Gradient Domain
High Dynamic Range Compression
SIGGRAPH 2002

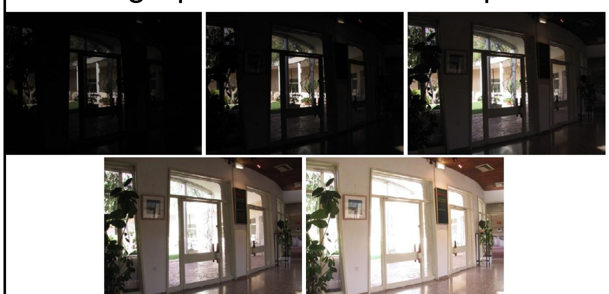
Authors
Raanan Fattal
Dani Lischinski
Michael Werman
School of Computer Science and Engineering
The Hebrew University of Jerusalem

Presenter
Matt Moses
JHU Mechanical Engineering

Problem Statement

- HDR data arises from
  - physically-based lighting simulations
  - differently-exposed photographs of a scene
  - panoramic video scan
  - direct output of digital cameras
- How do you display HDR data on LDR display?
  - HDR Data 25000:1
  - LDR Display 100:1

Photographs with different exposure



One way: Global mapping

- Spatially invariant (tone reproduction curves)
  - naive: linearly scale to some interval [0,1]
  - gamma correction
  - histogram equalization
  - Ward Larson [1997]
    - improve histogram equalization
    - add models of human response
- Good
  - simple and fast
- Bad
  - lose local contrast when intensities in ROI populate entire dynamic range

Spatially Variant Mapping

- An image is a product of reflectance and illuminance at each point
  - HDR compression related to recovering reflectances
  - real world reflectances are LDR (100:1)
- In principle:
  - separate I into R and L
  - scale down L
  - recombine

$$I(x,y) = R(x,y)\, L(x,y)$$

$$\tilde{I}(x,y) = R(x,y)\, \tilde{L}(x,y)$$

Spatially Variant Mapping

- In general, finding R and L is ill-posed
- Need to make simplifying assumptions
- Stockman [1972]
  - L varies slowly, R abruptly, extract R by highpass
- Horn [1974]
  - L is smooth, R is piecewise constant, threshold Laplacian of I to recover L
- Problems with shadows and halos

more Spatially Variant Mapping

- Jobson [1997]
  - Land's Retinex: R(x,y) is ratio of I(x,y) to lowpass of I(x,y)
  - compute log(retinex) for several lowpass filters, take linear combination
- Similar work
  - Chiu [1993]
  - Schlick [1994],
  - Tanaka and Ohnishi [1997]
- Problem: Halos!

Halos

- Halos are a problem with any multi-resolution operator that compresses each resolution band differently
  - DiCarlo and Wandell [2001]
  - Tumblin and Turk [1999]
- Low curvature image simplifier (LCIS)
  - fixes halos
  - complicated (8 parameters, comp. intensive)
  - still produces artifacts

Gradient Domain HDR Compression

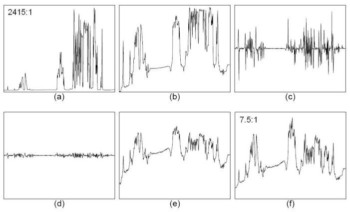
- Assumption
  - human vision sensitive to changes in local intensity
  - not absolute luminances
- Basic idea
  - identify large gradients at various scales
  - attenuate gradients while maintaining direction
  - reconstruct image from attenuated gradient field
- Benefits
  - straightforward, fast(ish), less artifacts

Sample: Larson

Sample: Gradient Domain

Sample: LCIS

Basic Idea: 1D case



(a) High-Dynamic-Range scanline

(b) H(x) = log(scanline)

(c) derivative H'(x)

(d) G(x) = attenuated H'(x) $G(x) = H'(x)\,\Phi(x)$

(e) $I(x) = C + \int_0^x G(t)\,dt.$

(f) exp(I(x))

2D Case

- We can't just integrate G(x,y)

$$G(x,y) = \nabla H(x,y)\;\Phi(x,y)$$

Unlike the 1D case we cannot simply obtain a compressed dynamic range image by integrating G, since it is not necessarily integrable. In other words, there might not exist an image I such that $G = \nabla I$! In fact, the gradient of a potential function (such as a 2D image) must be a *conservative field* [Harris and Stocker 1998]. In other words, the gradient $\nabla I = (\partial I/\partial x, \partial I/\partial y)$ must satisfy

$$\frac{\partial^2 I}{\partial x \partial y} = \frac{\partial^2 I}{\partial y \partial x},$$

which is rarely the case for our G.

