Image Blending and Compositing

© NASA
Image Compositing
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
Need blending
Alpha Blending / Feathering

\[ I_{\text{blend}} = \alpha I_{\text{left}} + (1-\alpha)I_{\text{right}} \]
Affect of Window Size
Affect of Window Size
Good Window Size

“Optimal” Window: smooth but not ghosted
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)
What if the Frequency Spread is Wide

Idea (Burt and Adelson)

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

Better implemented in \textit{spatial domain}
Octaves in the Spatial Domain

Lowpass Images

Bandpass Images
Pyramid Blending

Left pyramid

blend

Right pyramid
Pyramid Blending
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:
   \[ LS(i,j) = GR(l,j) * LA(l,j) + (1 - GR(l,j)) * LB(l,j) \]
4. Collapse the $LS$ pyramid to get the final blended image
Blending Regions
Results from this class (fall 2005)
Season Blending (St. Petersburg)
Season Blending (St. Petersburg)
Simplification: Two-band Blending

Brown & Lowe, 2003

- Only use two bands: high freq. and low freq.
- Blends low freq. smoothly
- Blend high freq. with no smoothing: use binary alpha
2-band Blending

Low frequency ($\lambda > 2$ pixels)

High frequency ($\lambda < 2$ pixels)
Linear Blending
2-band Blending
Don’t blend, CUT!

Moving objects become ghosts

So far we only tried to blend between two images. What about finding an optimal seam?
Davis, 1998

Segment the mosaic

- Single source image per segment
- Avoid artifacts along boundaries
  - Dijkstra’s algorithm
Minimal error boundary

overlapping blocks

vertical boundary

\[ \text{overlap error} \]

\[ \text{min. error boundary} \]
Seam Carving

Seam Carving for Content-Aware Image Resizing

Shai Avidan
Mitsubishi Electric Research Labs

Ariel Shamir
The Interdisciplinary Center & MERL

http://www.youtube.com/watch?v=6NclJXTlugo
Graphcuts

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

- Dynamic programming can’t handle loops
Graph cuts – a more general solution

Minimum cost cut can be computed in polynomial time
(max-flow/min-cut algorithms)
Actually, for this example, DP will work just as well…
Lazy Snapping

Interactive segmentation using graphcuts
Gradient Domain

In Pyramid Blending, we decomposed our image into 2nd derivatives (Laplacian) and a low-res image

Let us now look at 1st derivatives (gradients):

- No need for low-res image
  - captures everything (up to a constant)

- Idea:
  - Differentiate
  - Blend / edit / whatever
  - Reintegrate
Gradient Domain blending (1D)

Two signals

Regular blending

Blending derivatives
Gradient Domain Blending (2D)

Trickier in 2D:

- Take partial derivatives $dx$ and $dy$ (the gradient field)
- Fiddle around with them (smooth, blend, feather, etc)
- Reintegrate
  - But now integral($dx$) might not equal integral($dy$)
- Find the most agreeable solution
  - Equivalent to solving Poisson equation
  - Can be done using least-squares
Perez et al., 2003
Limitations:

- Can’t do contrast reversal (gray on black -> gray on white)
- Colored backgrounds “bleed through”
- Images need to be very well aligned
Gradients vs. Pixels

Can we use this for range compression?
Thinking in Gradient Domain

Real-Time Gradient-Domain Painting

James McCann*
Carnegie Mellon University

Nancy S. Pollard†
Carnegie Mellon University

Our very own Jim McCann::

James McCann
Real-Time Gradient-Domain Painting,
SIGGRAPH 2009
Gradient Domain as Image Representation

See GradientShop paper as good example:

GradientShop: A Gradient-Domain Optimization Framework for Image and Video Filtering

Pravin Bhat\textsuperscript{1} C. Lawrence Zitnick\textsuperscript{2} Michael Cohen\textsuperscript{1,2} Brian Curless\textsuperscript{1}
\\textsuperscript{1}University of Washington \textsuperscript{2}Microsoft Research

http://www.gradientshop.com/
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation

(pixel gradient)
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
      - object boundaries
      - depth discontinuities
      - shadows
      - ...

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image features
    - Edges
    - Texture
      - visual richness
      - surface properties
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
      - lighting
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
      - lighting
      - shape

sculpting the face using shading (makeup)
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
      - lighting
      - shape

sculpting the face using shading (makeup)
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
      - lighting
      - shape

sculpting the face using shading (makeup)
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
      - lighting
      - shape

sculpting the face using shading (makeup)
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
    - Artifacts
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
    - Artifacts
      - noise

sensor noise
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
    - Artifacts
      - noise
      - seams
  - seams in composite images
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
    - Artifacts
      - noise
      - seams
      - compression artifacts

blocking in compressed images
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
    - Artifacts
      - noise
      - seams
      - compression artifacts
  - ringing in compressed images
Motivation for gradient-domain filtering?

• Can be used to exert high-level control over images
  • gradients – give rise to high level image-features
    • Edges
    • Texture
    • Shading
    • Artifacts
      • noise
      • seams
      • compression artifacts
      • flicker

flicker from exposure changes & film degradation
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
• Optimization framework
GradientShop

- Optimization framework
  - Input unfiltered image – $u$
GradientShop

- Optimization framework
  - Input unfiltered image – $u$
  - Output filtered image – $f$
Optimization framework

- Input unfiltered image – \( u \)
- Output filtered image – \( f \)
- Specify desired pixel-differences – \((g^x, g^y)\)

Energy function

\[
\min_f (f_x - g^x)^2 + (f_y - g^y)^2
\]
Optimization framework

- Input unfiltered image – $u$
- Output filtered image – $f$
- Specify desired pixel-differences – $(g^x, g^y)$
- Specify desired pixel-values – $d$

Energy function

$$
\min_{f} \quad (f_x - g^x)^2 + (f_y - g^y)^2 + (f - d)^2
$$
• Optimization framework
  • Input unfiltered image – \( u \)
  • Output filtered image – \( f \)
  • Specify desired pixel-differences – \( (g_x, g_y) \)
  • Specify desired pixel-values – \( d \)
  • Specify constraints weights – \( (w_x, w_y, w_d) \)

Energy function

\[
\min_f w_x (f_x - g_x)^2 + w_y (f_y - g_y)^2 + w_d (f - d)^2
\]
GradientShop

Inputs

\( u \)

\( u_x \)

\( u_y \)
GradientShop

Inputs

Application specific filtering

Constraints

Least squares solver

Solution - f
Least Squares Example

Say we have a set of data points \((X_1, X_1'), (X_2, X_2'), (X_3, X_3'), \) etc. (e.g. person’s height vs. weight)

We want a nice compact formula (a line) to predict \(X\)’s from \(X\)s:

\[ Xa + b = X' \]

We want to find \(a\) and \(b\)

How many \((X,X')\) pairs do we need?

\[
\begin{align*}
X_1a + b &= X_1' \\
X_2a + b &= X_2'
\end{align*}
\]

What if the data is noisy?

\[
\begin{bmatrix}
X_1 \\
X_2 \\
\vdots
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix} = \begin{bmatrix}
X_1' \\
X_2' \\
\vdots
\end{bmatrix}
\]

min \(\|Ax - B\|^2\)

overconstrained
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)
Interactive Digital Photomontage

Aseem Agarwala, Mira Dontcheva
Maneesh Agrawala, Steven Drucker, Alex Colburn
Brian Curless, David Salesin, Michael Cohen