Computational Photography (I)

Digital Visual Effects, Spring 2008

Yung-Yu Chuang

2008/5/20

with slides by Fredo Durand, Ramesh Raskar, Sylvain Paris, Soonmin Bae, Amit Agrawal, Ramesh Raskar

Computational photography

wikipedia:

Computational photography refers broadly to computational imaging techniques that enhance or extend the capabilities of digital photography. The output of these techniques is an ordinary photograph, but one that could not have been taken by a traditional camera.

What is computational photography

- Convergence of image processing, computer vision, computer graphics and photography
- Digital photography:
  - Simply mimics traditional sensors and recording by digital technology
  - Involves only simple image processing
- Computational photography
  - More elaborate image manipulation, more computation
  - New types of media (panorama, 3D, etc.)
  - Camera design that take computation into account

Computational photography

- One of the most exciting fields.
- Symposium on Computational Photography and Video, 2005
- Full-semester courses in MIT, CMU, Stanford, GaTech, University of Delaware
- A new book by Raskar and Tumblin in SIGGRAPH 2007.

Siggraph 2006 Papers (16/86=18.6%)

Hybrid Images
Drag-and-Drop Pasting
Two-scale Tone Management for Photographic Look
Interactive Local Adjustment of Tonal Values
Image-Based Material Editing
Flash Matting
Natural Video Matting using Camera Arrays
Removing Camera Shake From a Single Photograph
Coded Exposure Photography: Motion Deblurring
Photo Tourism: Exploring Photo Collections in 3D
AutoCollage
Photographing Long Scenes With Multi-Viewpoint Panoramas
Projection Defocus Analysis for Scene Capture and Image Display
Multiview Radial Catadioptric Imaging for Scene Capture
Light Field Microscopy
Fast Separation of Direct and Global Components of a Scene Using High Frequency Illumination

Siggraph 2007 Papers (23/108=21.3%)

Image Deblurring with Blurred/Noisy Image Pairs
Photo Clip Art
Scene Completion Using Millions of Photographs
Soft Scissors: An Interactive Tool for Realtime High Quality Matting
Seam Carving for Content-Aware Image Resizing
Detail-Preserving Shape Deformation in Image Editing
Veiling Glare in High Dynamic Range Imaging
Do HDR Displays Support LDR content? A Psychophysical Evaluation
Ldr2hdr: On-the-fly Reverse Tone Mapping of Legacy Video and Photographs
Rendering for an Interactive 360-Degree Light Field Display
Multiscale Shape and Detail Enhancement from Multi-light Image Collections
Post-Production Facial Performance Relighting Using Reflectance Transfer
Active Refocusing of Images and Videos
Multi-aperture Photography
Dappled Photography: Mask-Enhanced Cameras for Heterodyned Light Fields and Coded Aperture Refocusing
Image and Depth from a Conventional Camera with a Coded Aperture
Capturing and Viewing Gigapixel Images
Efficient Gradient-Domain Compositing Using Quadtrees
Image Upsampling via Imposed Edges Statistics
Joint Bilateral Upsampling
Factored Time-Lapse Video
Computational Time-Lapse Video
Real-Time Edge-Aware Image Processing With the Bilateral Grid

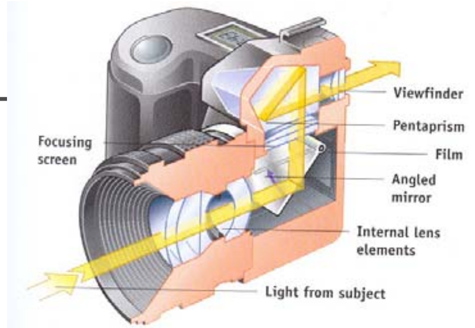
Scope

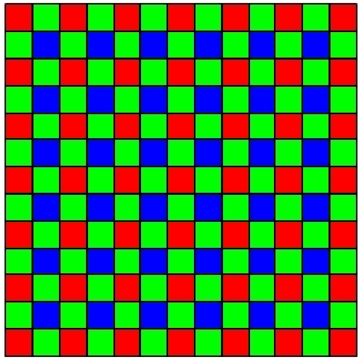
- We can't yet set its precise definition. The following are scopes of what researchers are exploring in this field.
  - Record a richer visual experience
  - Overcome long-standing limitations of conventional cameras
  - Enable new classes of visual signal
  - Enable synthesis impossible photos

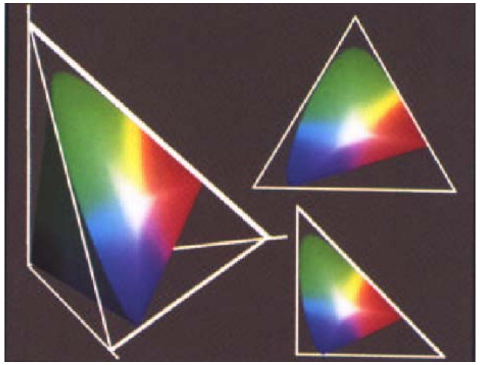
Scope

- Image formation

- Color and color perception

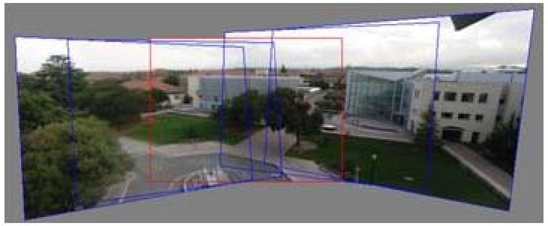




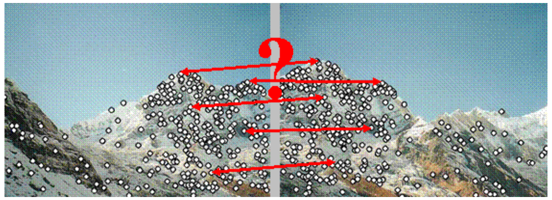


Scope

- Panoramic imaging



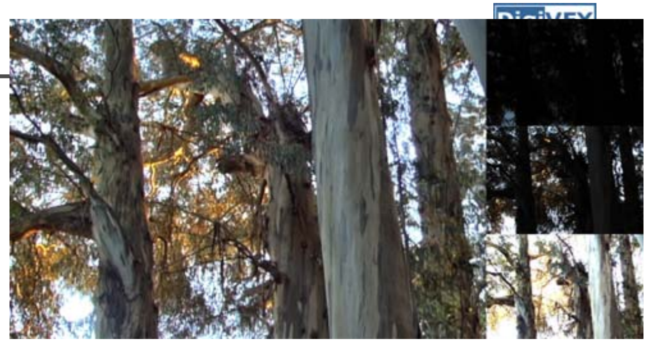
- Image and video registration



- Spatial warping operations

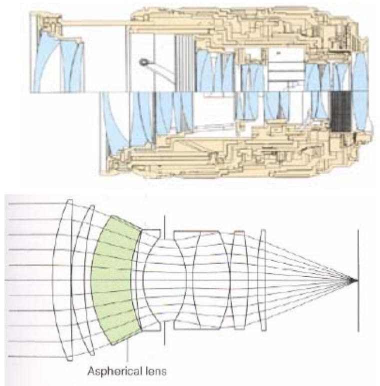
Scope

- High Dynamic Range Imaging
- Bilateral filtering and HDR display
- Matting

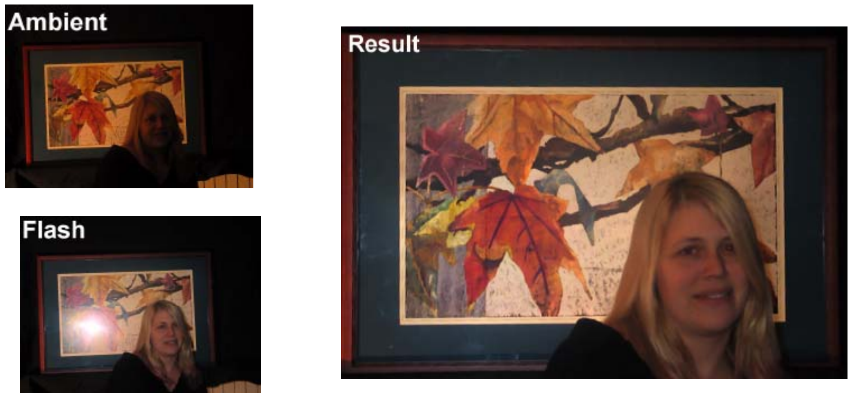


Scope

- Active flash methods
- Lens technology
- Depth and defocus



Removing Photography Artifacts using Gradient Projection and Flash-Exposure Sampling



Continuous flash

Flash = 0.0

Flash = 1.0

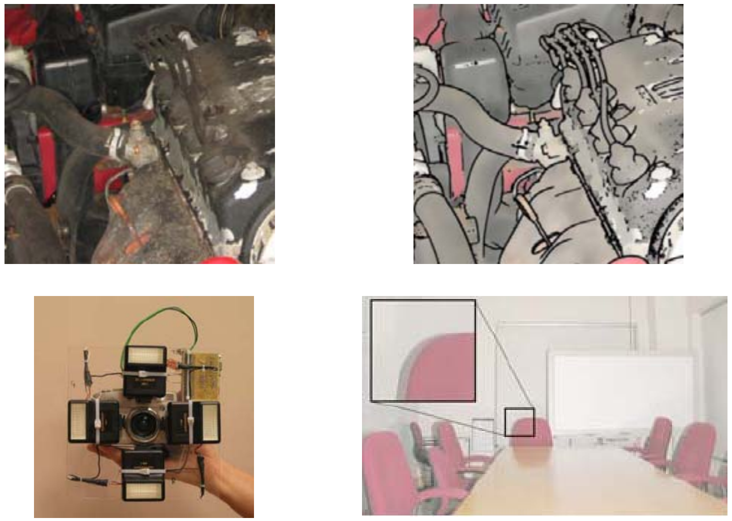
Flash = 0.3

Flash = 0.7

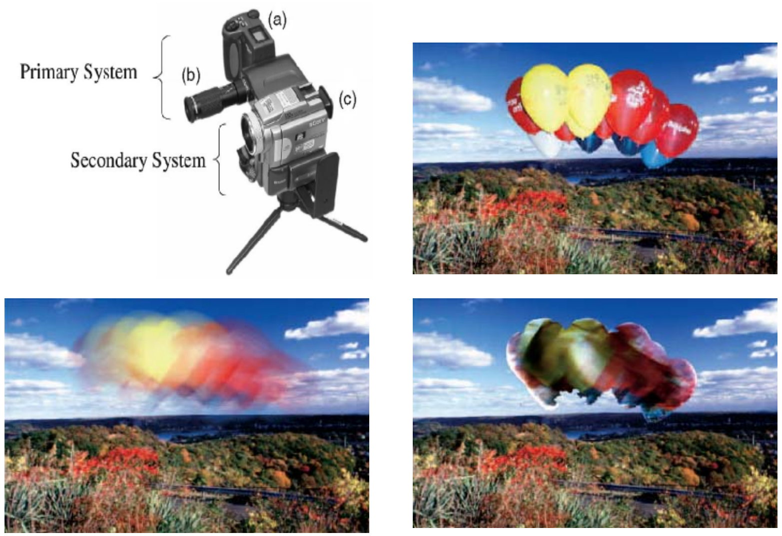
Flash = 1.4

Flash matting

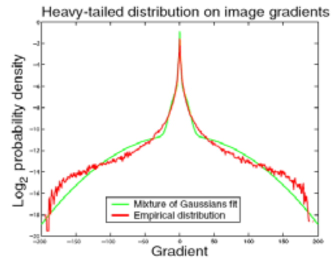
Depth Edge Detection and Stylized Rendering Using a Multi-Flash Camera



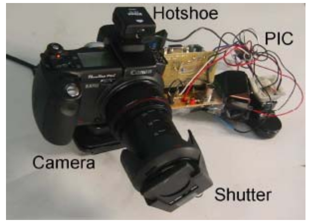
Motion-Based Motion Deblurring

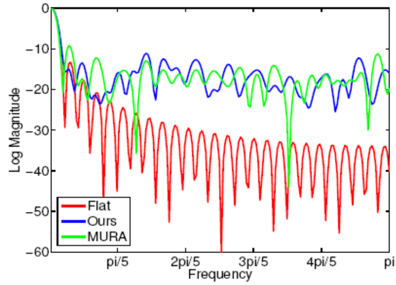


Removing Camera Shake from a Single Photograph



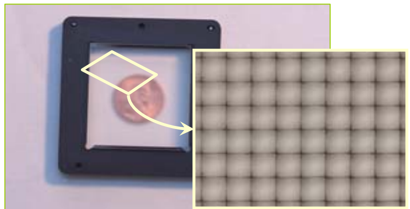
Motion Deblurring using Fluttered Shutter





Scope

- Future cameras
- Plenoptic function and light fields



Scope

- Gradient image manipulation

sources/destinations cloning seamless cloning

Scope

- Taking great pictures

Art Wolfe

Ansel Adams

Scope

- Non-parametric image synthesis, inpainting, analogies

Figure 1 An image analogy. Our problem is to compute a new "analogous" image B' that relates to B in "the same way" as A' relates to A. Here, A, A', and B are inputs to our algorithm, and B' is the output. The full-size images are shown in Figures 10 and 11.

