Lecture: Image Resizing

Juan Carlos Niebles and Ranjay Krishna
Stanford Vision and Learning Lab
Today’s agenda

• Image resizing
  – Seam carving
  – Dynamic programming
• Applications
Display Devices
Content Retargeting
Simple Media Retargeting Operators
Content-aware Retargeting Operators

Content-aware

“Important” content

Content-oblivious
Content-aware Retargeting

Input

Scale

Crop

Content-aware

“less-Important” content
Image Retargeting

• Problem statement:
  – Input Image $I_{nxm}$, and new size $n'x'm'$
  – Output Image $I'$ of size $n'x'm'$ which will be “good representative” of the original image $I$

• To date, no agreed definition, or measure, as to what a good representative is in this context!
Image/Video Retargeting

- In large, we would expect retargeting to:
  1. Adhere to the geometric constraints (display/aspect ratio)
  2. Preserve the important *content* and *structures*
  3. Limit *artifacts*

- Very Ill-posed!
  - How do we define important? Is there a universal ground truth?
  - Would different viewers think the same about a retargeted image?
  - What about artistic impression in the original content?
Importance (Saliency) Measures

• A function $S: p \rightarrow [0,1]$

$\times$ =  

• More sophisticated: attention models, eye tracking (gazing studies), face detectors, ...
General Retargeting Framework

1. Define an energy function $E(I)$ (interest, importance, saliency)

2. Use some operator(s) to change the image $I$

- Setlur et al. [2005]
- Santella et al. [2005]
- Gal et al. [2006]
Previous Retargeting Approaches

• Optimal Cropping Window

• For videos: “Pan and scan”
  Still done manually in the movie industry

Cropping

![Cropping Examples](image-url)
Seam Carving

• Assume $m \times n \rightarrow m \times n'$, $n' < n$ (summarization)

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

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

• 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
Pixel Removal

Optimal

Least-energy pixels (per row)

Least-energy columns
A Seam

• A connected path of pixels from top to bottom (or left to right). Exactly one in each row

\[ s^x = \{ s^x_i \}_{i=1}^n = \{(x(i), i)\}_{i=1}^n, \text{ s.t. } \forall i, |x(i) - x(i - 1)| \leq 1 \]
A Seam

- A connected path of pixels from top to bottom (or left to right). Exactly one in each row

\[
\begin{align*}
  s^x &= \{s^x_i\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n, \text{ s.t. } \forall i, |x(i) - x(i - 1)| \leq 1 \\
  s^y &= \{s^y_j\}_{j=1}^m = \{(j, y(j))\}_{j=1}^m, \text{ s.t. } \forall j, |y(j) - y(j - 1)| \leq 1
\end{align*}
\]
Finding the Seam?
The Optimal Seam

\[ E(I) = |\frac{\partial}{\partial x} I| + |\frac{\partial}{\partial y} I| \rightarrow s^* = \arg\min_s E(s) \]
The Optimal Seam

• The recursion relation

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

• Can be solved efficiently using dynamic programming in \( O(s \cdot n \cdot m) \)
  
  \( (s=3 \text{ in the original algorithm}) \)
Dynamic Programming

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

![Matrix Diagram]

Energy - $E(I,j)$
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

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

Energy - $E(I,j)$
Dynamic Programming

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

Energy - \( E(I, j) \)
Searching for Minimum

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

The seam is backtrack from 14 to 15 to 16.
Backtracking the Seam

\[
\begin{array}{cccc}
5 & 8 & 12 & 3 \\
9 & 7 & 6 & 12 \\
14 & 9 & 10 & 8 \\
14 & 14 & 15 & 16 \\
\end{array}
\]
Backtracking the Seam

```plaintext
5  8  12  3
9  7  6   12
14 9  10  8
14 14 15  16
```
H & V Cost Maps

Horizontal Cost  Vertical Cost

Low cost

High cost
Seam Carving
The Seam-Carving Algorithm

SEAM-CARVING(im,n’) // size(im) = mxn

1. Do (n-n’) times
   2.1. E ← Compute energy map on im
   2.2. s ← Find optimal seam in E
   2.3. im ← Remove s from im
2. Return im

• For vertical resize: transpose the image

• Running time:
  2.1 O(mn) 2.2 O(mn) 2.3 O(mn)
  ⇒O(dmn)  d=n-n’
Changing Aspect Ratio
Changing Aspect Ratio

Original

Seam Carving

Scaling
Changing Aspect ratio

Cropping

Seams

Scaling
Changing Aspect Ratio

Original

Retarget

Scaling
Changing Aspect Ratio

Original

Retarget

Scaling
Example seam carving
Another example
Content Enhancement

How would you use seam carving to do this?
Seam Carving in the Gradient Domain
Questions?

• Q: Will the result be the same if the image is flipped upside down?
  • A: Yes (up to numerical stability)

• Q: Can we improve the running time?
  • A: Yes. by factor (account for locality of operations)
A Local Operator
Questions?

• Q: Will the result be the same if the image is flipped upside down?
  • A: Yes (up to numerical stability)

• Q: Can we improve the running time?
  • A: By factor (local operations)

• Q: What happens to the overall energy in the image during seam carving?
Preserved Energy

While resizing: remove as many low energy pixels and as few high energy pixels!
Preserved Energy

If we measure the average energy of pixels in the image after applying a resizing operator…

…the average should increase!

