Image Composition

© NASA

Modeled after lecture by Alexei Efros.
Slides by Efros, Durand, Freeman, Hays, Fergus, Lazebnik, Agarwala, Shamir, and Perez.

Image Compositing

Slide credit: A. Efros

Compositing Procedure

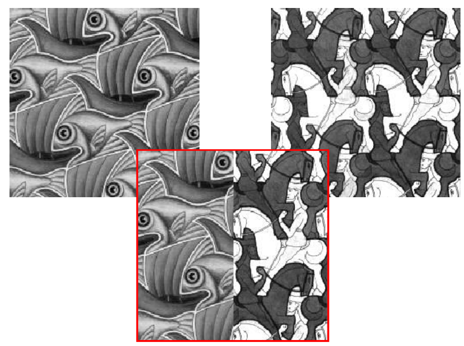
1. Extract Sprites (e.g using *Intelligent Scissors* in Photoshop)

2. Blend them into the composite (in the right order)

Composite by
David Dewey

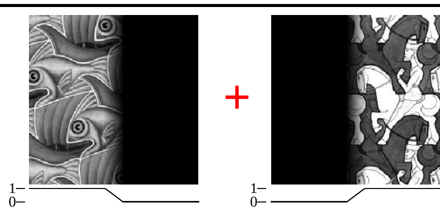
Slide credit: A. Efros

Need blending



Slide credit: A. Efros

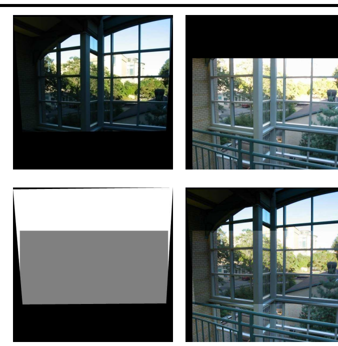
Alpha Blending / Feathering



$$I_{blend} = \alpha I_{left} + (1-\alpha) I_{right}$$

Slide credit: A. Efros

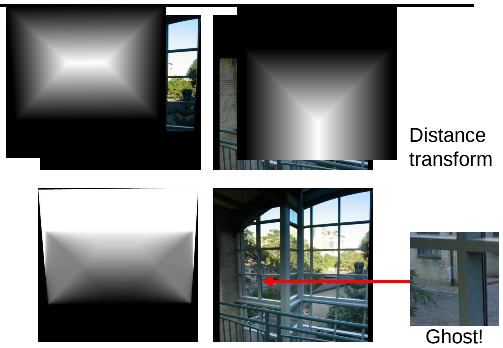
Setting alpha: simple averaging



Alpha = .5 in overlap region

Slide credit: A. Efros

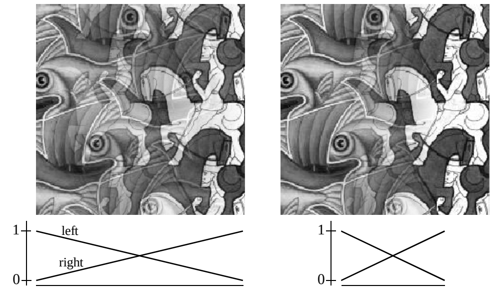
Setting alpha: center weighting



Alpha = dtrans1 / (dtrans1+dtrans2)

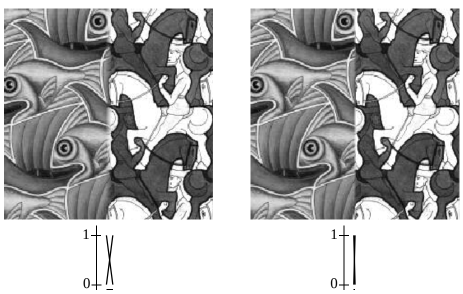
Slide credit: A. Efros

Affect of Window Size



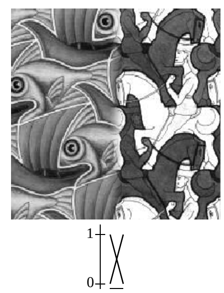
Slide credit: A. Efros

Affect of Window Size



Slide credit: A. Efros

Good Window Size



"Optimal" Window: smooth but not ghosted

Slide credit: A. Efros

What is the Optimal Window?

To avoid seams

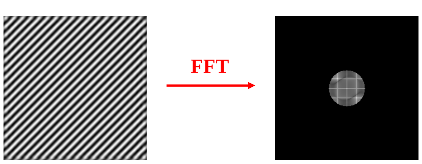
- window = size of largest prominent feature

To avoid ghosting

- window <= 2*size of smallest prominent feature

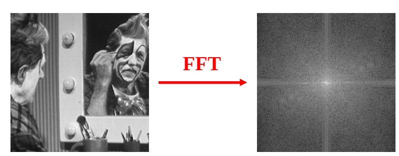
Natural to cast this in the *Fourier domain*

- largest frequency <= 2*size of smallest frequency
- image frequency content should occupy one "octave" (power of two)



Slide credit: A. Efros

What if the Frequency Spread is Wide



Idea (Burt and Adelson)