Scope

Motion analysis

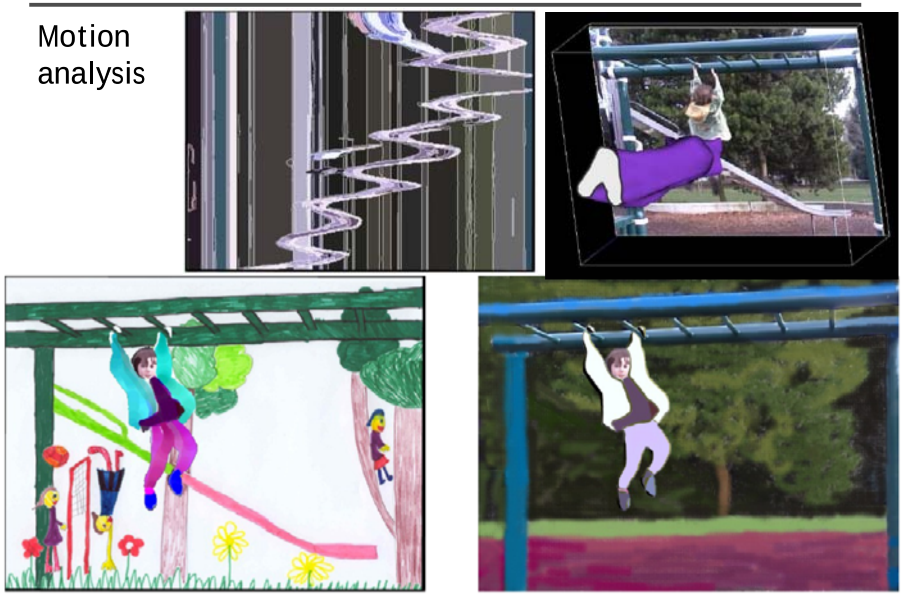
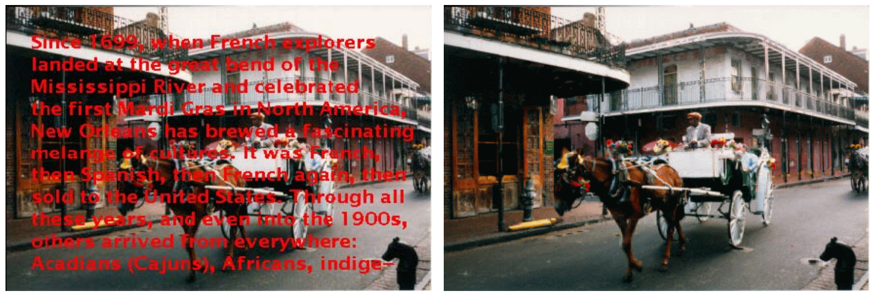


Image Inpainting



Object Removal by Exemplar-Based Inpainting

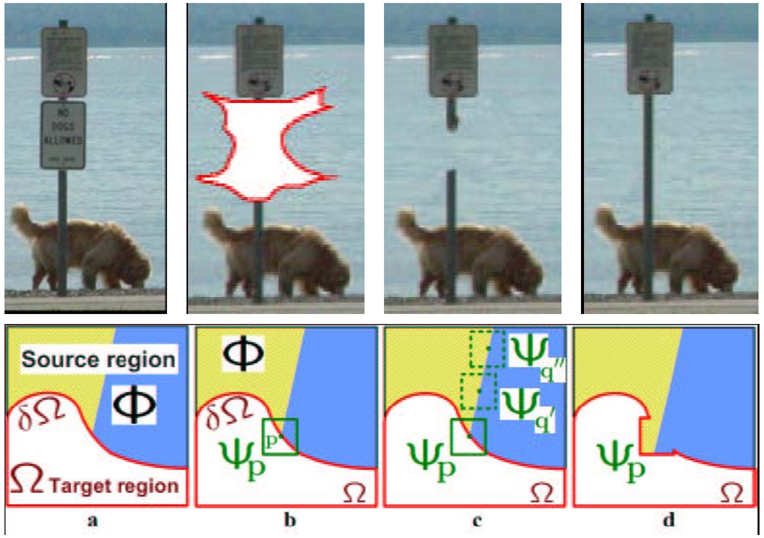


Image Completion with Structure Propagation

Lazy snapping

Grab Cut - Interactive Foreground Extraction using Iterated Graph Cuts

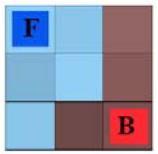
Image Tools

- Graph cuts,
  - Segmentation and mosaicing
- Gradient domain operations,
  - Tone mapping, fusion and matting
- Bilateral and Trilateral filters,
  - Denoising, image enhancement

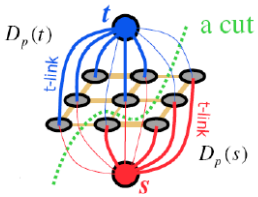
Graph cut

Graph cut

- Interactive image segmentation using graph cut
- Binary label: foreground vs. background
- User labels some pixels
  - similar to trimap, usually sparser
- Exploit
  - Statistics of known Fg & Bg
  - Smoothness of label
- Turn into discrete graph optimization
  - Graph cut (min cut / max flow)

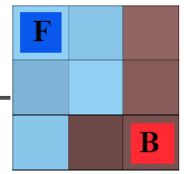


F	F	B
F	F	B
F	B	B



Energy function

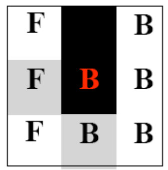
- Labeling: one value per pixel, F or B
- Energy(labeling) = data + smoothness
  - Very general situation
  - Will be minimized
- Data: for each pixel
  - Probability that this color belongs to F (resp. B)
  - Similar in spirit to Bayesian matting
- Smoothness (aka regularization): per neighboring pixel pair
  - Penalty for having different label
  - Penalty is downweighted if the two pixel colors are very different
  - Similar in spirit to bilateral filter



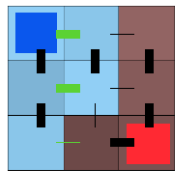
One labeling (ok, not best)

F	B	B
F	B	B
F	B	B

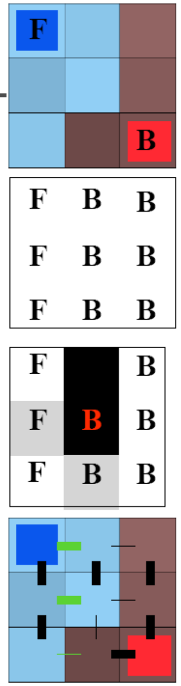
Data



Smoothness



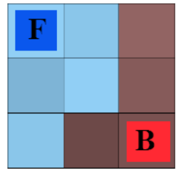
Data term



- A.k.a regional term
  (because integrated over full region)
- $D(L)=\Sigma_i -\log h[L_i](C_i)$
- Where *i* is a pixel
  L_i is the label at *i* (F or B),
  C_i is the pixel value
  $h[L_i]$ is the histogram of the observed Fg (resp Bg)
- Note the minus sign

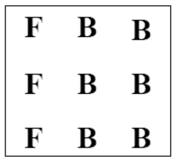
Hard constraints

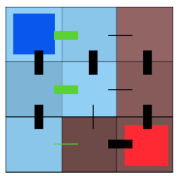
- The user has provided some labels
- The quick and dirty way to include constraints into optimization is to replace the data term by a huge penalty if not respected.
- D(L_i)=0 if respected
- D(L_i)=K if not respected
  - e.g. K=- #pixels



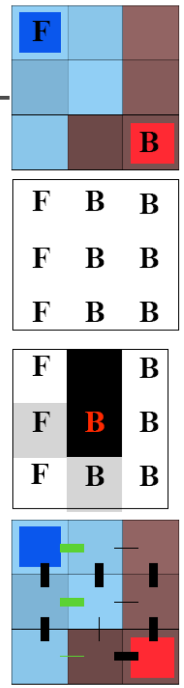
Smoothness term

- a.k.a boundary term, a.k.a. regularization
- $S(L)=\Sigma_{\{j,\ i\}\ \mathrm{in}\ N}\ B(C_i,C_j)\ \delta(L_i-L_j)$
- Where i,j are neighbors
  - e.g. 8-neighborhood
    (but I show 4 for simplicity)
- $\delta(L_i-L_j)$ is 0 if $L_i=L_j$, 1 otherwise
- $B(C_i,C_j)$ is high when C_i and C_j are similar, low if there is a discontinuity between those two pixels
  - e.g. $\exp(-||C_i-C_j||^2/2\sigma^2)$
  - where σ can be a constant or the local variance
- Note positive sign





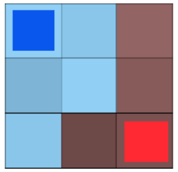
Optimization



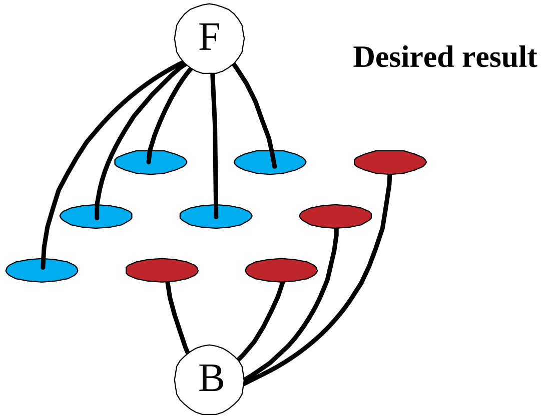
- $E(L)=D(L)+\lambda\ S(L)$
- λ is a black-magic constant
- Find the labeling that minimizes E
- In this case, how many possibilities?
  - 2^9 (512)
  - We can try them all!
  - What about megapixel images?

Labeling as a graph problem

- Each pixel = node
- Add two nodes F & B
- Labeling: link each pixel to either F or B

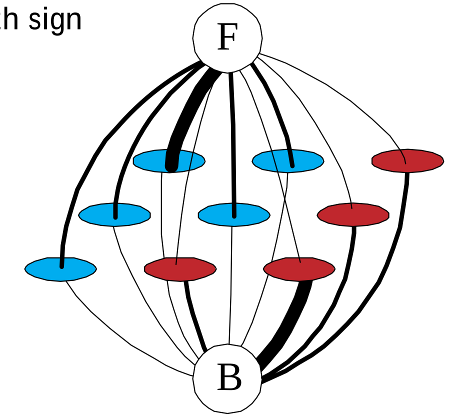


F	F	B
F	F	B
F	B	B



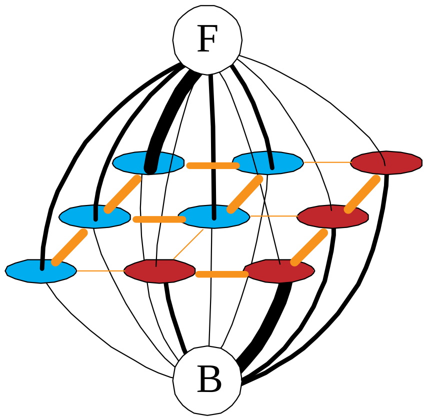
Data term

- Put one edge between each pixel and F & G
- Weight of edge = minus data term
  - Don't forget huge weight for hard constraints
  - Careful with sign



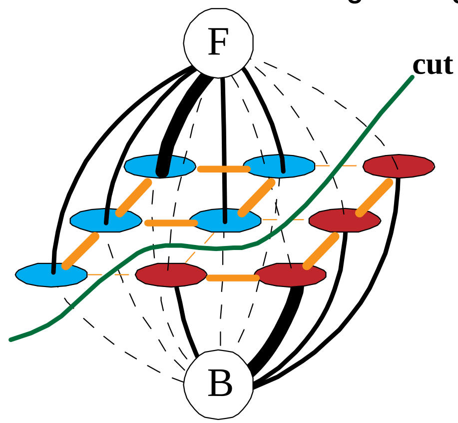
Smoothness term

- Add an edge between each neighbor pair
- Weight = smoothness term



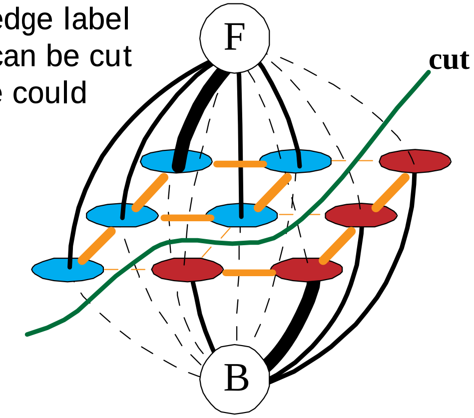
Min cut

- Energy optimization equivalent to min cut
- Cut: remove edges to disconnect F from B
- Minimum: minimize sum of cut edge weight



Min cut <=> labeling

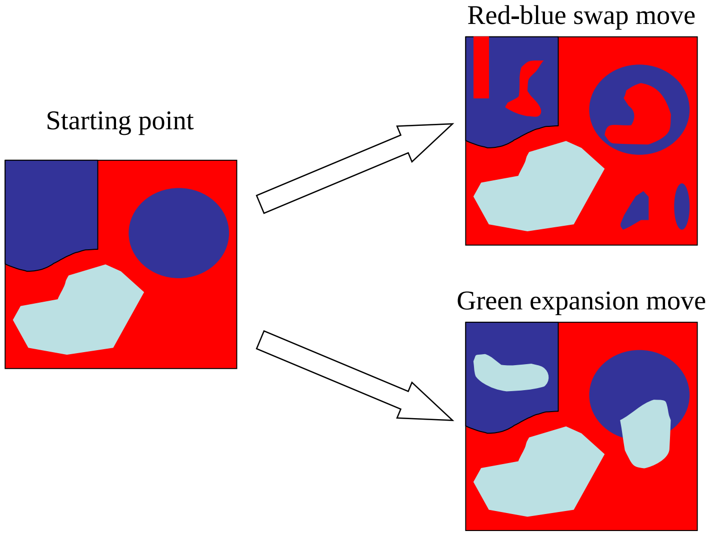
- In order to be a cut:
  - For each pixel, either the F or G edge has to be cut
- In order to be minimal
  - Only one edge label per pixel can be cut (otherwise could be added)



Computing a multiway cut

- With 2 labels: classical min-cut problem
  - Solvable by standard flow algorithms
    - polynomial time in theory, nearly linear in practice
  - More than 2 terminals: NP-hard
    [Dahlhaus *et al.*, STOC '92]
- Efficient approximation algorithms exist
  - Within a factor of 2 of optimal
  - Computes local minimum in a strong sense
    - even very large moves will not improve the energy
  - Yuri Boykov, Olga Veksler and Ramin Zabih, Fast Approximate Energy Minimization via Graph Cuts, International Conference on Computer Vision, September 1999.