Solve for I in a least-squared-error way

I should minimize the integral

$$\iint F(\nabla I, G)\,dx\,dy, \qquad (2)$$

where $F(\nabla I, G) = \|\nabla I - G\|^2 = \left(\frac{\partial I}{\partial x} - G_x\right)^2 + \left(\frac{\partial I}{\partial y} - G_y\right)^2.$

According to the Variational Principle, a function I that minimizes the integral in (2) must satisfy the Euler-Lagrange equation

$$\frac{\partial F}{\partial I} - \frac{d}{dx}\frac{\partial F}{\partial I_x} - \frac{d}{dy}\frac{\partial F}{\partial I_y} = 0,$$

which is a partial differential equation in I. Substituting F we obtain the following equation:

$$2\left(\frac{\partial^2 I}{\partial x^2} - \frac{\partial G_x}{\partial x}\right) + 2\left(\frac{\partial^2 I}{\partial y^2} - \frac{\partial G_y}{\partial y}\right) = 0.$$

Dividing by 2 and rearranging terms, we obtain the well-known Poisson equation

$$\nabla^2 I = \operatorname{div} G \qquad (3)$$

What about the attenuation function?

- Attenuate gradient magnitude at each x,y
- Real world images have edges at many scales
- Need multi-resolution edge detection
- Can't just attenuate each edge at the resolution it is detected at
  - Otherwise: halos

Attenuation Function

- Construct Gaussian pyramid
- compute gradients at each level
- at each level, for each x,y compute attenuation
- propagate attenuations from coarse to fine, using linear interpolation and pointwise multiplication

How much to attenuate?

- gradients with magnitude larger than α are attenuated, those smaller are slightly magnified

$$\varphi_k(x,y) = \frac{\alpha}{\|\nabla H_k(x,y)\|}\left(\frac{\|\nabla H_k(x,y)\|}{\alpha}\right)^{\beta}$$

Sample: Attenuation

Implementation

Since both the Laplacian ∇^2 and div are linear operators, approximating them using standard finite differences yields a linear system of equations. More specifically, we approximate:

$$\nabla^2 I(x,y) \approx I(x+1,y) + I(x-1,y) + I(x,y+1) + I(x,y-1) - 4I(x,y)$$

taking the pixel grid spacing to be 1 at the full resolution of the image. The gradient ∇H is approximated using the forward difference

$$\nabla H(x,y) \approx (H(x+1,y) - H(x,y), H(x,y+1) - H(x,y)),$$

while for $\operatorname{div} G$ we use backward difference approximations

$$\operatorname{div} G \approx G_x(x,y) - G_x(x-1,y) + G_y(x,y) - G_y(x,y-1).$$

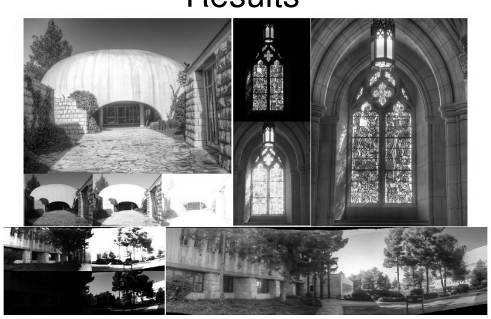
This combination of forward and backward differences ensures that the approximation of $\operatorname{div} G$ is consistent with the central difference scheme used for the Laplacian.

At the boundaries we use the same definitions, but assume that the derivatives around the original image grid are 0. For example, for each pixel on the left image boundary we have the equation $I(-1,y) - I(0,y) = 0$.

Results

Results

Results



Results

Results