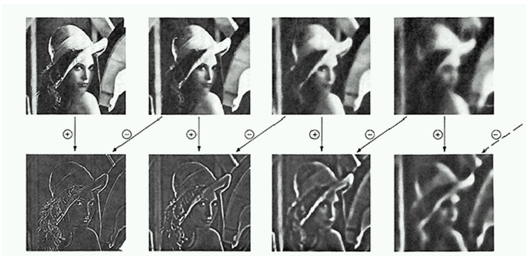
- Compute $F_{left} = FFT(I_{left})$, $F_{right} = FFT(I_{right})$
- Decompose Fourier image into octaves (bands)
  - $F_{left} = F_{left}^1 + F_{left}^2 + \ldots$
- Feather corresponding octaves F_{left}^i with F_{right}^i
  - Can compute inverse FFT and feather in spatial domain
- Sum feathered octave images in frequency domain

Better implemented in *spatial domain*

Slide credit: A. Efros

Octaves in the Spatial Domain

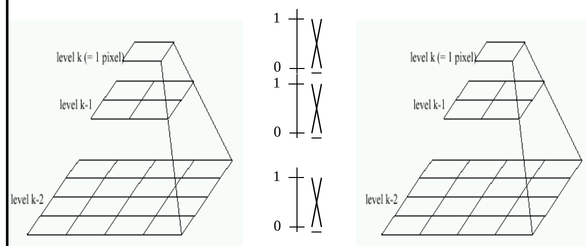
Lowpass Images



Bandpass Images

Slide credit: A. Efros

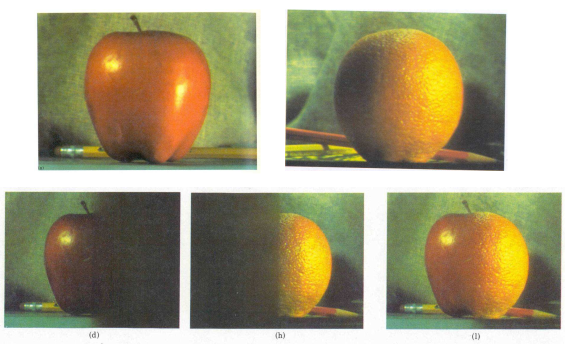
Laplacian Pyramid Blending



Left pyramid | blend | Right pyramid

Slide credit: A. Efros

Laplacian Pyramid Blending

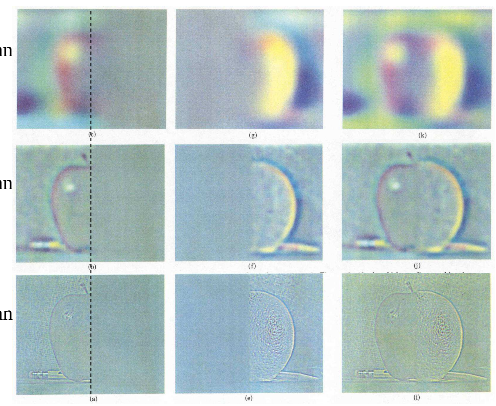


Slide credit: A. Efros

laplacian level 4

laplacian level 2

laplacian level 0



left pyramid | right pyramid | blended pyramid

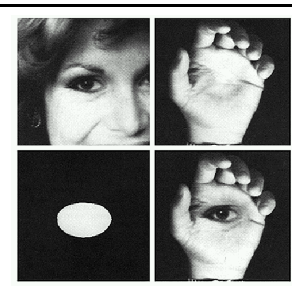
Laplacian Pyramid Blending

General Approach:

1. Build Laplacian pyramids *LA* and *LB* from images *A* and *B*
2. Build a Gaussian pyramid *GR* from selected region *R*
3. Form a combined pyramid *LS* from *LA* and *LB* using nodes of *GR* as weights:
   - *LS(i,j) = GR(I,j,)*LA(I,j) + (1-GR(I,j))*LB(I,j)*
4. Collapse the *LS* pyramid to get the final blended image

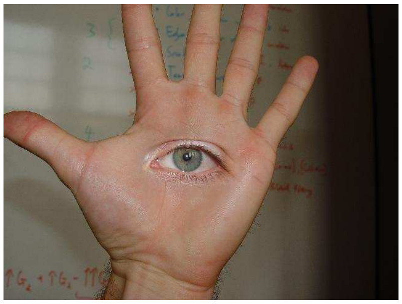
Slide credit: A. Efros

Laplacian Pyramid Blending



Slide credit: A. Efros

Laplacian Pyramid Blending



© david dmartin (Boston College)

Slide credit: A. Efros

Problems with blending

Misaligned (moving) objects become ghosts

Slide credit: A. Efros

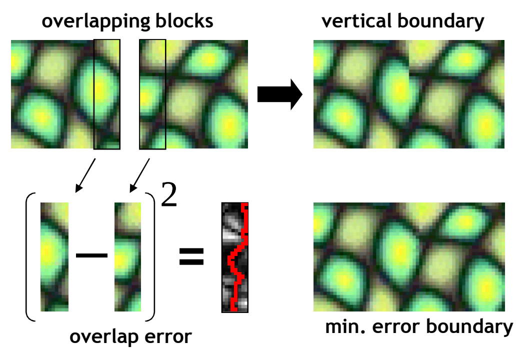
Seams

Segment the images

- Single source image per segment
- Find optimal seams between segments
- Optionally blend across seams