Move examples



GrabCut
Interactive Foreground Extraction using Iterated Graph Cuts

Carsten Rother
Vladimir Kolmogorov
Andrew Blake

Microsoft Research Cambridge-UK

Demo

- video

Interactive Digital Photomontage

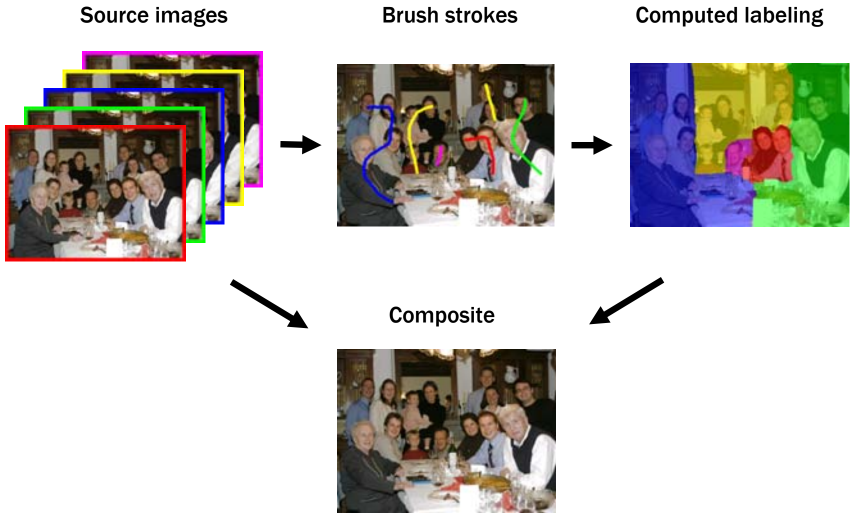
- Combining multiple photos
- Find seams using graph cuts
- Combine gradients and integrate

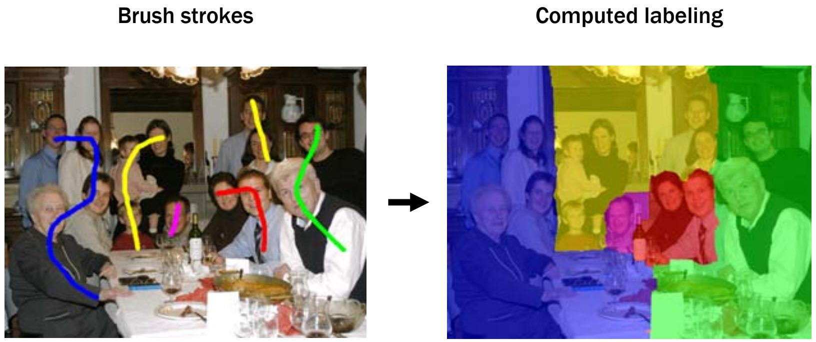
Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, Michael Cohen, "Interactive Digital Photomontage", SIGGRAPH 2004




set of originals

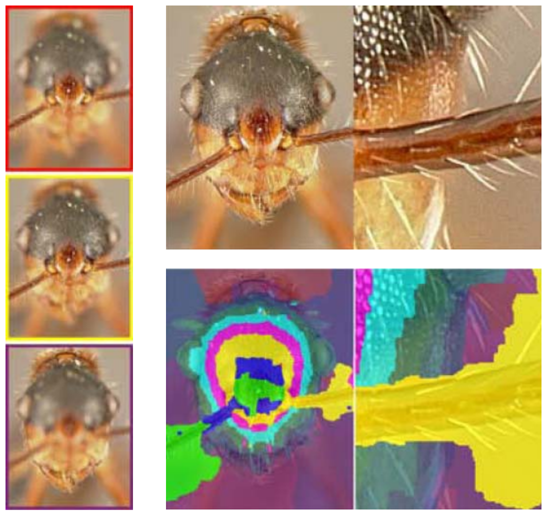
photomontage





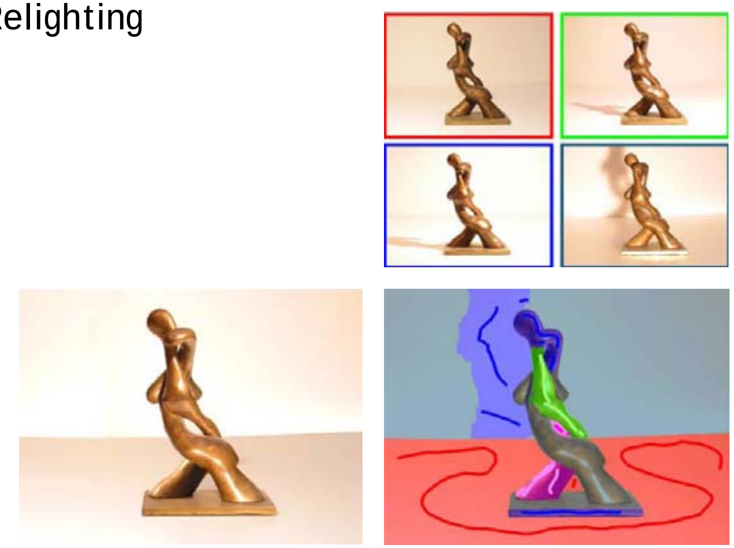
Interactive Digital Photomontage

- Extended depth of field



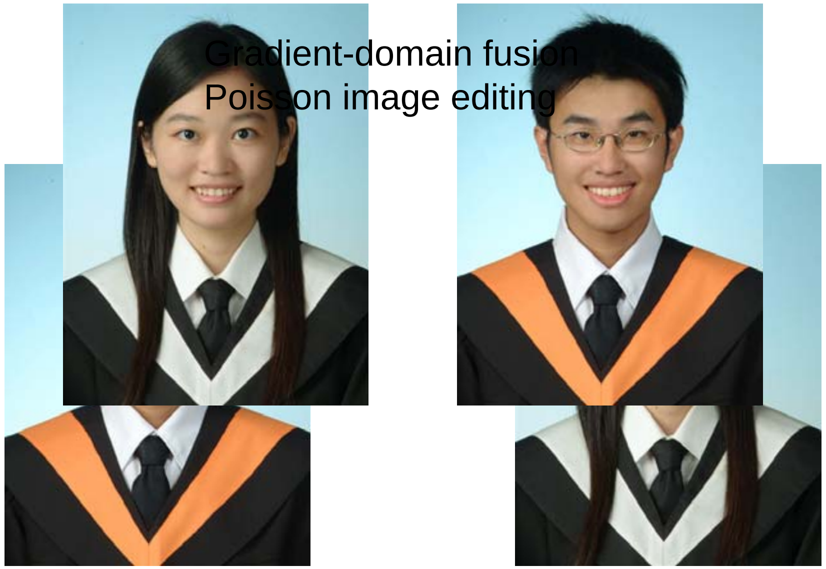
Interactive Digital Photomontage

- Relighting



Interactive Digital Photomontage

Interactive Digital Photomontage



Gradient-domain fusion
Poisson image editing

Demo

- video

Gradient domain operators

sources/destinations — cloning — seamless cloning

Gradient Domain Manipulations

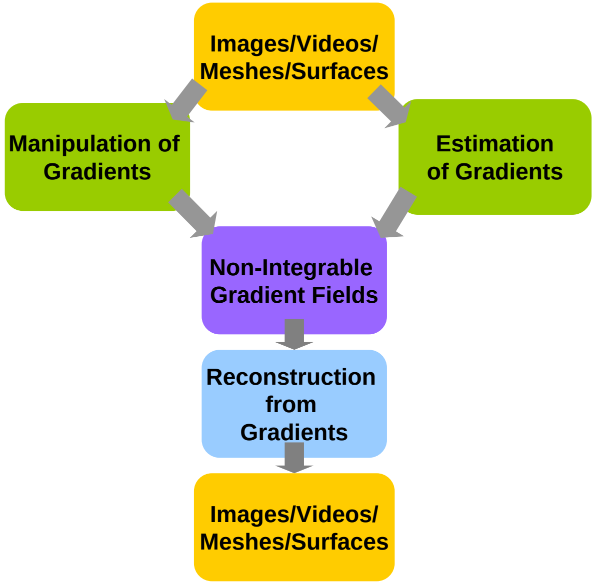
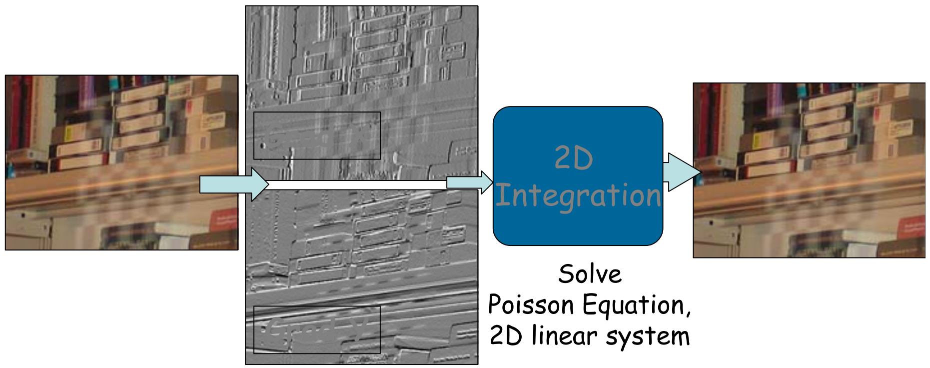
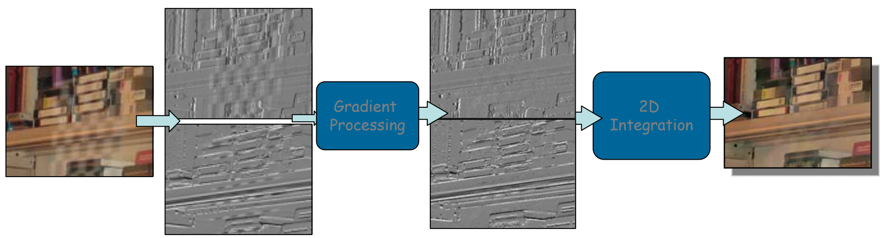


Image Intensity Gradients in 2D



Intensity Gradient Manipulation

A Common Pipeline



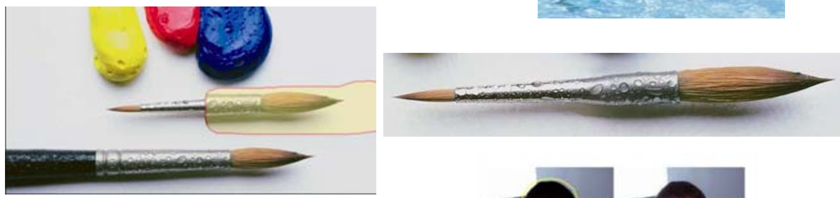
1. Gradient manipulation
2. Reconstruction from gradients

Example Applications

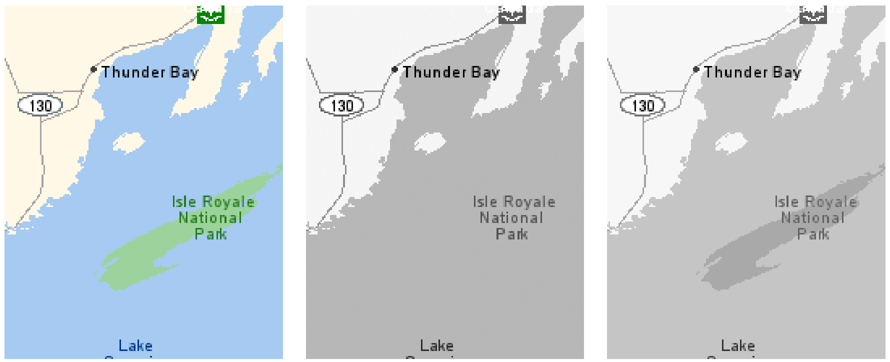
Removing Glass Reflections

Seamless Image Stitching

Image Editing



Changing Local Illumination



Original PhotoshopGrey Color2Gray

Color to Gray Conversion

High Dynamic Range Compression

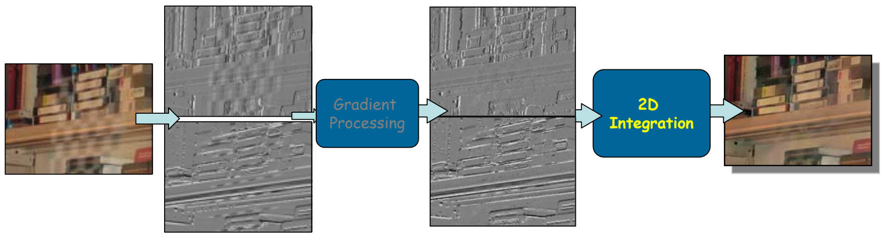
Edge Suppression under Significant Illumination Variations

Fusion of day and night images

Intensity Gradient Manipulation

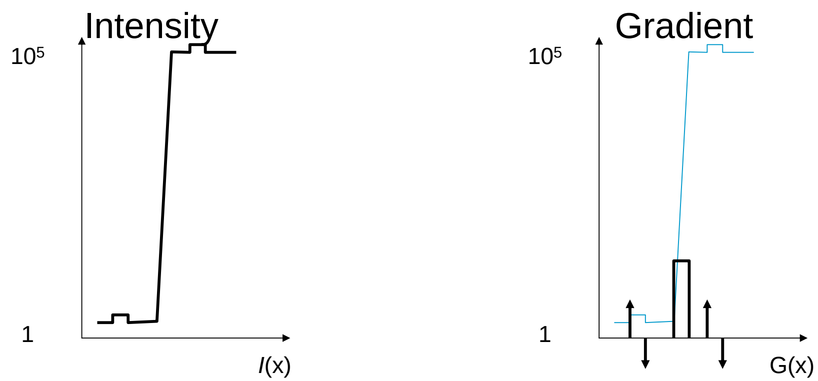
DigiVFX

A Common Pipeline



Intensity Gradient in 1D

DigiVFX



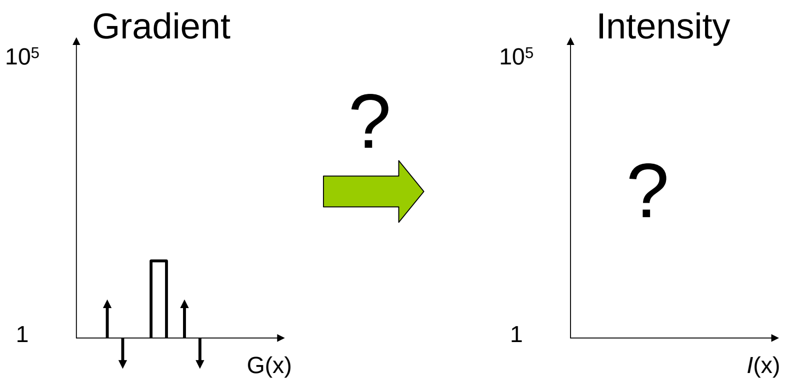
Gradient at x,

$G(x) = I(x+1) - I(x)$

Forward Difference

Reconstruction from Gradients

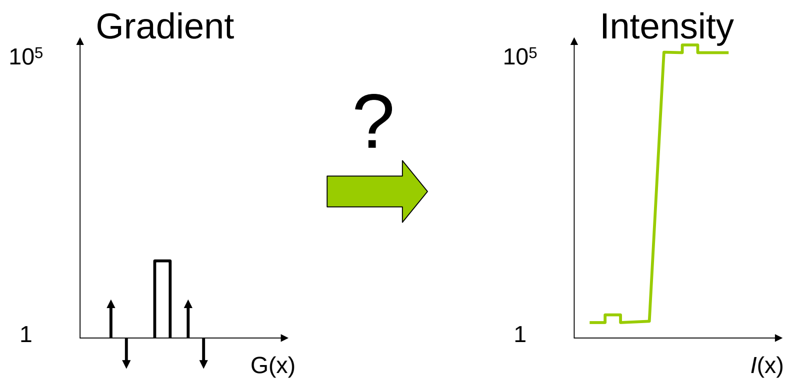
DigiVFX



For *n* intensity values, about *n* gradients

Reconstruction from Gradients

DigiVFX

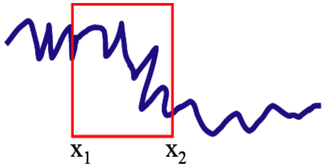


1D Integration

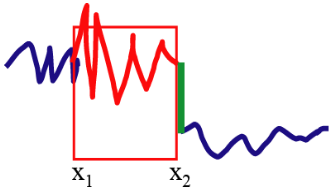
$I(x) = I(x-1) + G(x)$

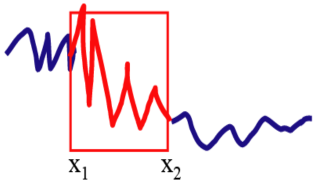
Cumulative sum

1D case with constraints

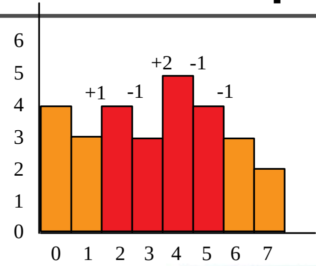
Seamlessly paste  onto 

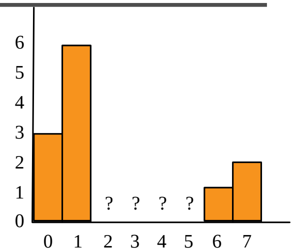
Just add a linear function so that the boundary condition is respected





