Image Blending and Compositing

CS194: Image Manipulation & Computational Photography
Alexei Efros, UC Berkeley, Fall 2014
Image Compositing
Compositing Procedure

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

2. Blend them into the composite (in the right order)
Pyramid Blending
Gradient Domain vs. Frequency Domain

In Pyramid Blending, we decomposed our images into several frequency bands, and transferred them separately

• But boundaries appear across multiple bands

But what representation based on 1st derivatives (gradients) of the image?:

• Represents local change (across all frequencies)
• No need for low-res image
  – captures everything (up to a constant)
• Blending/Editing in Gradient Domain:
  – Differentiate
  – Blend / edit / whatever
  – Reintegrate
Gradients vs. Pixels

Craik-O’Brien Cornsweet Effect

Actual Luminance Profile  Perceived Luminance Profile
Gilchrist Illusion
(c.f. Exploratorium)
White?
White?
James McCann & Nancy Pollard
Real-Time Gradient-Domain Painting, SIGGRAPH 2009
(paper came out of this class!)

http://www.youtube.com/watch?v=RvhkAfrA0-w&feature=youtu.be
Gradient Domain blending (1D)

Two signals

Regular blending

Blending derivatives

bright

dark
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 (\ in Matlab)
What’s the difference?

gradient domain blending

- =

no blending
Limitations:

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

Pixel gradient
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
GradientShop

- Optimization framework

Pravin Bhat et al
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

\[
\min_{f} \quad w^x(f_x - g^x)^2 + w^y(f_y - g^y)^2 + w^d(f - d)^2
\]
Inputs

Application specific filtering

Constraints

Least squares solver

Solution - f
Pseudo image relighting

- change scene illumination in post-production
- example

input
Pseudo image relighting

- change scene illumination in post-production
- example
Pseudo image relighting

- change scene illumination in post-production
- example
Pseudo image relighting

- change scene illumination in post-production
- example

GradientShop relight
Pseudo image relighting

- change scene illumination in post-production
- example

GradientShop relight
Pseudo image relighting

- change scene illumination in post-production
- example

GradientShop relight
Pseudo image relighting

- change scene illumination in post-production
- example

GradientShop relight
Pseudo image relighting
Pseudo image relighting

Energy function

$$\min_{f} \quad w^x(f_x - g_x)^2 + w^y(f_y - g_y)^2 + w^d(f - d)^2$$
Pseudo image relighting

Energy function

\[
\min_f \left( w^x(f_x - g^x)^2 + w^v(f_y - g^y)^2 + w^d(f - d)^2 \right)
\]

- Definition:
  - \( d = u \)
Pseudo image relighting

Energy function

\[
\min_{f} \sum_{x} w_x (f_x - g_x)^2 + \sum_{y} w_y (f_y - g_y)^2 + \sum_{d} w_d (f - d)^2
\]

- Definition:
  - \(d = u\)
  - \(g_x(p) = u_x(p) \times (1 + a(p))\)
  - \(a(p) = \max(0, -\nabla u(p).o(p))\)
Pseudo image relighting

Energy function

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

Definition:
- $$d = u$$
- $$g^x(p) = u_x(p) \cdot (1 + a(p))$$
- $$a(p) = \max(0, -\nabla u(p).o(p))$$
Sparse data interpolation

- Interpolate scattered data over images/video
Sparse data interpolation

- Interpolate scattered data over images/video
- Example app: Colorization*

*Levin et al. – SIGGRAPH 2004
Sparse data interpolation

\[ u \quad \text{user data} \]

\[ f \]
Sparse data interpolation

Energy function

$$\min_{f} \ w^x (f_x - g_x)^2 + w^y (f_y - g_y)^2 + w^d (f - d)^2$$
Sparse data interpolation

Energy function

\[
\min_{f} \quad w^x (f_x - g^x)^2 + w^y (f_y - g^y)^2 + w^d (f - d)^2
\]

