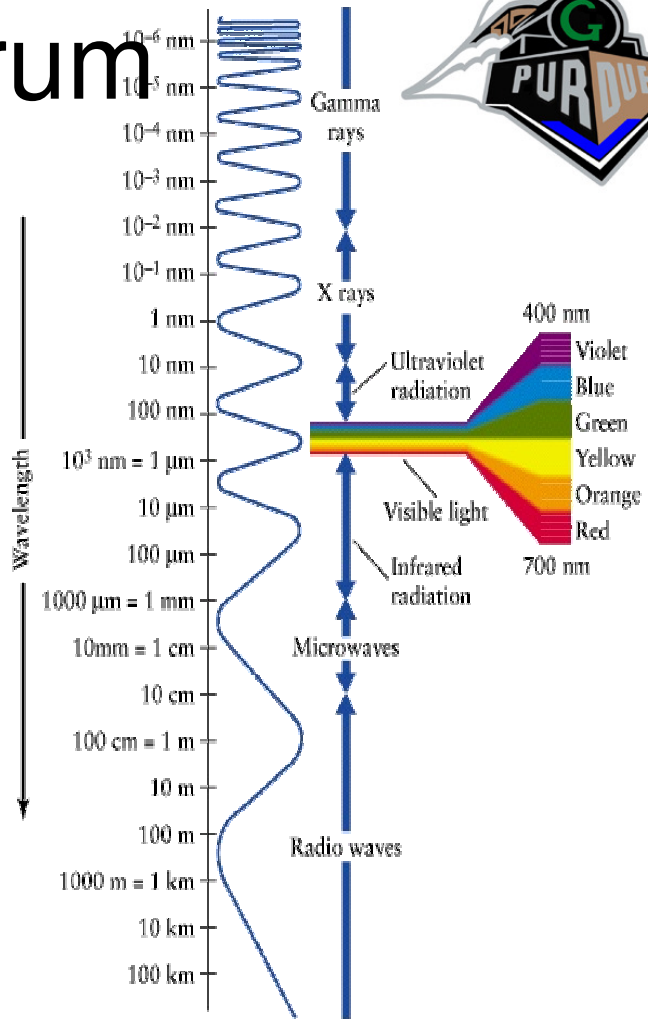
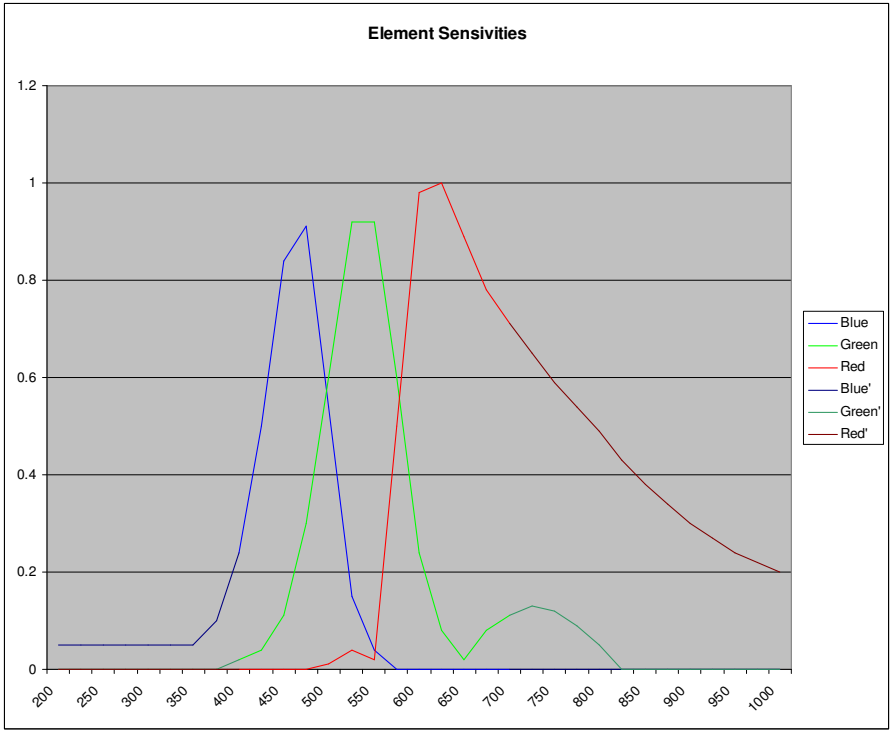



Image Courtesy the *Universe* by Freedman and Kaufmann.