Discrete 1D example: minimization

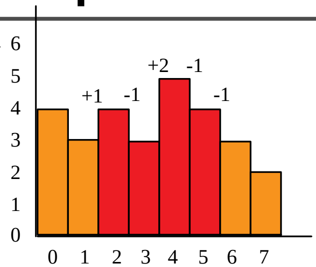
- Copy 

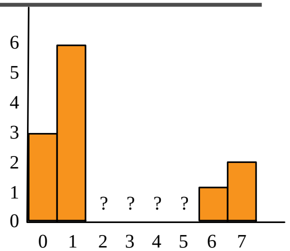
to 

- Min $((f_2-f_1)-1)^2$
- Min $((f_3-f_2)-(-1))^2$
- Min $((f_4-f_3)-2)^2$
- Min $((f_5-f_4)-(-1))^2$
- Min $((f_6-f_5)-(-1))^2$

With
$f_1=6$
$f_6=1$

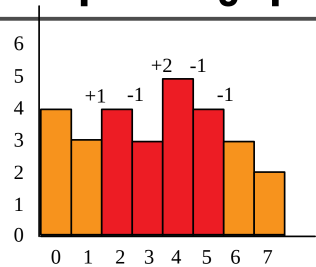
1D example: minimization

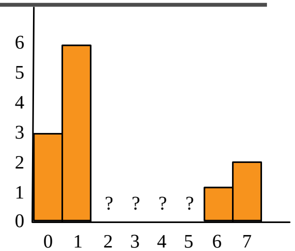
- Copy 

to 

- Min $((f_2-6)-1)^2$ ==> $f_2^2+49-14f_2$
- Min $((f_3-f_2)-(-1))^2$ ==> $f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$
- Min $((f_4-f_3)-2)^2$ ==> $f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$
- Min $((f_5-f_4)-(-1))^2$ ==> $f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$
- Min $((1-f_5)-(-1))^2$ ==> $f_5^2+4-4f_5$

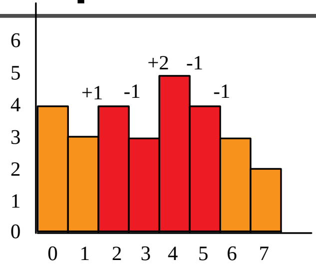
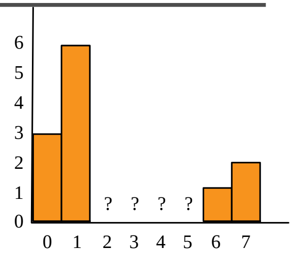
1D example: big quadratic

- Copy 

to 

- Min ($f_2^2+49-14f_2$
  $+ f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$
  $+ f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$
  $+ f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$
  $+ f_5^2+4-4f_5$)
  Denote it Q

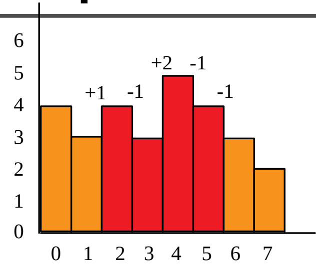
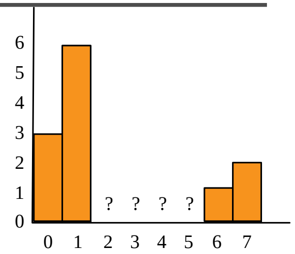
1D example: derivatives

- Copy  to 

Min (f_2^2+49-14f_2
+ f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2
+ f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3
+ f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4
+ f_5^2+4-4f_5)
Denote it Q

$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 16$$
$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2 + 2f_3 - 2f_4 + 4$$
$$\frac{dQ}{df_4} = 2f_4 - 2f_3 - 4 + 2f_4 - 2f_5 - 2$$
$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2 + 2f_5 - 4$$

1D example: set derivatives to zero

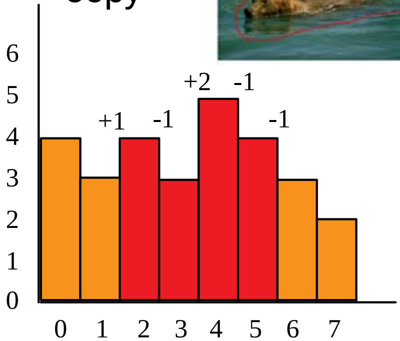
- Copy  to 

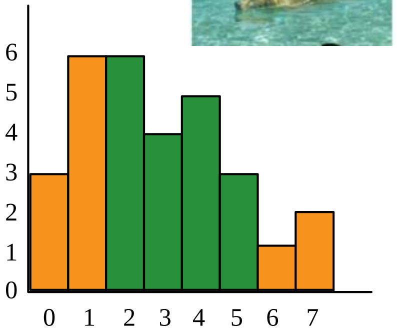
$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 16$$
$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2 + 2f_3 - 2f_4 + 4$$
$$\frac{dQ}{df_4} = 2f_4 - 2f_3 - 4 + 2f_4 - 2f_5 - 2$$
$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2 + 2f_5 - 4$$

==> $$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix}$$

1D example

- Copy  to 

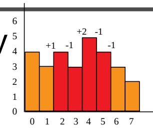
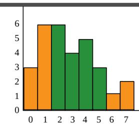




$$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix}$$

$$\boxed{\begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ 5 \\ 3 \end{pmatrix}}$$

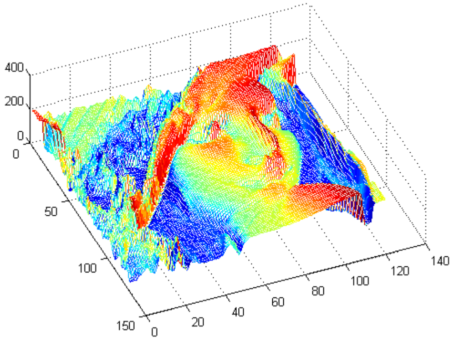
1D example: remarks

- Copy  to  $\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix}$

- Matrix is sparse
- Matrix is symmetric
- Everything is a multiple of 2
  - because square and derivative of square
- Matrix is a convolution (kernel -2 4 -2)
- Matrix is independent of gradient field. Only RHS is
- Matrix is a second derivative

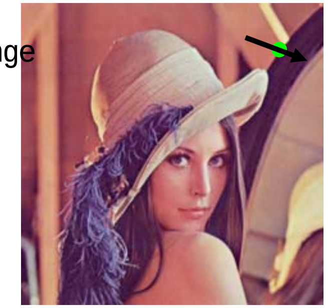
Basics

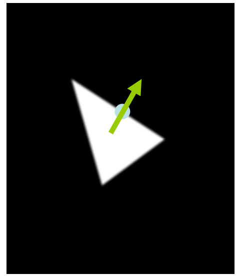
- Images as scalar fields
  - R^2 -> R



Gradients

- Vector field (gradient field)
  - Derivative of a scalar field
- Direction
  - Maximum rate of change of scalar field
- Magnitude
  - Rate of change





Gradient Field

- Components of gradient
  - Partial derivatives of scalar field

$$I(x, y) \qquad \nabla I = \{\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\}$$

$$I(x, y, t) \qquad \nabla I = \{\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t}\}$$

Example

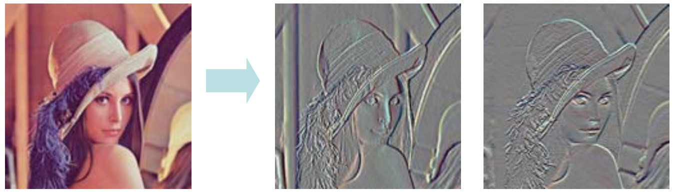


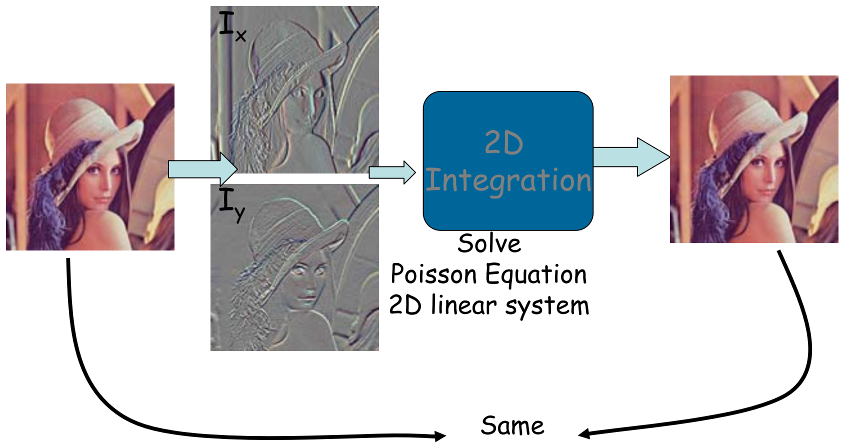
Image I(x,y) $\qquad I_x \qquad I_y$

Gradient at x,y as Forward Differences

$$G_x(x,y) = I(x+1, y) - I(x,y)$$
$$G_y(x,y) = I(x, y+1) - I(x,y)$$
$$G(x,y) = (G_x, G_y)$$

Reconstruction from Gradients

Sanity Check:
Recovering Original Image

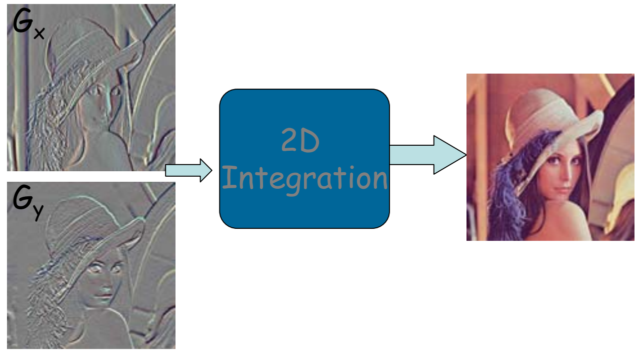


Reconstruction from Gradients

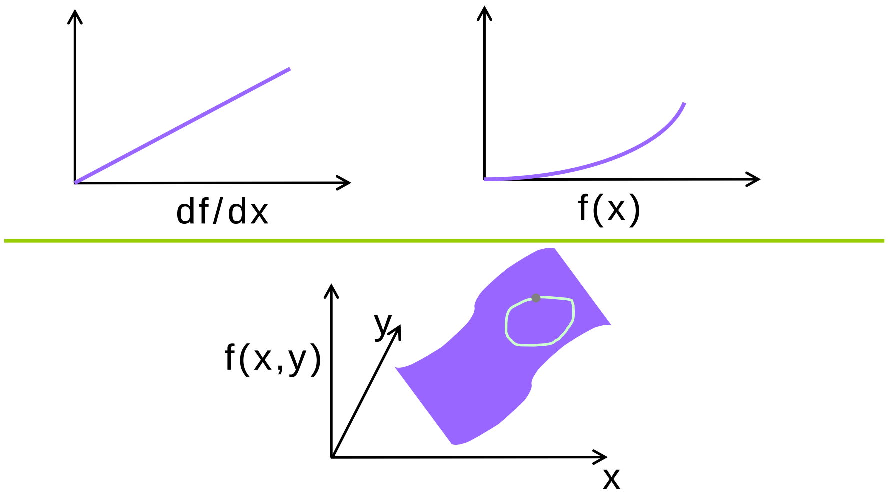
Given $G(x,y) = (G_x, G_y)$

How to compute $I(x,y)$ for the image ?

For n^2 image pixels, $2n^2$ gradients !



2D Integration is non-trivial



Reconstruction depends on chosen path

Reconstruction from Gradient Field G

- Look for image I with gradient closest to G in the least squares sense.
- I minimizes the integral: $\iint F(\nabla I, G)\,dx\,dy$

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

Poisson Equation

$$\nabla^2 I = div(G_x, G_y) = \frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial x}$$

Second order PDE

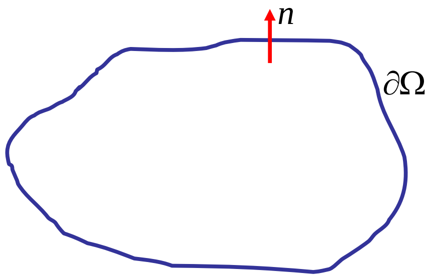
Boundary Conditions

- Dirichlet: Function values at boundary are known

$$I(x,y) = I_0(x,y) \forall (x,y) \in \partial\Omega$$

- Neumann: Derivative normal to boundary = 0

$$\nabla I(x,y) \bullet n(x,y) = 0, \forall (x,y) \in \partial\Omega$$



Numerical Solution

- Discretize Laplacian

$$\nabla^2 \longrightarrow \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

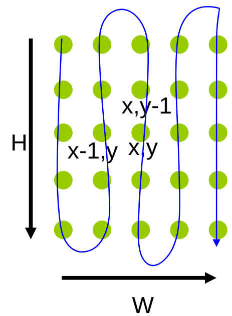
$$\nabla^2 I = div(G_x, G_y) = u(x,y)$$

$$-4I(x,y) + I(x,y+1) + I(x,y-1) + I(x+1,y) + I(x-1,y) = h^2 u(x,y)$$

h = grid size

Linear System

$$-4I(x,y) + I(x,y+1) + I(x,y-1) + I(x+1,y) + I(x-1,y) = u(x,y)$$

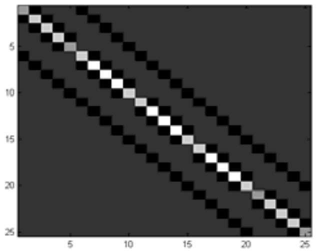


$$\underbrace{\begin{bmatrix} . & \underbrace{1 & . & . & .}_{H} & 1 & -4 & 1 & . & . & . & 1 & . \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} . \\ . \\ I(x-1,y) \\ . \\ . \\ . \\ I(x,y-1) \\ I(x,y) \\ I(x,y+1) \\ . \\ . \\ . \\ I(x+1,y) \\ . \\ . \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} . \\ . \\ . \\ . \\ . \\ . \\ . \\ u(x,y) \\ . \\ . \\ . \\ . \\ . \\ . \\ . \end{bmatrix}}_{b}$$