- Definition:
  - \(d = \text{user\_data}\)
Sparse data interpolation

Energy function

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

- **Definition:**
  - \( d = \text{user\_data} \)
  - if \( \text{user\_data}(p) \) defined
    \( w^d(p) = 1 \)
  - else
    \( w^d(p) = 0 \)
Sparse data interpolation

Energy function

\[
\min_f \quad \sum w^x (f_x - g^x)^2 + \sum w^y (f_y - g^y)^2 + w^d (f - d)^2
\]

- **Definition:**
  - \( d = \text{user\_data} \)
  - if \( \text{user\_data}(p) \) defined
    - \( w^d(p) = 1 \)
  - else
    - \( w^d(p) = 0 \)
  - \( g^x(p) = 0; \ g^y(p) = 0 \)
Sparse data interpolation

Energy function

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

- **Definition:**
  - \( d = \text{user\_data} \)
  - \( \text{if user\_data}(p) \text{ defined} \)
    - \( w_d^d(p) = 1 \)
    - \( \text{else} \)
      - \( w_d^d(p) = 0 \)
  - \( g_x^x(p) = 0; \ g_y^y(p) = 0 \)
  - \( w_x^x(p) = 1/(1 + c^*|u_x(p)|) \)
  - \( w_y^y(p) = 1/(1 + c^*|u_y(p)|) \)
Don’t blend, CUT!

Moving objects become ghosts

So far we only tried to blend between two images. What about finding an optimal seam?
Segment the mosaic

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

overlapping blocks

vertical boundary

overlap error

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=6NclJXTlugc
Seam Carving

• **Basic Idea:** remove unimportant pixels from the image
  – Unimportant = pixels with less “energy”

\[ E_1(I) = |\frac{\partial}{\partial x} I| + |\frac{\partial}{\partial y} I|. \]

• **Intuition for gradient-based energy:**
  – Preserve strong contours
  – Human vision more sensitive to edges – so try remove content from smoother areas
  – Simple, enough for producing some nice results
  – See their paper for more measures they have used
Finding the Seam?

Michael Rubinstein — MIT CSAIL — mrub@mit.edu
The Optimal Seam

\[ E(I) = \left| \frac{\partial}{\partial x} I \right| + \left| \frac{\partial}{\partial y} I \right| \Rightarrow s^* = \arg \min_s E(s) \]
Dynamic Programming

- **Invariant property:**
  - $M(i,j) =$ minimal cost of a seam going through $(i,j)$ (satisfying the seam properties)
Dynamic Programming

\[
M(i, j) = E(i, j) + \min(M(i - 1, j - 1), M(i - 1, j), M(i - 1, j + 1))
\]

<table>
<thead>
<tr>
<th>5</th>
<th>8</th>
<th>12</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>2+5</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>
Dynamic Programming

\[ M(i, j) = E(i, j) + \min(M(i-1, j-1), M(i-1, j), M(i-1, j+1)) \]
Dynamic Programming

\[ M(i, j) = E(i, j) + \min(M(i - 1, j - 1), M(i - 1, j), M(i - 1, j + 1)) \]
Searching for Minimum

- Backtrack (can store choices along the path, but do not have to)
Backtracking the Seam

<table>
<thead>
<tr>
<th>5</th>
<th>8</th>
<th>12</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>7</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
</tbody>
</table>
# Backtracking the Seam

<table>
<thead>
<tr>
<th></th>
<th>5</th>
<th>8</th>
<th>12</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>7</td>
<td>6</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td></td>
</tr>
</tbody>
</table>

Michael Rubinstein — MIT CSAIL — mrub@mit.edu
Backtracking the Seam
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)
e.g. Lazy Snapping

Interactive segmentation using graphcuts

Also see the original Boykov&Jolly, ICCV’01, “GrabCut”, etc, etc, etc.
Putting it all together

Compositing images

• Have a clever blending function
  – Feathering
  – 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

http://www.youtube.com/watch?v=kzV-5135bGA