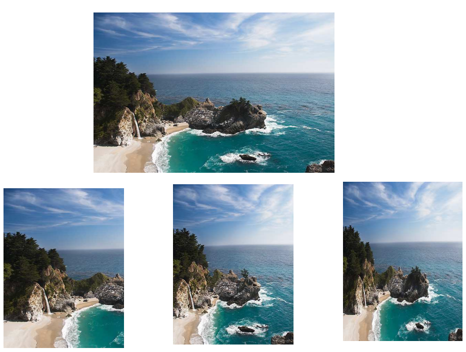
Slide credit: A. Efros

Seams in texture synthesis



Slide credit: A. Efros

Seam Carving



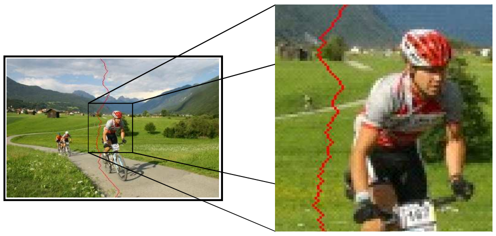
Cropping *Seam Carving* *Scaling*

Shamir

Seam Carving

Shamir

Seam Carving

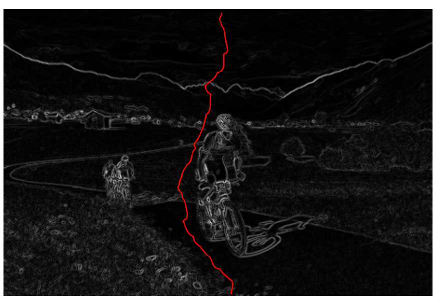


Shamir

Seam Carving

Shamir

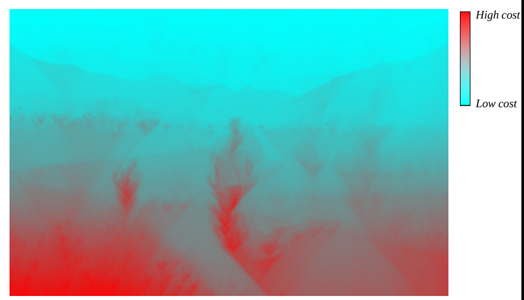
Seam Carving



$$E(\mathbf{I}) = |\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}| \Rightarrow s^* = \arg\min_s E(s)$$

Shamir

Seam Carving



Shamir

Seam Carving

Dynamic programming

Removal of ***vertical*** *seams*

Removal of ***horizontal*** *seams*

0	13	16	19		
16	17	22	28		
19	31	25	35		
24	28	29	???		
32	35	33			
41	38	35			

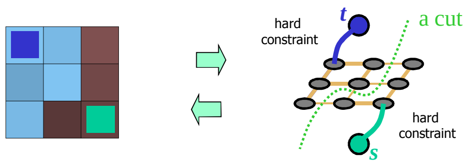
Shamir

Seams with Graphcuts

What if we want similar “cut-where-things-agree” idea, but for closed regions?

- Dynamic programming can’t handle loops

Seams with Graph cuts

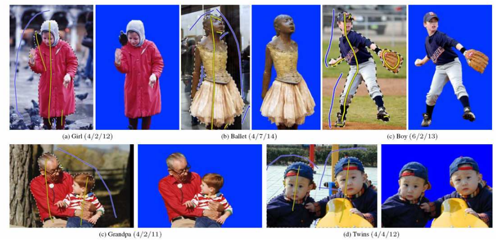


Minimum cost cut can be computed in polynomial time

(max-flow/min-cut algorithms)

Boykov&Jolly, ICCV'01

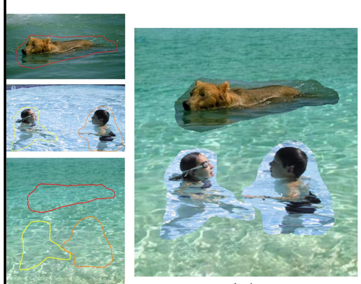
Seams with Graph Cuts



Lazy Snapping
Interactive segmentation using graphcuts

Problem with seams

What if colors/intensities are different?