Sparse Linear system

$$\begin{bmatrix}
 & & 1 & -4 & 1 & & & & 1 & & & & \\
 & & & 1 & -4 & 1 & & & & 1 & & & \\
1 & & & & 1 & -4 & 1 & & & & 1 & & \\
 & 1 & & & & 1 & -4 & 1 & & & & 1 & \\
 & & 1 & & & & 1 & -4 & 1 & & & & 1 \\
 & & & 1 & & & & 1 & -4 & 1 & & & \\
 & & & & 1 & & & & 1 & -4 & 1 & &
\end{bmatrix}$$

A matrix



Solving Linear System

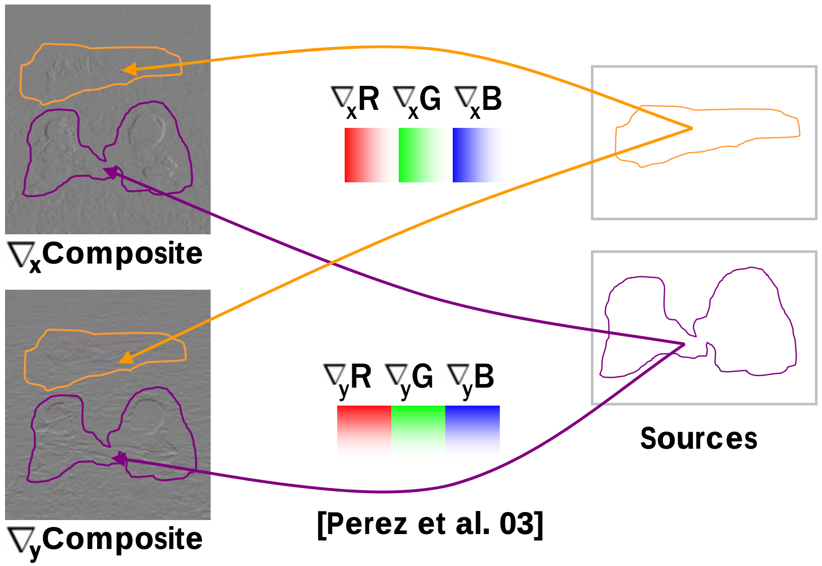
- Image size N*N
- Size of A ~ N^2 by N^2
- Impractical to form and store A

- Direct Solvers
- Basis Functions
- Multigrid
- Conjugate Gradients

Approximate Solution for Large Scale Problems

- Resolution is increasing in digital cameras
- Stitching, Alignment requires solving large linear system

Gradient-domain compositing



Gradient-domain compositing

$$I_{i,j} - I_{i+1,j} = \nabla_x \text{Composite}$$

$$I_{i,j} - I_{i,j+1} = \nabla_y \text{Composite}$$

$$\begin{bmatrix} \quad A \quad \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}$$

Scalability problem

10 X 10 MP X 50% overlap =

50 Megapixel Panorama

$$\begin{bmatrix} \quad A \quad \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}$$

Scalability problem

$$\begin{bmatrix} \quad \cancel{A} \quad \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}$$

50 million element vectors!

Approximate Solution

- Reduce size of linear system
- Handle high resolution images
- Part of Photoshop CS3

Aseem Agarwala. "Efficient gradient-domain compositing using quadtrees," ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)

The key insight

Desired solution x

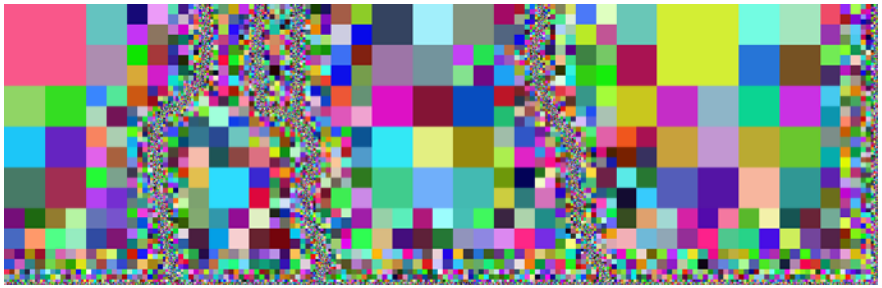
−

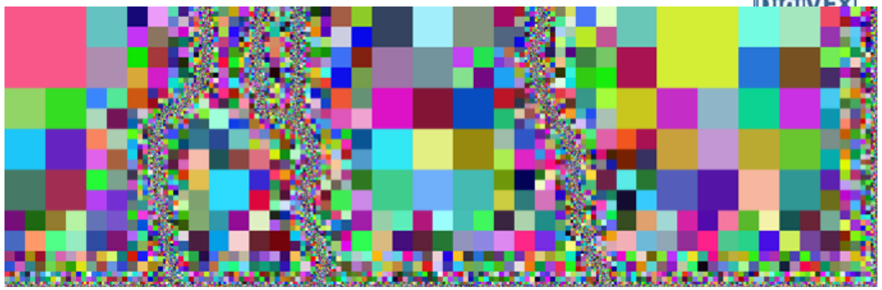
Initial Solution x_0

=

Difference x_δ

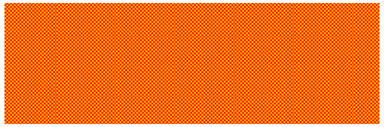
Quadtree decomposition





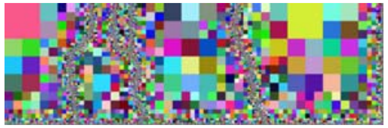
- Maximally subdivide quadtree along seams
- Variables placed at node corners
- Restricted quadtree
- Bi-linear interpolation reconstructs full solution
- Square nodes

Reduced space



x

n variables

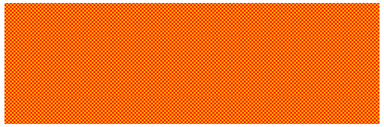


y

m variables

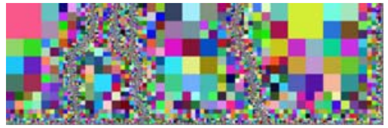
$$m << n$$

Reduced space



x

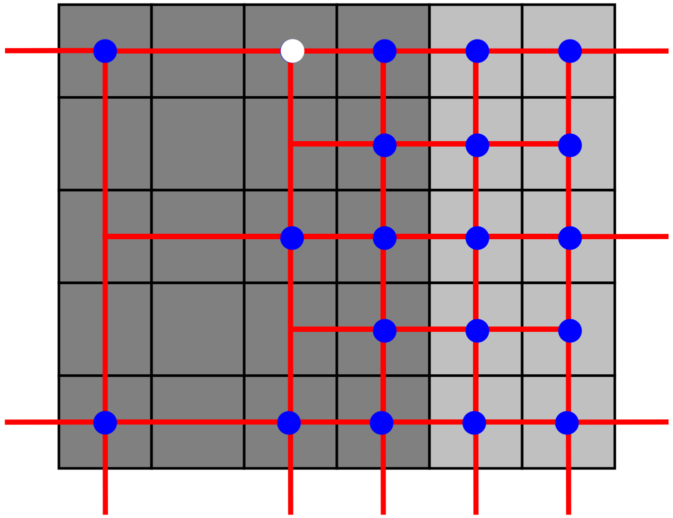
n variables



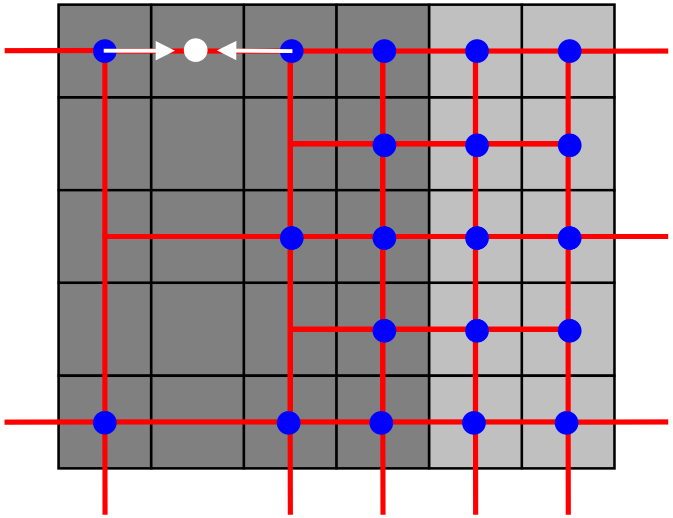
y

m variables

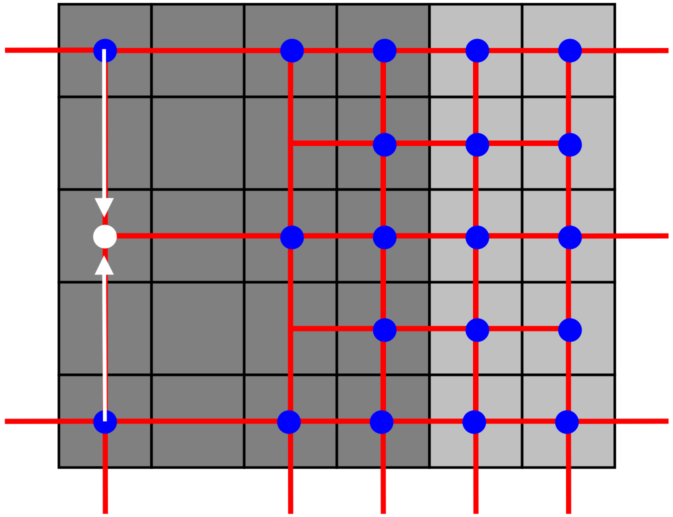
$$x = Sy$$



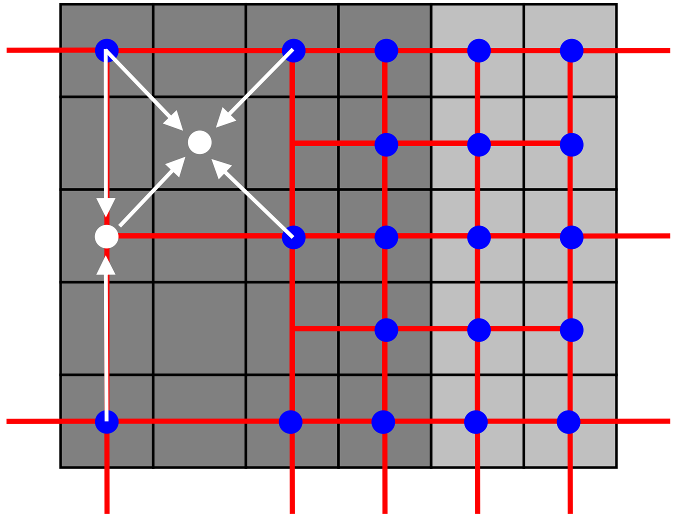
$$x = Sy$$



$$x = Sy$$

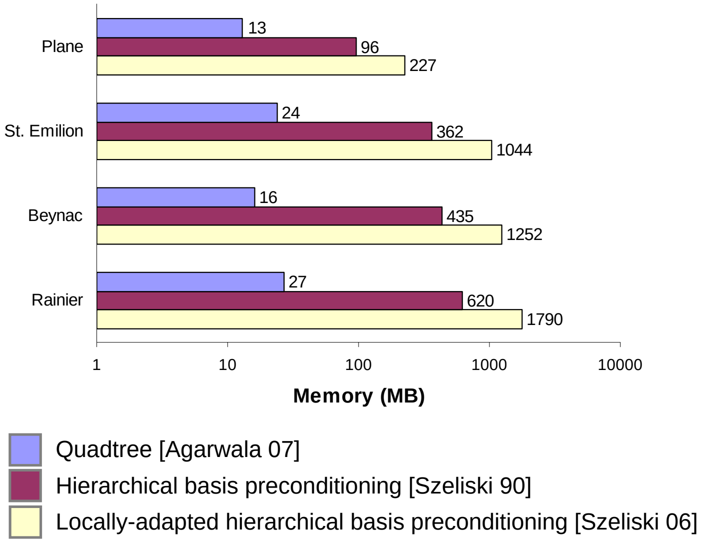


$x = Sy$



$x = Sy$

Performance

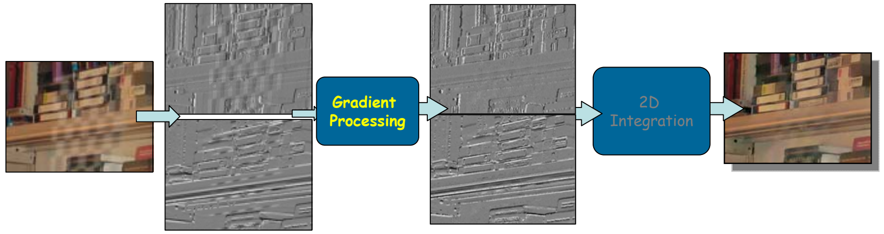


Cut-and-paste

Cut-and-paste

Intensity Gradient Manipulation

A Common Pipeline



Gradient Domain Manipulations: Overview

(A) Per pixel

(B) Corresponding gradients in two images

(C) Corresponding gradients in multiple images

(D) Combining gradients along seams

Gradient Domain Manipulations: Overview

(A) Per pixel
- Non-linear operations (HDR compression, local illumination change)
- Set to zero (shadow removal, intrinsic images, texture de-emphasis)
- Poisson Matting

(B) Corresponding gradients in two images
- Vector operations (gradient projection)
  - Combining flash/no-flash images, Reflection removal
- Projection Tensors
  - Reflection removal, Shadow removal
- Max operator
  - Day/Night fusion, Visible/IR fusion, Extending DoF
- Binary, choose from first or second, copying
  - Image editing, seamless cloning

Gradient Domain Manipulations

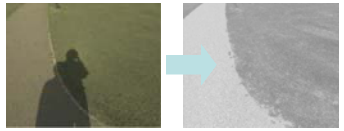
(C) Corresponding gradients in multiple images
- Median operator
  - Specularity reduction
  - Intrinsic images
- Max operation
  - Extended DOF

(D) Combining gradients along seams
- Weighted averaging
- Optimal seam using graph cut
  - Image stitching, Mosaics, Panoramas, Image fusion
  - A usual pipeline: Graph cut to find seams + gradient domain fusion

A. Per Pixel Manipulations

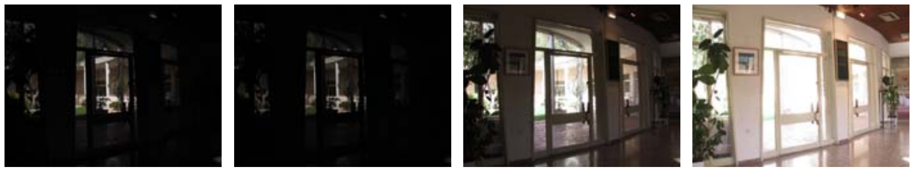
- Non-linear operations
  - HDR compression, local illumination change