# Overview

- Exploit the overlapping spectral response of CCD elements during the creation of radiance maps



# Reciprocity Equation


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

- A pixel will have the same value after doubling the energy, halving the exposure time
- From reciprocity
  - $Z_{ij}$ - Pixel Value
  - $E_i$ - Film Irradiance
  - $\Delta t_i$ - Exposure Time
  - f() - Non-linear transforms caused by film/signal processing

# Reciprocity Equation


- Reciprocity as defined in Debevec 96 (2)
- Irradiance breaks into an integration of filtered light times scene irradiance over all wavelengths (3)
- Approximate the integration and substitute back into reciprocity equation (4)
- Take the inverse of the non-linear mapping function (5)

$$Z_{ijk} = f(E_{ik}\Delta t_j) \quad (2)$$

$$E_{ik} = \int Filter_k(\lambda)\, Irradiance_i(\lambda) \quad d\lambda \quad (3)$$

$$Z_{ijk} = f(\sum H_k(\lambda) L_i(\lambda) \Delta t_j) \quad (4)$$

$$f^{-1}(Z_{ijk}) = \sum H_k(\lambda) L_i(\lambda) \Delta t_j \quad (5)$$

# Minimization


$$O = \sum_{j=1}^{P} [f^{-1}(Z_{ijk}) - (\sum H_k(\lambda) L_i(\lambda) \Delta t_j)]^2 \quad (6)$$

- Because both $f^{-1}()$ and $L_i()$ are unknown, if SVD is used solution will be all zeros
- Borrow the formulation of $f^{-1}()$ from Debevec96, then this function can be safely minimized

# Final Equation

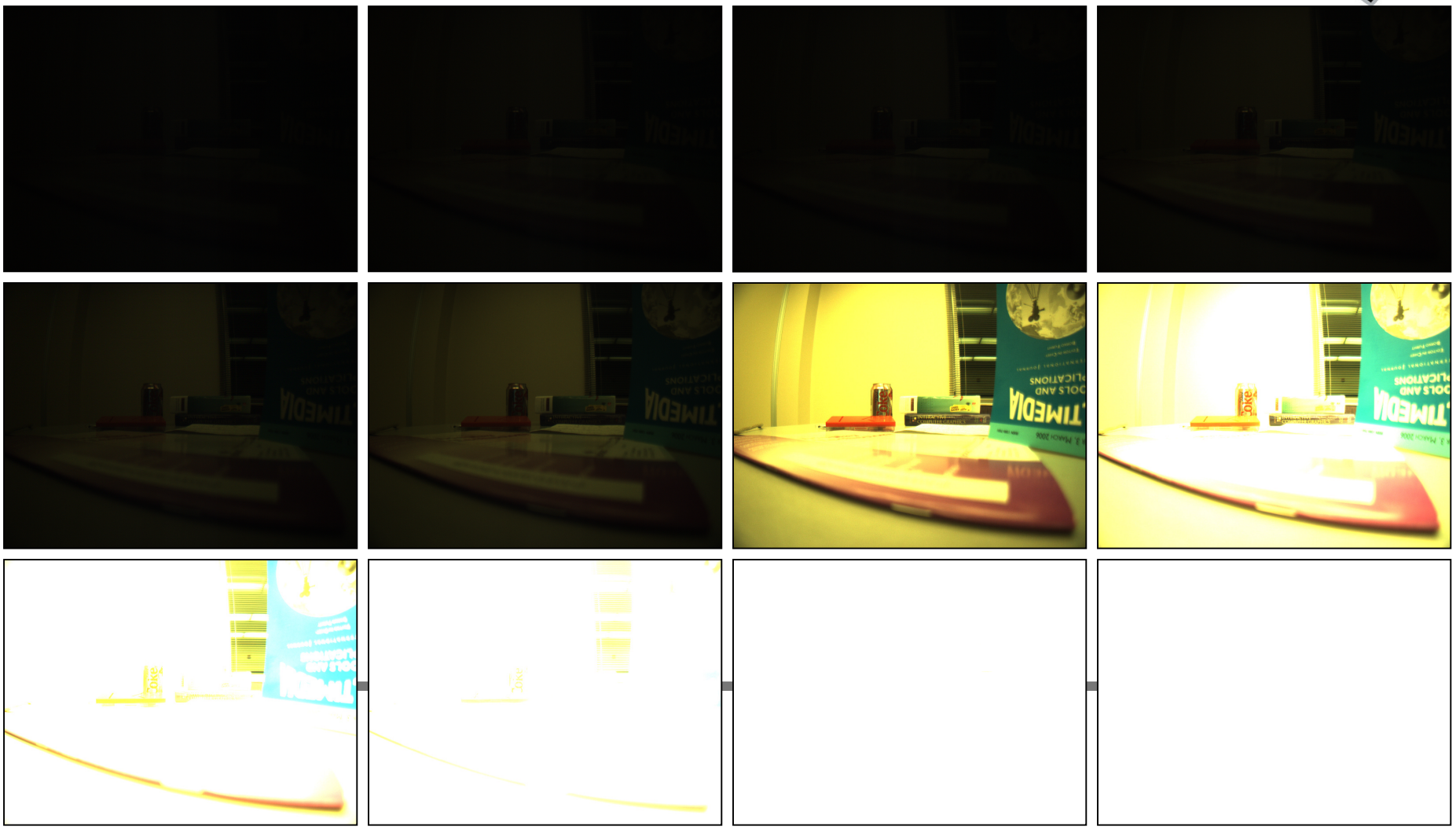

$$O = \sum_{j=1}^{P} \{ w(Z_{ijk}) [ f^{-1}(Z_{ijk}) - ( \sum H_k(\lambda) L_i(\lambda) \Delta t_j ) ] \}^2 \qquad (7)$$

- Only 256 values for the non-linear mapping (assuming 8-bits of data)
  - Calculated via minimization using Debevec96
- Weighting function is added
  - Saturated and low power pixels add noise to the system
- Scene irradiance must be calculated for every pixel
  - Requires performing many small minimizations
  - Very parallel operation (could possibly be implemented on the GPU?)