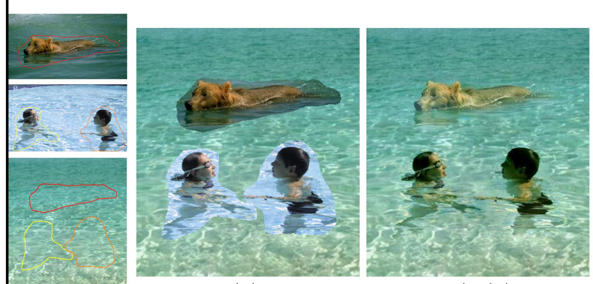


sources/destinations cloning

Slide credit: F. Durand

Problem with seams

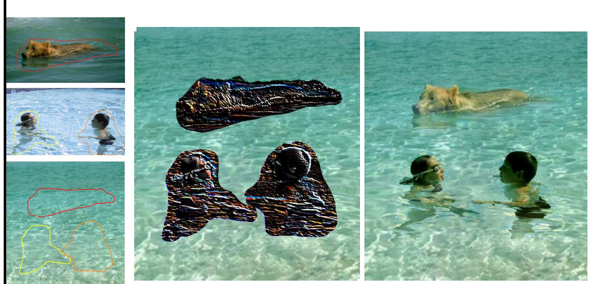
What if colors/intensities are different?



sources/destinations cloning seamless cloning

Slide credit: F. Durand

Gradient domain image editing



sources/destinations cloning seamless cloning

Slide credit: F. Durand

Gradient domain image editing

Motivation:

- Human visual system is very sensitive to gradient
- Gradient encode edges and local contrast quite well

Approach:

- Edit in the gradient domain
- Reconstruct image from gradient

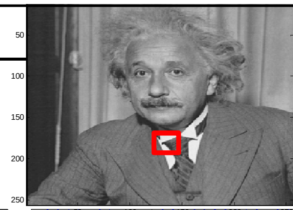
r

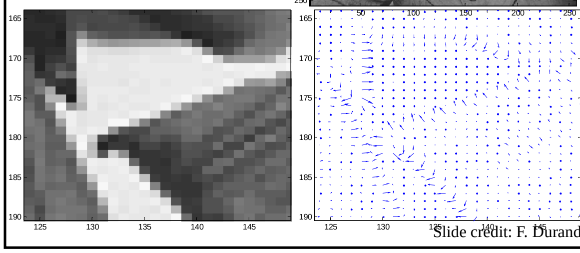
Various instances of this idea, I'll mostly follow Perez et al. Siggraph 2003

http://research.microsoft.com/vision/cambridge/papers/perez_siggraph03.pdf

Slide credit: F. Durand

Gradient domain view of image



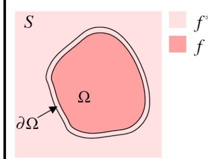


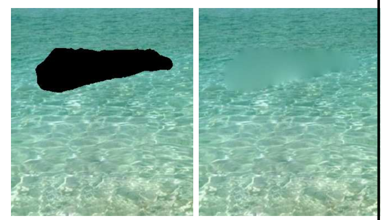
Slide credit: F. Durand

Membrane interpolation

Laplace equation (a.k.a. membrane equation)

$$\min_f \iint_\Omega |\nabla f|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

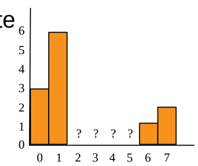




Slide credit: F. Durand

1D example: minimization

Minimize derivatives to interpolate



Min $(f_2-f_1)^2$

Min $(f_3-f_2)^2$

Min $(f_4-f_3)^2$

Min $(f_5-f_4)^2$

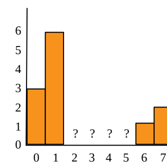
Min $(f_6-f_5)^2$

With
$f_1=6$
$f_6=1$

Slide credit: F. Durand

1D example: derivatives

Minimize derivatives to interpolate



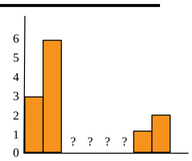
Min ($f_2^2+36-12f_2$
$+ f_3^2+f_2^2-2f_3f_2$
$+ f_4^2+f_3^2-2f_3f_4$
$+ f_5^2+f_4^2-2f_5f_4$
$+ f_5^2+1-2f_5$)
Denote it Q

$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 12$$
$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2f_3 - 2f_4$$
$$\frac{dQ}{df_4} = 2f_4 - 2f_3 + 2f_4 - 2f_5$$
$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2f_5 - 2$$

Slide credit: F. Durand

1D example: set derivatives to zero

Minimize derivatives to interpolate



$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 12$$
$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2f_3 - 2f_4$$
$$\frac{dQ}{df_4} = 2f_4 - 2f_3 + 2f_4 - 2f_5$$
$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2f_5 - 2$$

==> $\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 12 \\ 0 \\ 0 \\ 2 \end{pmatrix}$

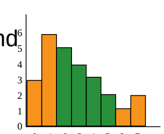
Slide credit: F. Durand

1D example

Minimize derivatives to interpolate

Pretty much says that second derivative should be zero

(-1 2 -1)
is a second derivative filter



$$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 12 \\ 0 \\ 0 \\ 2 \end{pmatrix} \qquad \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 5 \\ 4 \\ 3 \\ 2 \end{pmatrix}$$

Slide credit: F. Durand

Membrane interpolation

Laplace equation (a.k.a. membrane equation)

$$\min_f \iint_\Omega |\nabla f|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

Mathematicians will tell you there is an Associated Euler-Lagrange equation:

$$\Delta f = 0 \text{ over } \Omega \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

- Where the Laplacian Δ is similar to -1 2 -1in 1D

Kind of the idea that we want a minimum, so we kind of derive and get a simpler equation