- Set to zero
  - Shadow removal, intrinsic images, texture de-emphasis



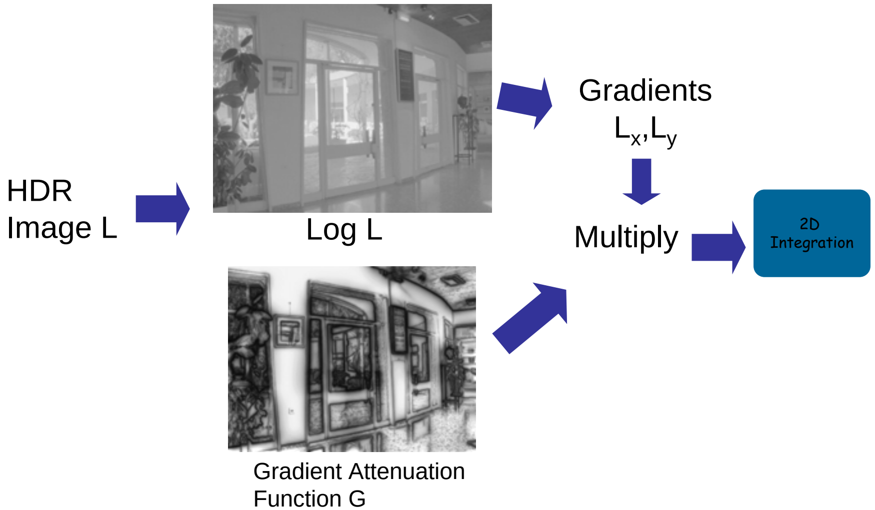
- Poisson Matting

High Dynamic Range Imaging



Images from Raanan Fattal

Gradient Domain Compression



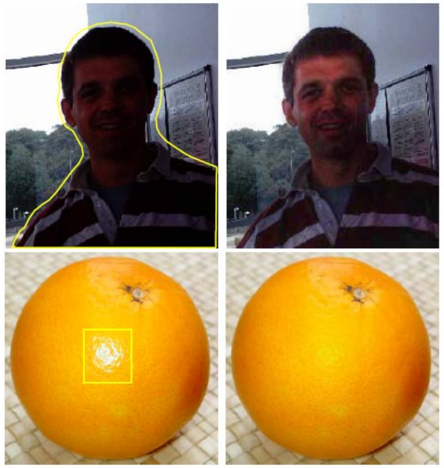
Local Illumination Change

Original Image: f

$$\mathbf{v} = \alpha^{\beta} |\nabla f^*|^{-\beta} \nabla f^*,$$

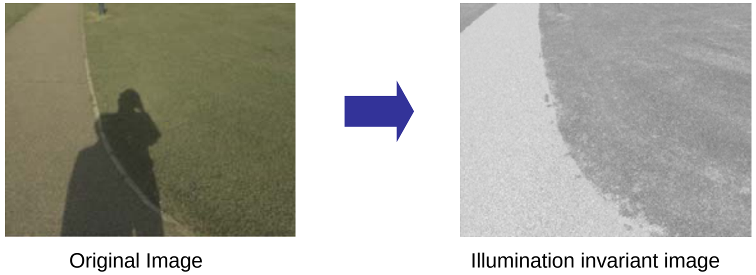
Original gradient field: ∇f^*

Modified gradient field: v



Perez et al. Poisson Image editing, SIGGRAPH 2003

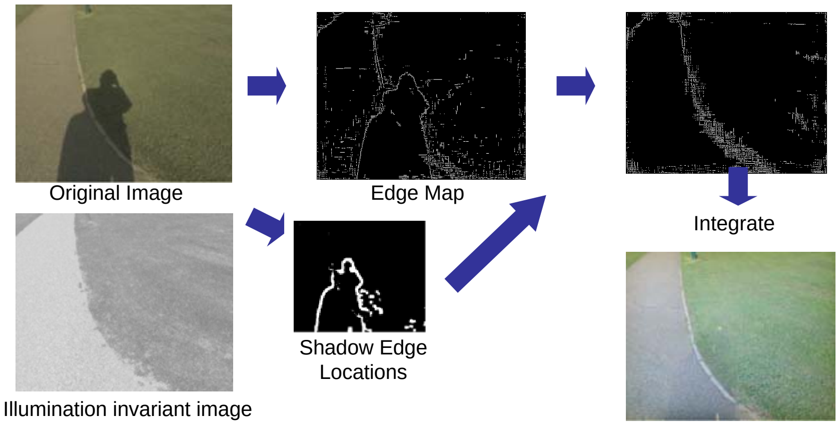
Illumination Invariant Image



- Assumptions
  - Sensor response = delta functions R, G, B in wavelength spectrum
  - Illumination restricted to Outdoor Illumination

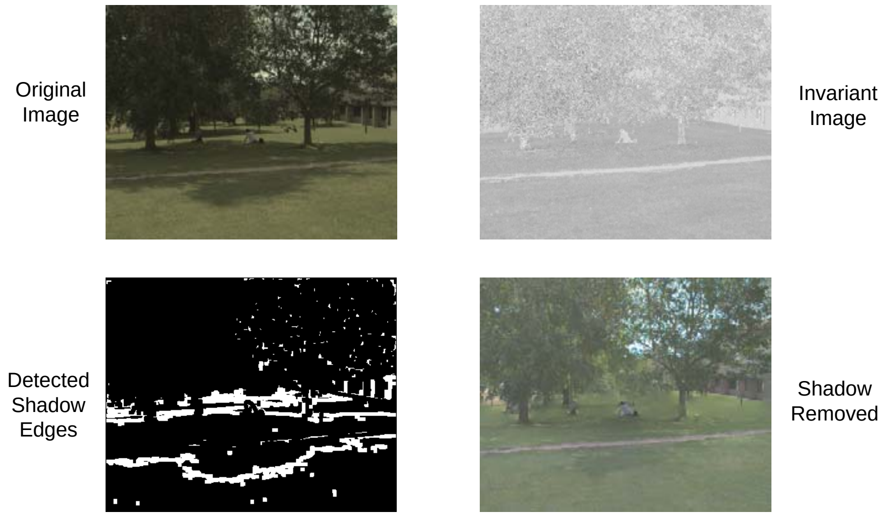
G. D. Finlayson, S.D. Hordley & M.S. Drew, Removing Shadows From Images, ECCV 2002

Shadow Removal Using Illumination Invariant Image



G. D. Finlayson, S.D. Hordley & M.S. Drew, Removing Shadows From Images, ECCV 2002

Illumination invariant image



G. D. Finlayson, S.D. Hordley & M.S. Drew, Removing Shadows From Images, ECCV 2002

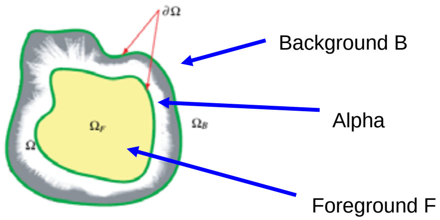
Intrinsic Image

- Photo = Illumination Image * Intrinsic Image
- Retinex [Land & McCann 1971, Horn 1974]
  - Illumination is smoothly varying
  - Reflectance, piece-wise constant, has strong edges
  - Keep strong image gradients, integrate to obtain reflectance

low-frequency attenuate more

high-frequency attenuate less

Poisson Matting



Trimap: User specified

Jian Sun, Jiaya Jia, Chi-Keung Tang, Heung-Yeung Shum, Poisson Matting, SIGGRAPH 2004

Poisson Matting

$$I = \alpha F + (1-\alpha)B$$

$$\nabla I = (F-B)\nabla\alpha + \alpha\nabla F + (1-\alpha)\nabla B$$

Approximate: Assume F and B are smooth

$$\nabla I = (F-B)\nabla\alpha$$

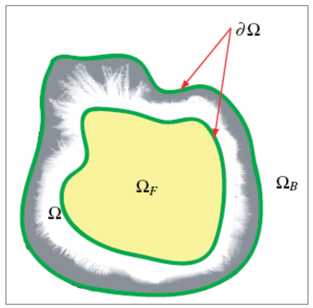
$$\nabla\alpha \approx \frac{1}{F-B}\nabla I \quad \Rightarrow \quad \Delta\alpha = div\left(\frac{\nabla I}{F-B}\right)$$

F and B in tri-map using nearest pixels

Poisson Equation

Poisson Matting

- Steps
  - Approximate F and B in trimap Ω
  - Solve for α $\quad \Delta\alpha = div\left(\frac{\nabla I}{F-B}\right)$
  - Refine F and B using α
  - Iterate



Gradient Domain Manipulations: Overview

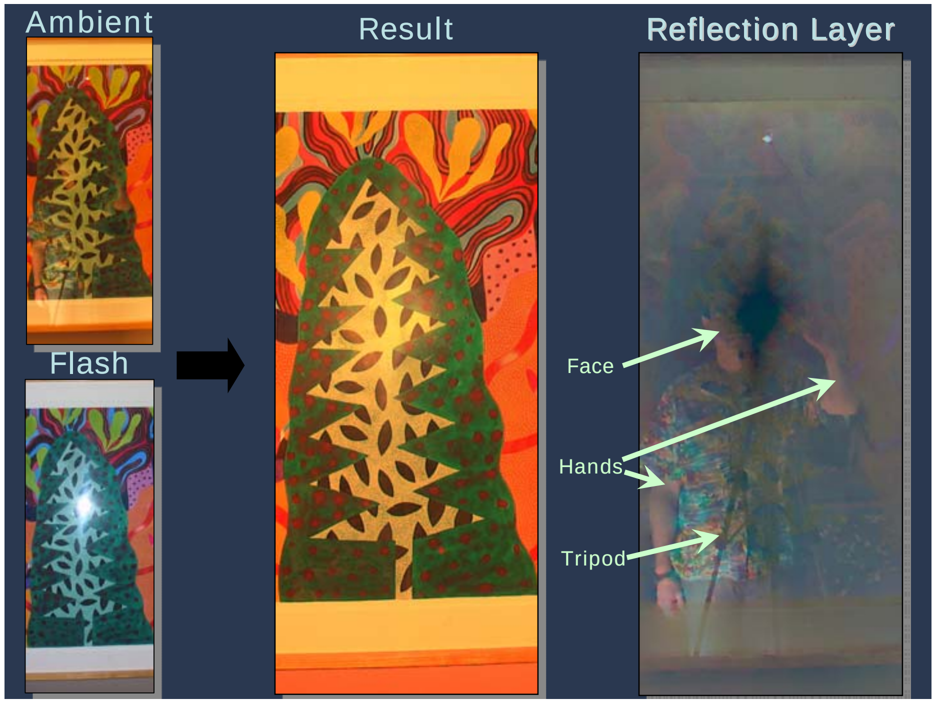
(A) Per pixel

(B) Corresponding gradients in two images

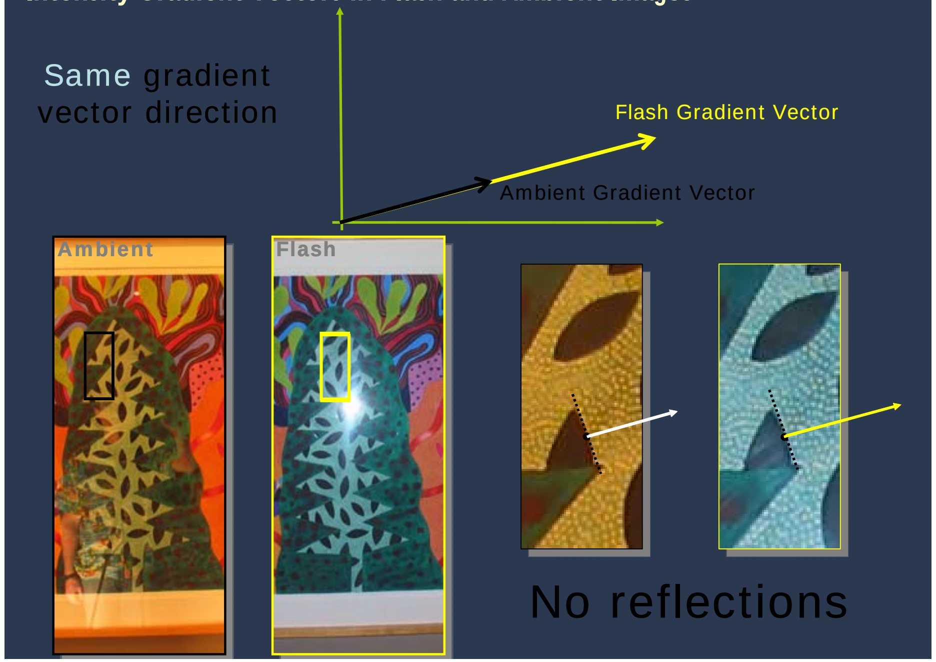
(C) Corresponding gradients in multiple images