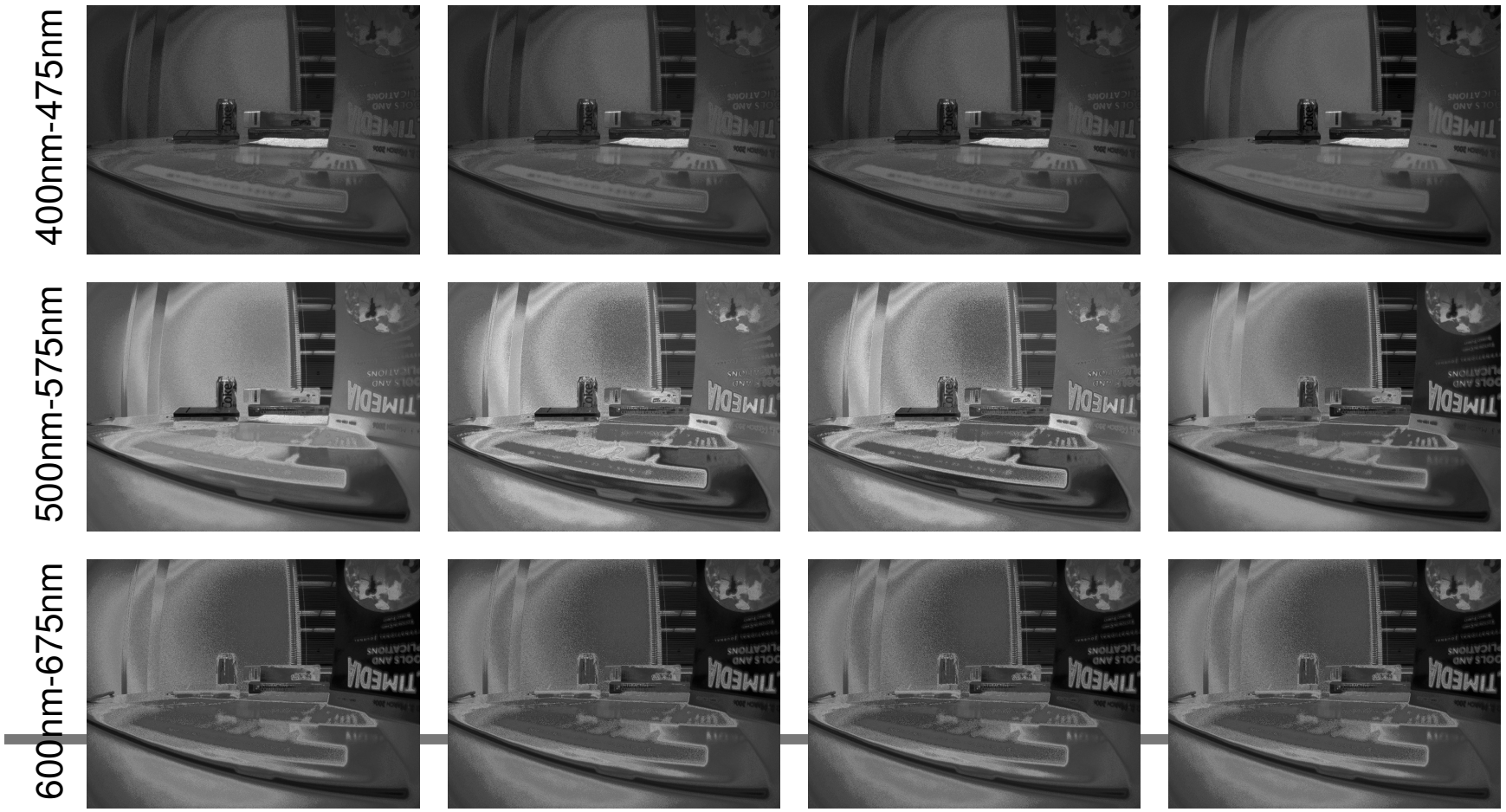
# Current Results


- 3 channel color (red, green, blue)
- 12 exposures taken (1.5ms – 5.8s)
- Calculated for wavelengths 400nm-700nm by increments of 25nm
- Computation takes ~5 minutes using a single thread on 3.2 ghz xeon

# Current Results



# Current Results



# Current Results

Blue (450nm)

Color Input

Green (550nm)

Red (650nm)

# Looking Forward


- Implementation Issues
  - Remove the dependency on Debevec's non-linear mapping function
  - Removal of noise
    - Need some coupling based upon proximity?
    - Coupling based upon color similarity?
- Apply spectral bandpass filters
- Efficient visualization of this data
- Compression of high dynamic range multispectral images