Slide credit: F. Durand

Seamless Poisson cloning

Given vector field *v* (pasted gradient), find the value of *f* in unknown region that optimize:

Previously, v was null

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

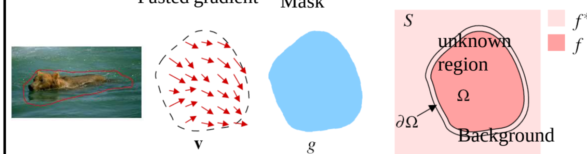


Figure 1: **Guided interpolation notations.** Unknown function f interpolates in domain Ω the destination function f^*, under guidance of vector field **v**, which might be or not the gradi

Slide credit: F. Durand

What if *v* is not null: 2D

Variational minimization (integral of a functional) with boundary condition

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega},$$

Euler-Lagrange equation:

$$\Delta f = \text{div}\mathbf{v} \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

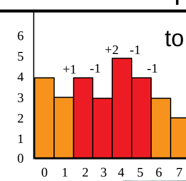
where $\text{div}\mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ is the divergence of $\mathbf{v} = (u, v)$

(Compared to Laplace, we have replaced Δ =0 by Δ = div)

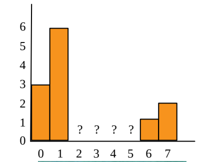
Slide credit: F. Durand

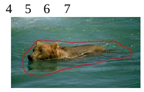
Discrete 1D example: minimization

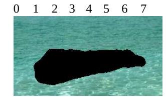
Copy



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

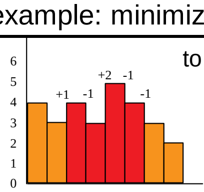
$f_1=6$

$f_6=1$

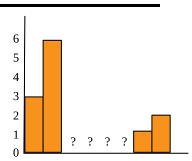
Slide credit: F. Durand

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 $((f_4-f_3)-2)^2$ ==> $f_4^2+f_3^2+4-2f_3f_4-4f_4+4f_3$

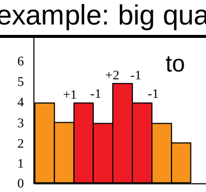
Min $((f_5-f_4)-(-1))^2$ ==> $f_5^2+f_4^2+1-2f_5f_4+2f_5-2f_4$

Min $((1-f_5)-(-1))^2$ ==> $f_5^2+4-4f_5$

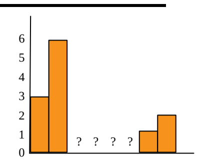
Slide credit: F. Durand

1D example: big quadratic

Copy



to

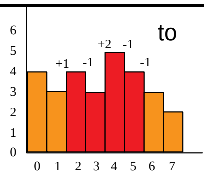
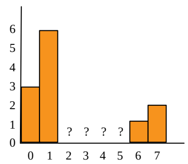


Min $(f_2^2+49-14f_2$
$+ f_3^2+f_2^2+1-2f_3f_2+2f_3-2f_2$
$+ f_4^2+f_3^2+4-2f_3f_4-4f_4+4f_3$
$+ f_5^2+f_4^2+1-2f_5f_4+2f_5-2f_4$
$+ f_5^2+4-4f_5)$

Denote it Q

Slide credit: F. Durand

1D example: derivatives

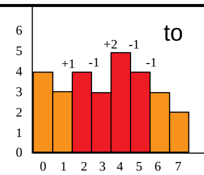
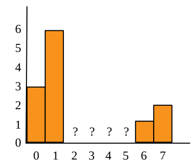
Copy  to 

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

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

Slide credit: F. Durand

1D example: set derivatives to zero

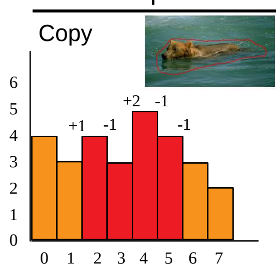
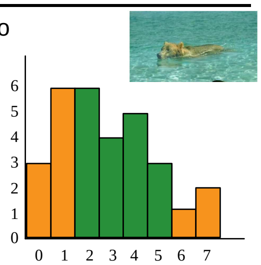
Copy  to 

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

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

Slide credit: F. Durand

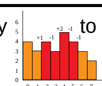
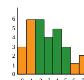
1D example

Copy  to 

$$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix} \qquad \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ 5 \\ 3 \end{pmatrix}$$

Slide credit: F. Durand

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

Slide credit: F. Durand

What if *v* is not null: 2D

Variational minimization (integral of a functional) with boundary condition

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega},$$

Euler-Lagrange equation:

$$\Delta f = \text{div}\mathbf{v} \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

where $\text{div}\mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ is the divergence of $\mathbf{v} = (u, v)$

(Compared to Laplace, we have replaced Δ =0 by Δ = div)

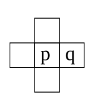
Slide credit: F. Durand

Discrete Poisson solver

Two approaches:
- Minimize variational problem $\quad \min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega},$
- Solve Euler-Lagrange equation $\quad \Delta f = \text{div}\mathbf{v} \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$