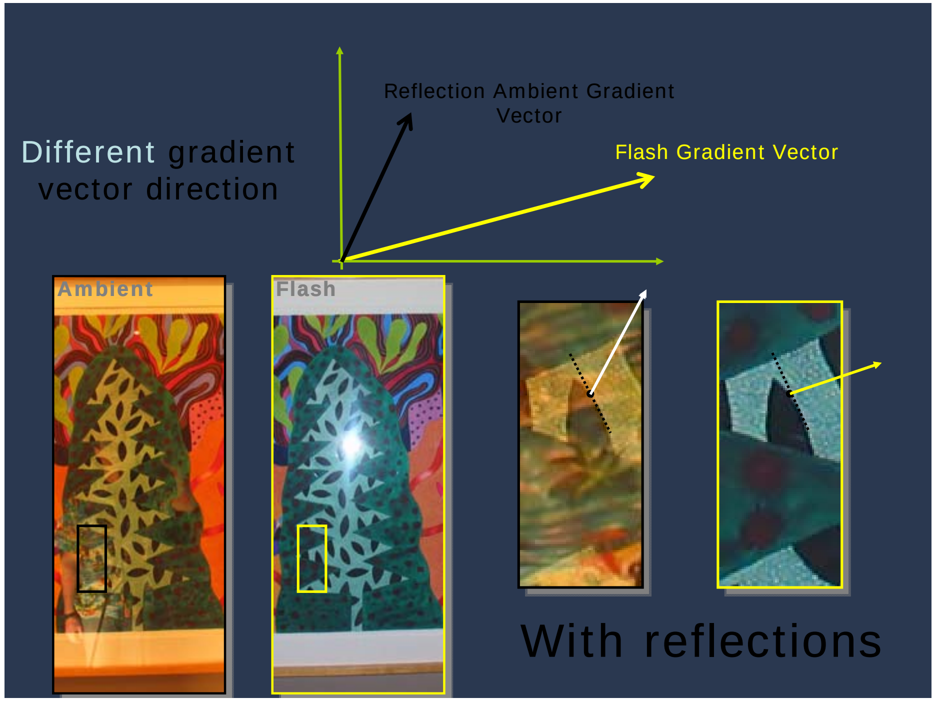
(D) Combining gradients along seams

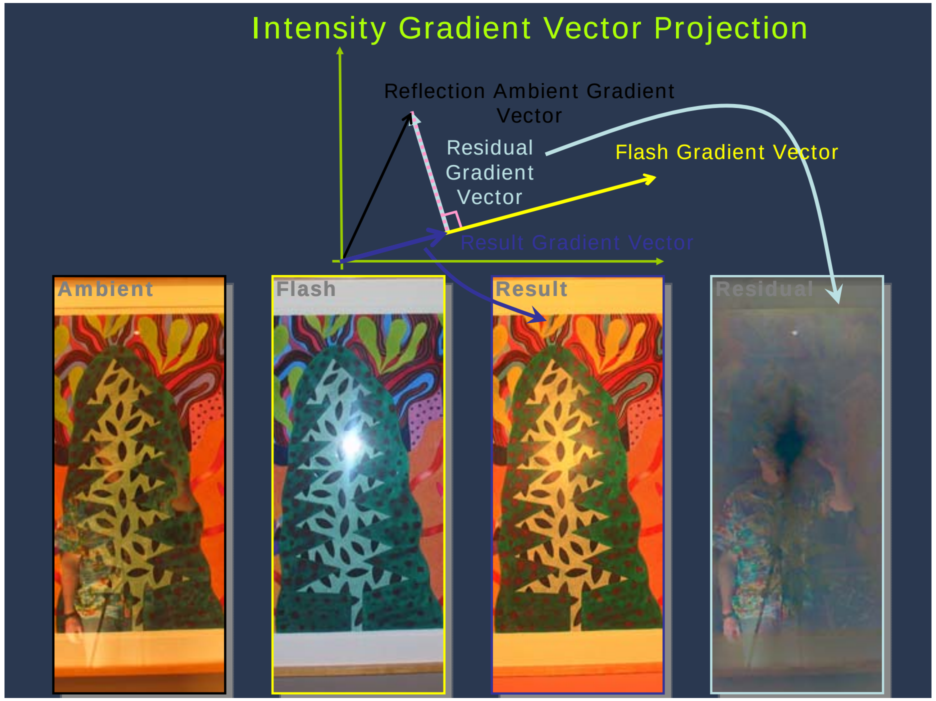
Self-Reflections and Flash Hotspot

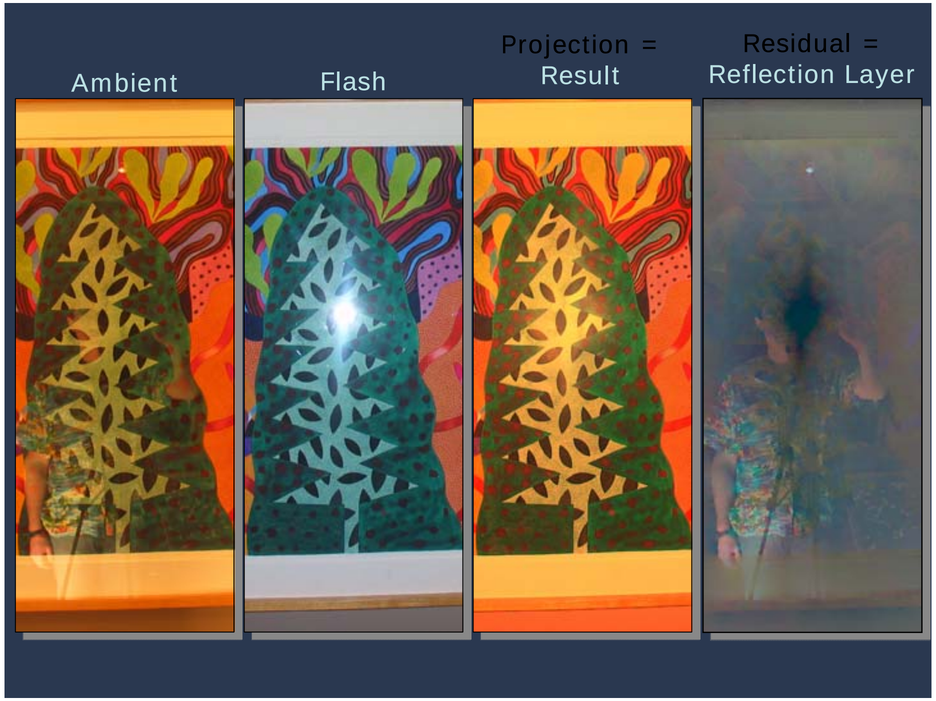


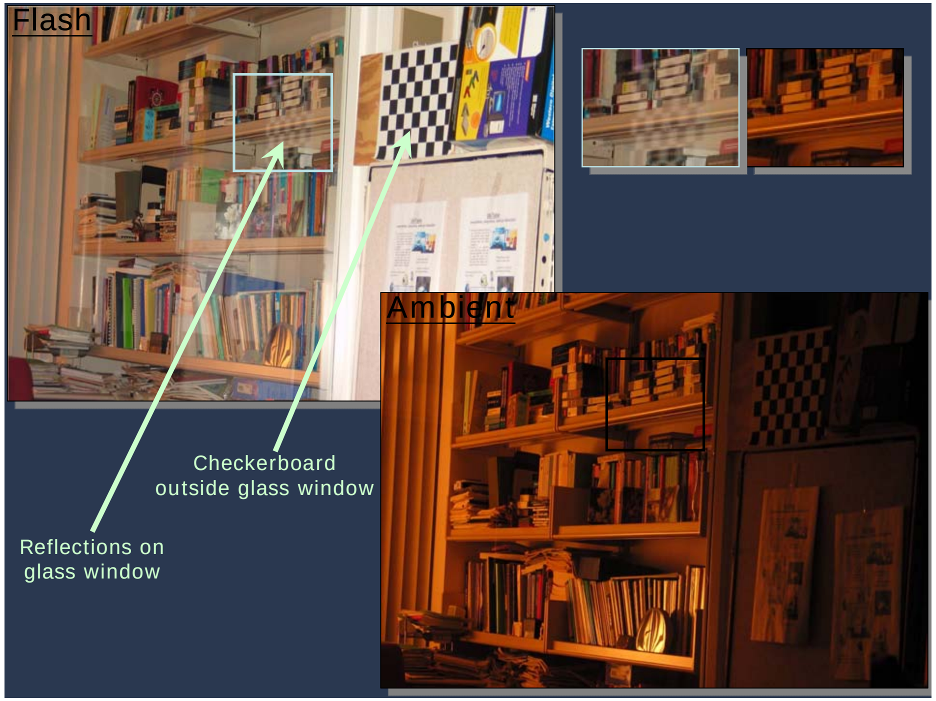
Intensity Gradient Vectors in Flash and Ambient Images

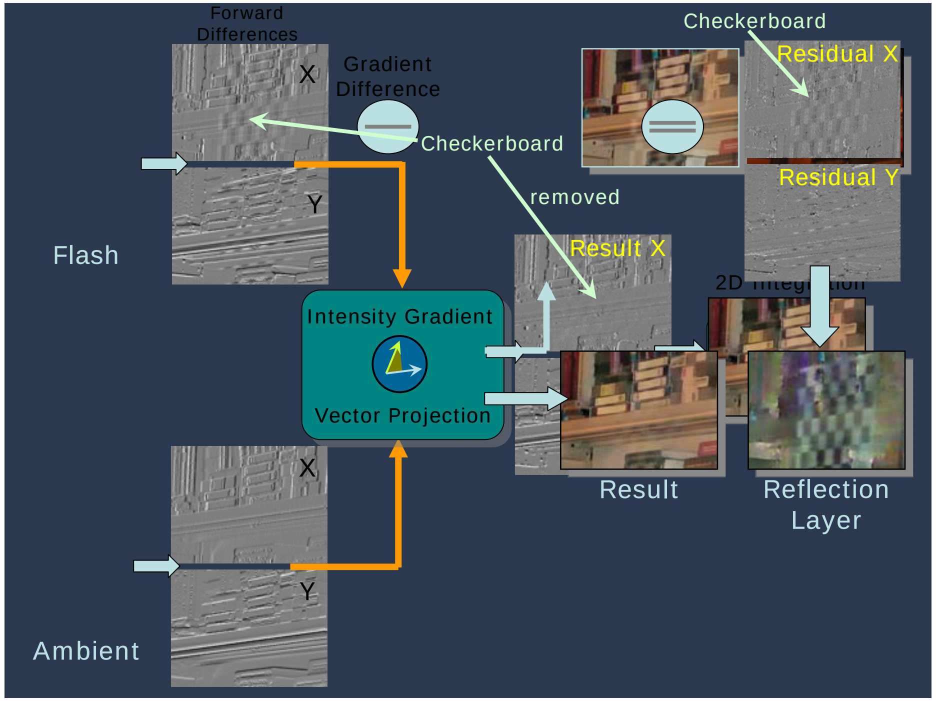



Reflection Ambient Gradient Vector
Different gradient vector direction
Flash Gradient Vector
Ambient
Flash
With reflections


Intensity Gradient Vector Projection
Reflection Ambient Gradient Vector
Residual Gradient Vector
Flash Gradient Vector
Result Gradient Vector
Ambient
Flash
Result
Residual


Ambient
Flash
Projection = Result
Residual = Reflection Layer


Flash
Ambient
Checkerboard outside glass window
Reflections on glass window



DigiVFX

Image Fusion for Context Enhancement and Video Surrealism

Ramesh Raskar

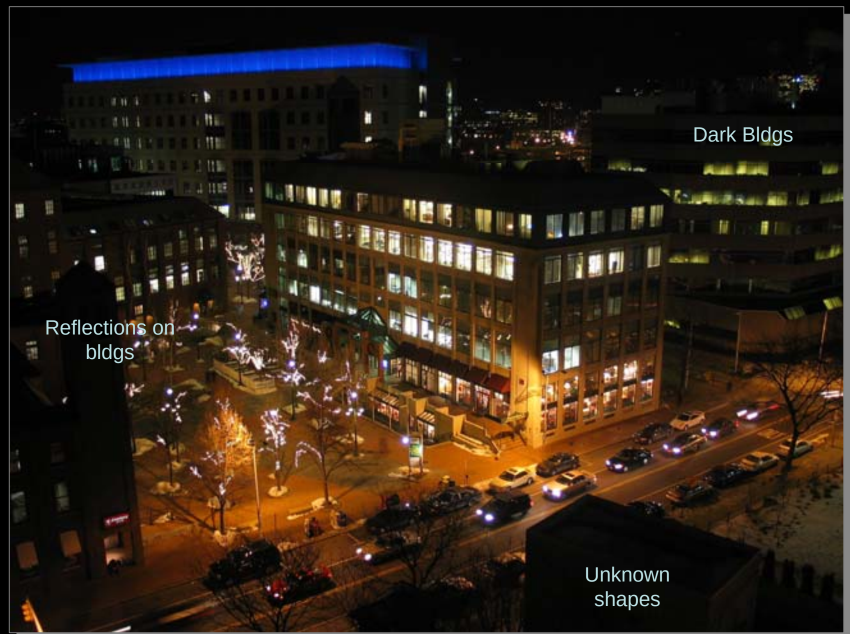
Mitsubishi Electric Research Labs, (MERL)