While resizing: remove as many low energy pixels and as few high energy pixels!
Preserved Energy

Average Pixel Energy

Image Reduction

crop  column  seam  pixel  optimal
Both Dimensions?
• $m \times n \rightarrow m' \times n'$

• Remove horizontal seam first?
• Remove vertical seams first?
• Alternate between the two?
• The optimal order can be found! $\rightarrow$ Dynamic Prog (again)
Retargeting in Both Dimensions

- The recursion relation:

\[
T(r, c) = \min(T(r - 1, c) + E(s^x(I_{n-r+1 \times m-c})), T(r, c - 1) + E(s^y(I_{n-r \times m-c-1}))),
\]

\[
\min_{s^x, s^y, \alpha} \sum_{i=1}^{k} E(\alpha_i s^x_i + (1 - \alpha_i) s^y_i)
\]
Retargeting in Both Dimensions

• The recursion relation:

\[ T(r, c) = \min(T(r - 1, c) + E(s^x(I_{n-r-1\times m-c})), T(r, c - 1) + E(s^y(I_{n-r\times m-c-1}))) \]

\[ \min_{s^x, s^y, \alpha} \sum_{i=1}^{k} E(\alpha_i s^x_i + (1 - \alpha_i) s^y_i) \]
## Optimal Order Map

### Removal of vertical seams

<table>
<thead>
<tr>
<th>0</th>
<th>3</th>
<th>16</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>17</td>
<td>22</td>
<td>28</td>
</tr>
<tr>
<td>19</td>
<td>31</td>
<td>25</td>
<td>35</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>29</td>
<td>???</td>
</tr>
<tr>
<td>32</td>
<td>35</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>38</td>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>

Removal of horizontal seams
Is it optimal...

• ... for removing ONE seam?

• ... for removing multiple seams?
Is it optimal...

• ... for removing ONE seam?

• ... for removing multiple seams?

  – Consider HVV (how many possibly orderings?)
  – Cost(V) on HV not necessarily equal Cost(V) on VH
  – But we keep track of only one: min(HV,VH)...
Image Expansion (Synthesis)
Image Expansion – take 2

Scaling
Enlarged or Reduced?
Combined Insert and Remove

Insert & remove seams

Scaling
Questions?
Multi-Size Images

- We can create a new *representation* of an image that will allow adapting it to different sizes!

1. Precompute all seams once
2. Realtime resizing / transmit with content
Multi-Size Images

- First to be removed
- Last to be removed
Multi-Size Image Representation
Multi-Size Image Representation
2D Multi-Size Representation?

• Alternatives
  – Use seams in one direction, row/column seams in the other direction
  – Compute seams in one direction and use them to constrain seams in the other direction
    - If you can do that you’ll have a nice publication 😊
Auxiliary Energy

- Recall our seam equation

\[ M(i, j) = E(i, j) + \min(M(i - 1, j - 1), M(i - 1, j), M(i - 1, j + 1)) \]
Object Removal
Object Removal
Find the Missing Shoe!
Solution
Limitations

Content

Structure
With face detector

-
With User Constraints
Preserved Energy - Revisited

Average Pixel Energy

Image Reduction

crop  column  seam  pixel  optimal
Inserted Energy

[Diagram showing the concept of inserted energy with images and plots demonstrating the process of seam removal and energy changes.]
Minimize Inserted Energy

• Instead of removing the seam of least energy, remove the seam that inserts the least energy to the image!
Tracking Inserted Energy

- Three possibilities when removing pixel $p_{i,j}$
Pixel $P_{i,j} : \text{Left Seam}$

$$C_L(i, j) = |I(i, j + 1) - I(i, j - 1)| + |I(i - 1, j) - I(i, j - 1)|$$
Pixel $P_{i,j}$: Right Seam

$$C_R(i,j) = |I(i, j + 1) - I(i, j - 1)| + |I(i - 1, j) - I(i, j + 1)|$$
Pixel $P_{i,j}$ : Vertical Seam

$$C_V(i, j) = |I(i, j + 1) - I(i, j - 1)| + |I(i - 1, j + 1) - I(i - 1, j - 1)|$$
Old “Backward” Energy

\[ M(i, j) = E(i, j) + \min \left\{ M(i - 1, j - 1), M(i - 1, j), M(i - 1, j + 1) \right\} \]
New Forward Looking Energy

\[ M(i, j) = \min \left\{ M(i - 1, j - 1) + C_L(i, j), \right. \]
\[ \left. M(i - 1, j) + C_U(i, j), \right. \]
\[ \left. M(i - 1, j + 1) + C_R(i, j) \right\} \]
Results

Input  Backward  Forward

Input  Forward

Backward
Results
Backward vs. Forward
Results
From Images to Videos

• In general, video processing is a much (much!) harder problem

1. Cardinality
   – Suppose 1min of video x 30 fps = 1800 frames
   – Say your algorithm processes an image in 1 minute $\Rightarrow$ 30 hours !!

2. Dimensionality/algorithmic
   – Temporal coherency: human visual system is highly sensitive to motion!
Seam-Carving Video?

- Naive... frame by frame independently
Frame-by-frame Seam-Carving

Let’s check out this video
From 2D to 3D

1D paths in images

2D manifolds in video cubes
Example video retargeting
Object detection + seam carving
References

• Seam Carving for Content-Aware Image Resizing – Avidan and Shamir 2007
• Content-driven Video Retargeting – Wolf et al. 2007
• Improved Seam Carving for Video Retargeting – Rubinstein et al. 2008
• *Optimized Scale*-and-*Stretch* for Image Resizing – Wang et al. 2008
• Summarizing Visual Data Using Bidirectional Similarity – Simakov et al. 2008
• Multi-operator Media Retargeting – Rubinstein et al. 2009
• Shift-Map Image Editing – Pritch et al. 2009
• Energy-Based Image Deformation – Karni et al. 2009

• Seam carving in Photoshop CS4:  