In practice, variational is best

In both cases, need to discretize derivatives
- Finite differences over 4 pixel neighbors
- We are going to work using pairs
  - Partial derivatives are easy on pairs
  - Same for the discretization of v



Slide credit: F. Durand

Discrete Poisson solver

Minimize variational problem $\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega},$

$$\min_{f|_\Omega} \sum_{\langle p,q\rangle \cap \Omega \neq \emptyset} (f_p - f_q - v_{pq})^2, \text{ with } f_p = f_p^*, \text{ for all } p \in \partial\Omega$$

Discretized gradient

(all pairs that are in Ω)

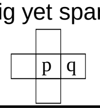
Discretized v: g(p)-g(q)

Boundary condition

Rearrange and call N_p the neighbors of p

$$\text{for all } p \in \Omega, \quad |N_p| f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq}$$

Big yet sparse linear system



Only for boundary pixels

Slide credit: F. Durand

Discrete Poisson solver

Minimize variational problem $\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega},$

$$\min_{f|_\Omega} \sum_{\langle p,q\rangle \cap \Omega \neq \emptyset} (f_p - f_q - v_{pq})^2, \text{ with } f_p = f_p^*, \text{ for all } p \in \partial\Omega$$

Discretized gradient

(all pairs that are in Ω)

Discretized v: g(p)-g(q)

Boundary condition

Rearrange and call N_p the neighbors of p

$$\text{for all } p \in \Omega, \quad |N_p| f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq}$$

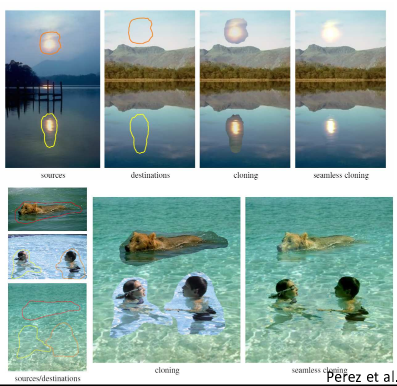
Big yet sparse linear system

p q

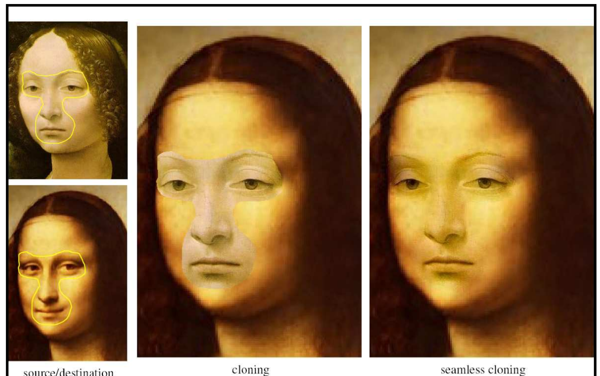
Only for boundary pixels

Slide credit: F. Durand

Image Composition Results



Perez et al. SIGGRAPH 03



Perez et al. SIGGRAPH 03

Figure 2: **Concealment**. By importing seamlessly a piece of the background, complete objects, parts of objects, and undesirable artifacts can easily be hidden. In both examples, multiple strokes (not shown) were used.

Perez et al. SIGGRAPH 03

Figure 5: **Monochrome transfer.** In some cases, such as texture transfer, the part of the source color remaining after seamless cloning might be undesirable. This is fixed by turning the source image monochrome beforehand.

Perez et al. SIGGRAPH 03

Putting it all together

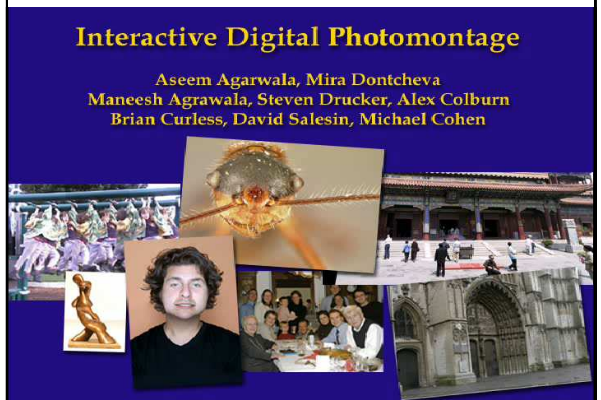
Compositing images

- Have a clever blending function
  - Feathering
  - Center-weighted
  - blend different frequencies differently
  - Gradient based blending
- Choose the right pixels from each image
  - Dynamic programming – optimal seams
  - Graph-cuts

Now, let's put it all together:

- Interactive Digital Photomontage, 2004 (video)