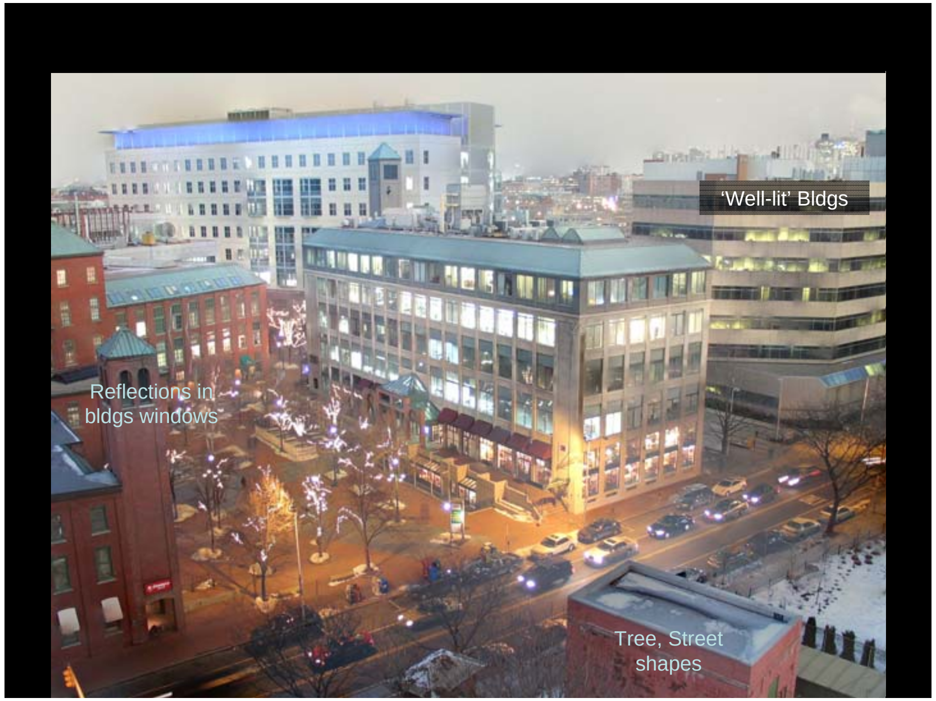
Adrian Ilie

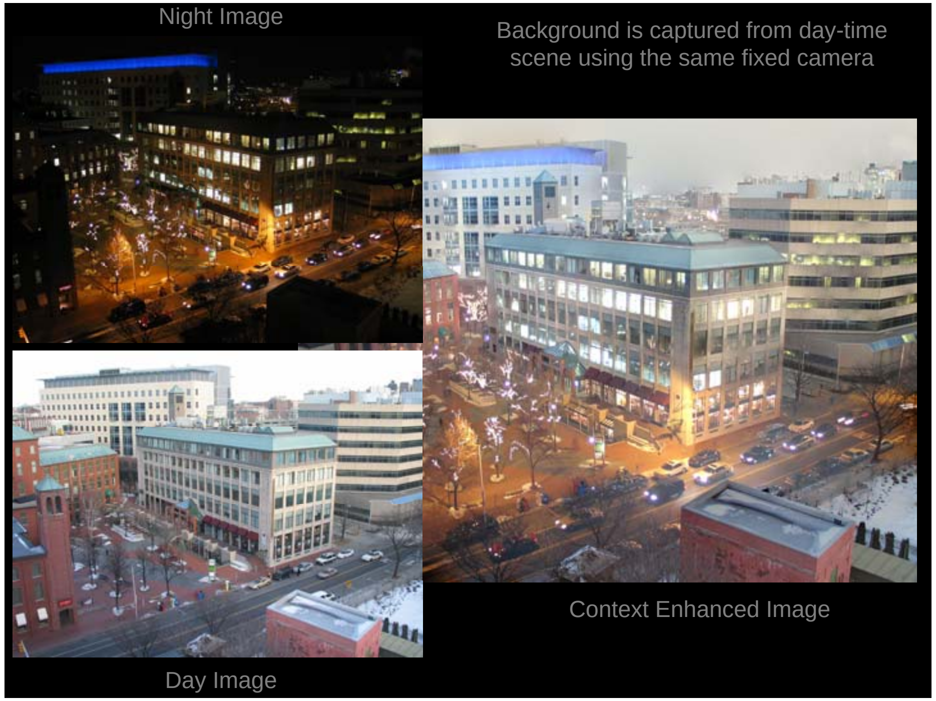
UNC Chapel Hill

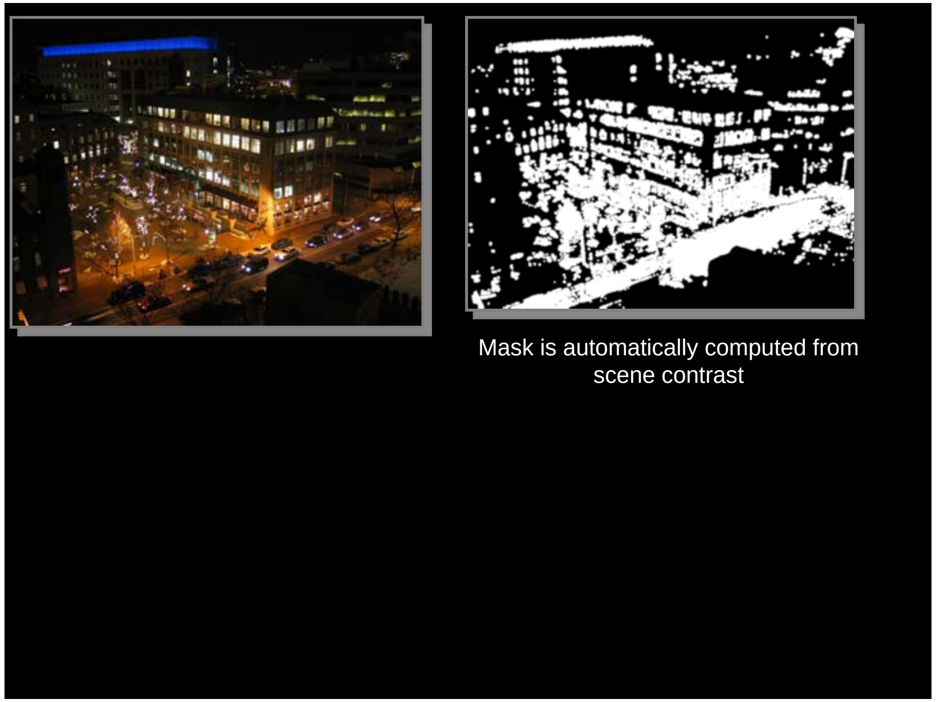
Jingyi Yu

MIT

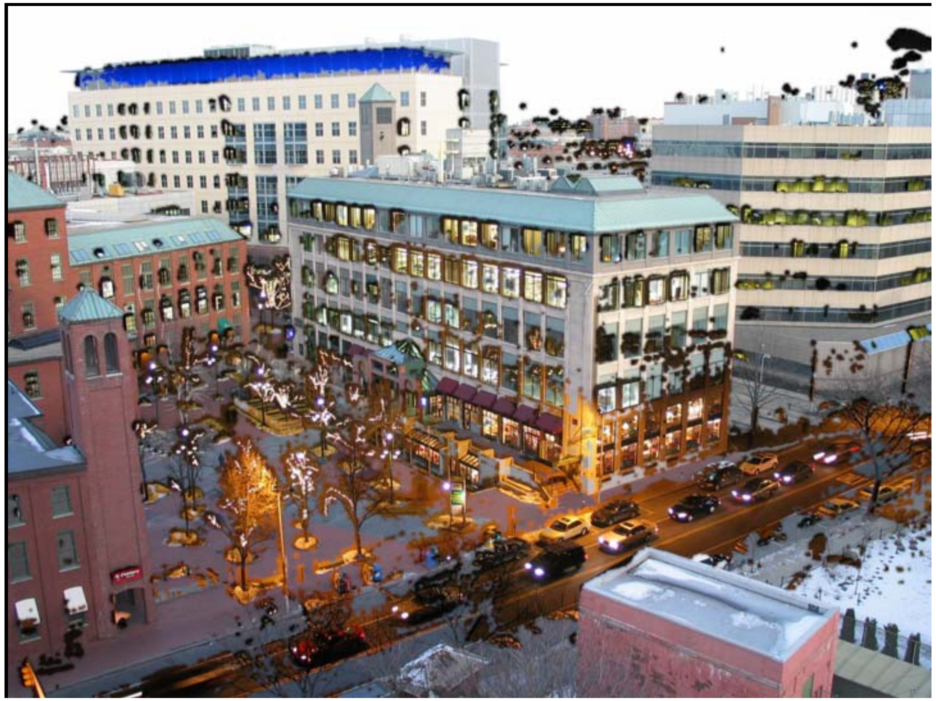


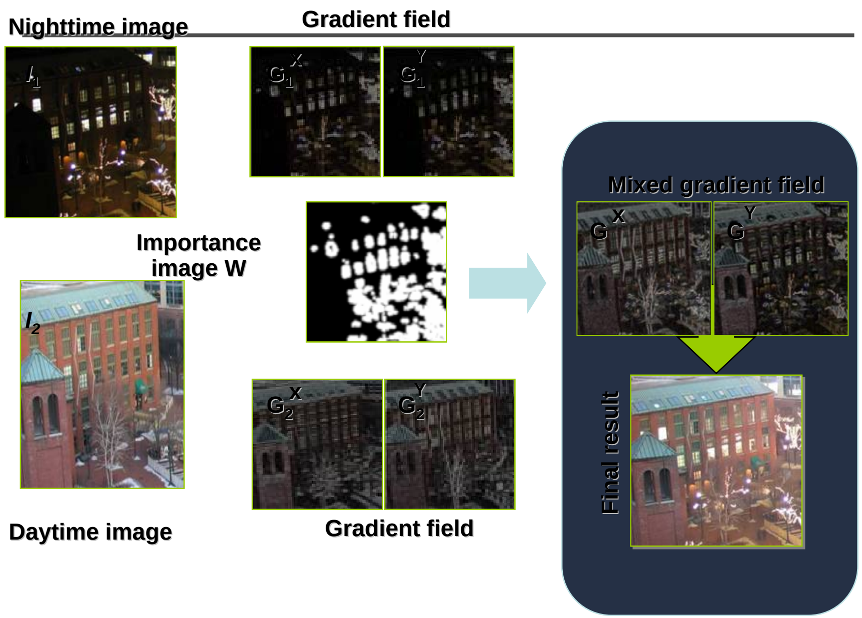

'Well-lit' Bldgs
Reflections in bldgs windows
Tree, Street shapes


Night Image
Background is captured from day-time scene using the same fixed camera
Context Enhanced Image
Day Image


Mask is automatically computed from scene contrast


But, Simple Pixel Blending Creates Ugly Artifacts





Reconstruction from Gradient Field

- Problem: minimize error $|\nabla I' - G|$
- Estimate I' so that

$$G = \nabla I'$$

- Poisson equation

$$\nabla^2 I' = div\ G$$

- Full multigrid solver

Poisson Image Editing: Inserting Objects

- Precise selection: tedious and unsatisfactory
- Alpha-Matting: powerful but involved
- Seamless cloning: loose selection but no seams?

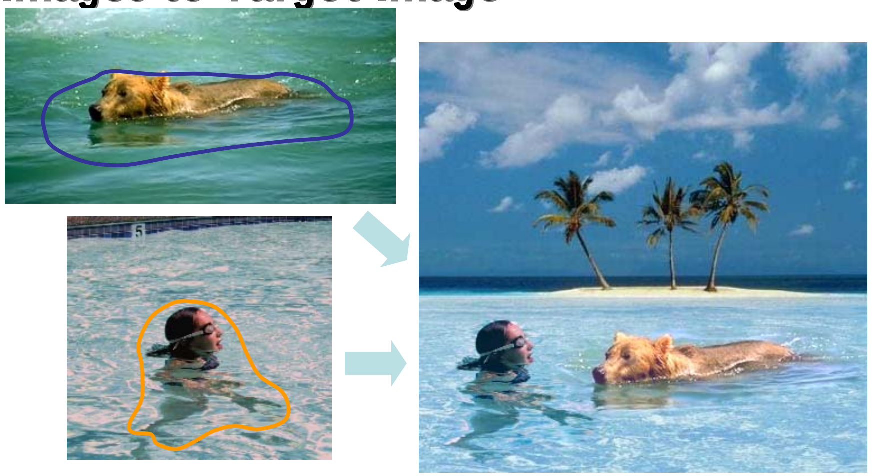
Smooth Correction: Copying Gradients

[redacted]

Conceal

Copy Background gradients (user strokes)

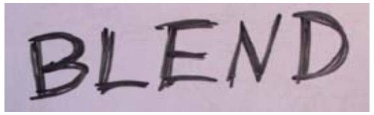
Compose: Copy gradients from Source Images to Target Image

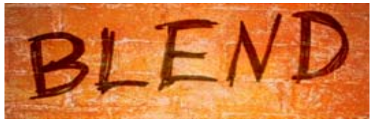


Source Images

Target Image

Transparent Cloning

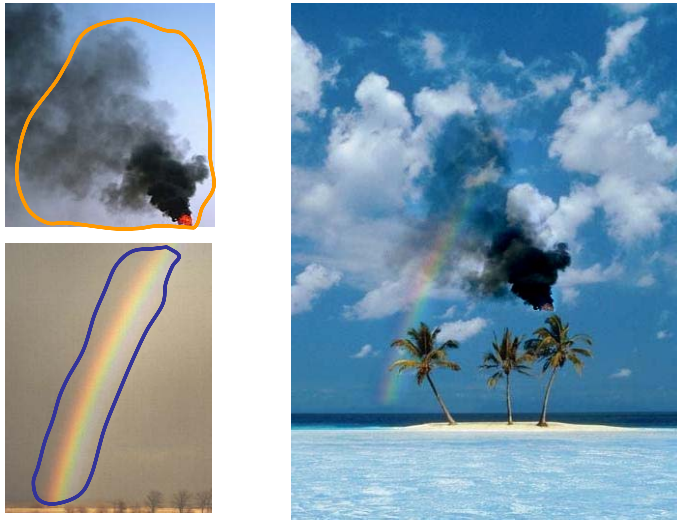




$$\mathbf{v} = \frac{\nabla f^* + \nabla g}{2}$$

Largest variation from source and destination at each point

Compose (transparent)



Gradient Domain Manipulations: Overview

(A) Per pixel

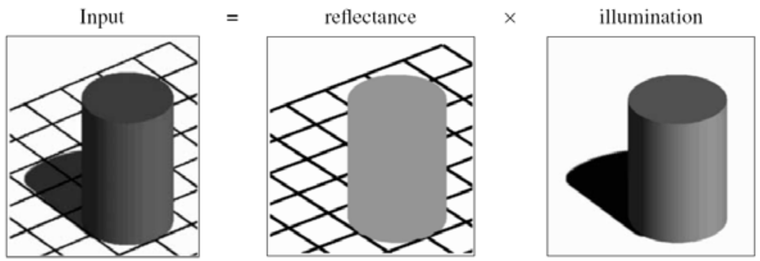
(B) Corresponding gradients in two images

(C) Corresponding gradients in multiple images

(D) Combining gradients along seams

Intrinsic images: Median of Gradient operator

- I = L * R
- L = illumination image
- R = reflectance image



Intrinsic images

– Use multiple images under different illumination
– Assumption
  • Illumination image gradients = Laplacian PDF
  • Under Laplacian PDF, Median = ML estimator

– At each pixel, take Median of gradients across images
– Integrate to remove shadows

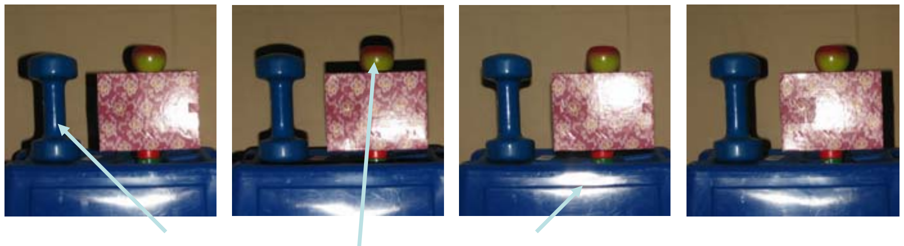
Yair Weiss, "Deriving intrinsic images from image sequences", ICCV 2001

frame 1 frame 11 ML reflectance

Shadow free
Intrinsic Image

Result = Illumination Image * (Label in Intrinsic Image)

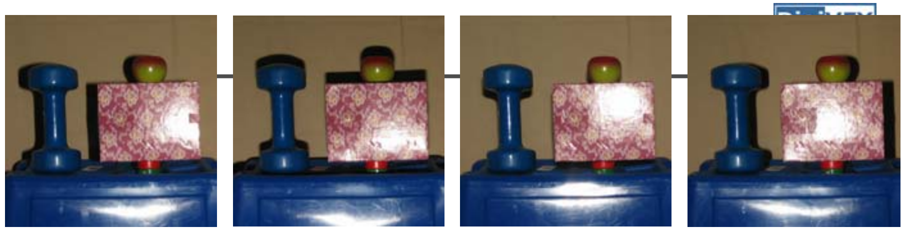
Specularity Reduction in Active Illumination



Line Specularity Point Specularity Area Specularity

Multiple images with same viewpoint, varying illumination

How do we remove highlights?



Specularity Reduced Image

Gradient Domain Manipulations: Overview

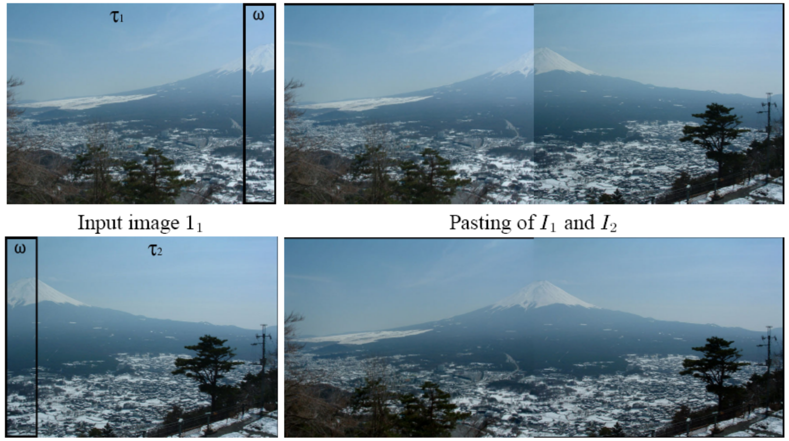
(A) Per pixel

(B) Corresponding gradients in two images

(C) Corresponding gradients in multiple images

(D) Combining gradients along seams

Seamless Image Stitching



Input image I_1 — Pasting of I_1 and I_2

Input image I_2 — Stitching result

Anat Levin, Assaf Zomet, Shmuel Peleg and Yair Weiss, "Seamless Image Stitching in the Gradient Domain", ECCV 2004