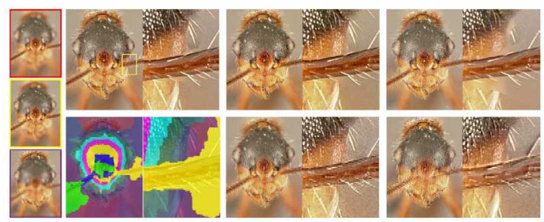
Slide credit: A. Efros



Interactive Digital Photomontage

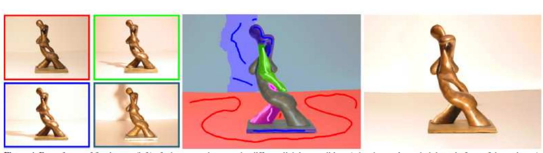
Agarwala et al. SIGGRAPH 04

Interactive Digital Photomontage



Agarwala et al. SIGGRAPH 04

Interactive Digital Photomontage



Agarwala et al. SIGGRAPH 04

Interactive Digital Photomontage

Agarwala et al. SIGGRAPH 04

Scene Completion Using Millions of Photographs

James Hays and Alexei A. Efros
SIGGRAPH 2007

Slides by J. Hays and A. Efros

Hays et al. SIGGRAPH 07

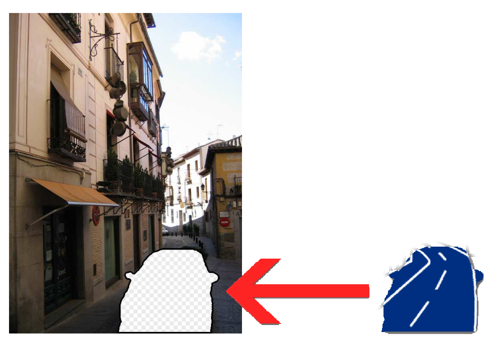
Hays et al. SIGGRAPH 07

Efros and Leung result

Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Scene Matching for Image Completion



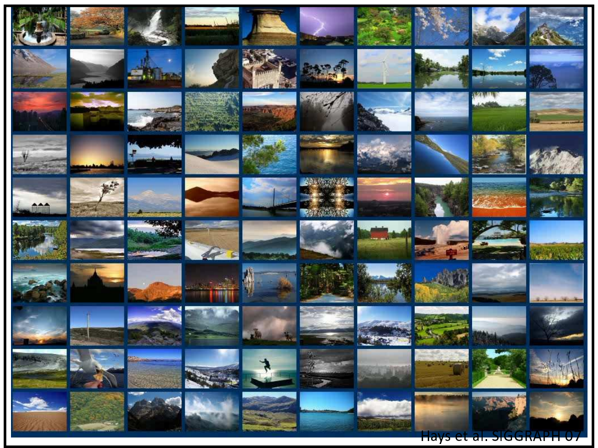
Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Data

2.3 Million unique images from Flickr groups and keyword searches.

Hays et al. SIGGRAPH 07

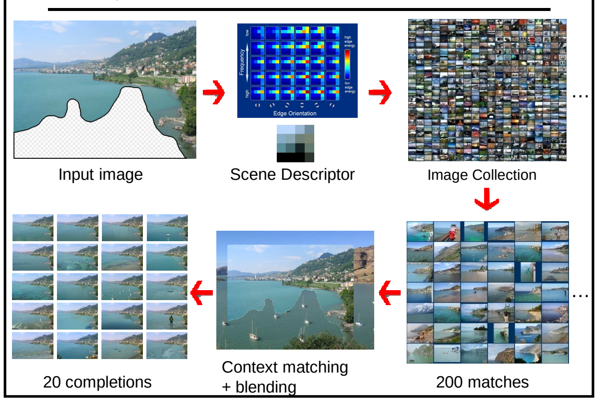


Hays et al. SIGGRAPH 07

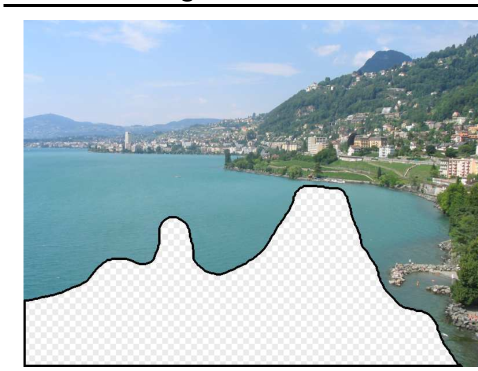
Scene Completion Result

Hays et al. SIGGRAPH 07

The Algorithm

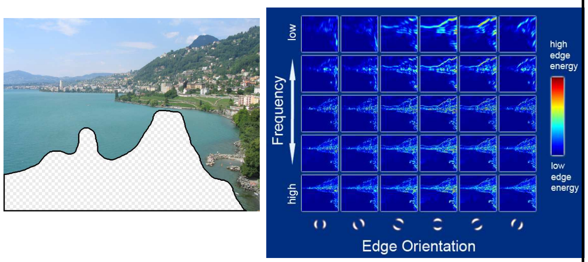


Scene Matching

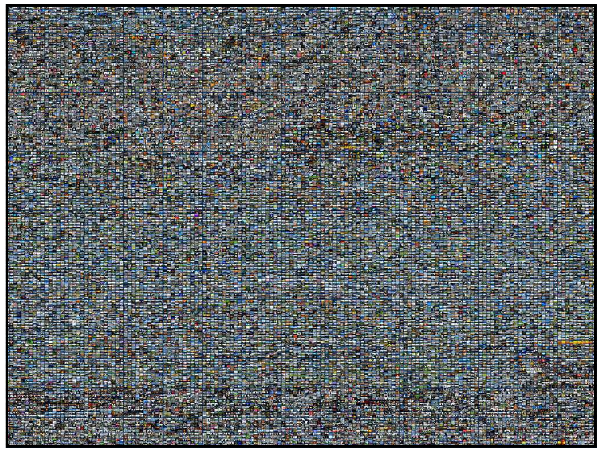


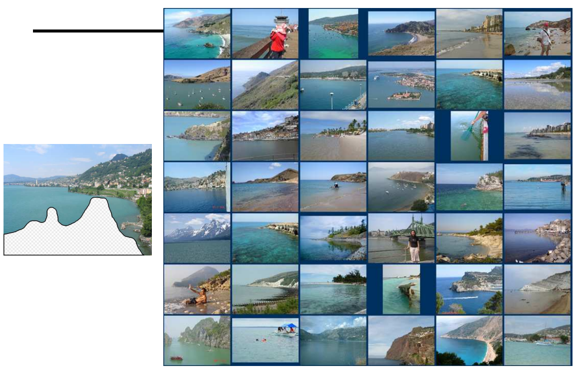
Hays et al. SIGGRAPH 07

Scene Descriptor



Hays et al. SIGGRAPH 07

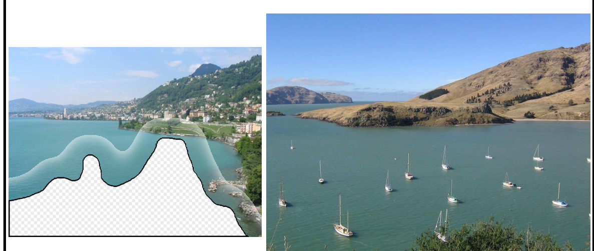




… 200 total

Hays et al. SIGGRAPH 07

Context Matching



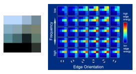
Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Result Ranking

We assign each of the 200 results a score which is the sum of:



The scene matching distance

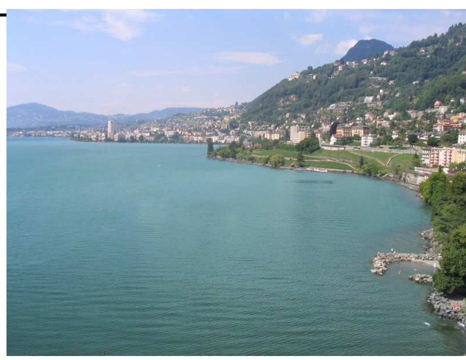
The context matching distance (color + texture)

The graph cut cost

Hays et al. SIGGRAPH 07

Top 20 Results

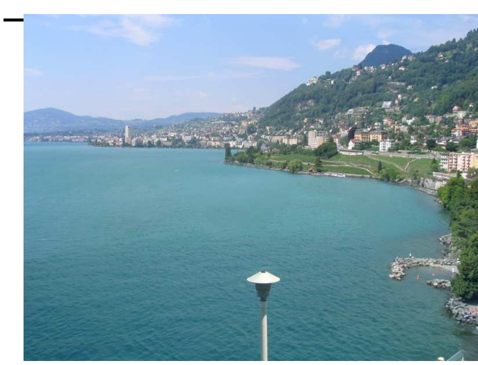
Hays et al. SIGGRAPH 07



Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

Hays et al. SIGGRAPH 07

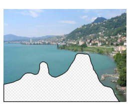


Hays et al. SIGGRAPH 07

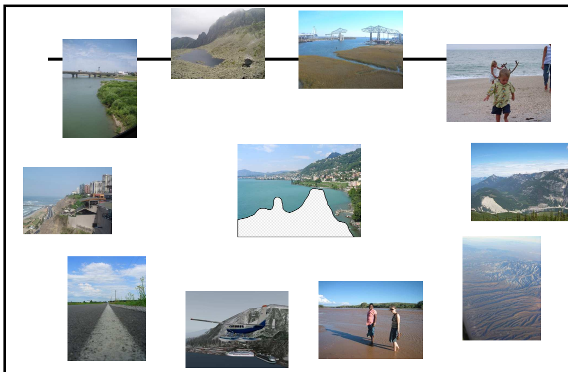
Hays et al. SIGGRAPH 07

Why does it work?

Hays and Efros, SIGGRAPH 2007

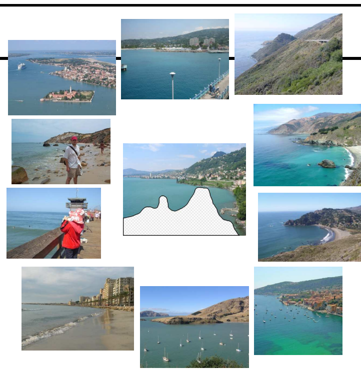


Hays and Efros, SIGGRAPH 2007



10 nearest neighbors from a collection of 20,000 images

Hays and Efros, SIGGRAPH 2007



10 nearest neighbors from a collection of 2 million images

Hays and Efros, SIGGRAPH 2007