



Lecture 6. Image Resizing

Pattern Recognition and Computer Vision

Guanbin Li,

School of Data and Computer Science, Sun Yat-Sen University

扫码签到



What we will learn today?

- Background
- What is image retargeting
- Seam carving
- Dynamic programming
- Applications

Background

- Display Devices



Background

- Content Retargeting

The screenshot shows the Wikipedia article for "Seam carving". The page title is "Seam carving". Below the title, it says "From Wikipedia, the free encyclopedia". The main content starts with a definition of seam carving as an algorithm for content-aware image resizing. It mentions the creators, Shai Avidan and Ariel Shamir, and their respective organizations, Mitsubishi Electric Research Laboratories (MERL) and the Interdisciplinary Center (IDC). The text explains how the algorithm works by establishing a number of "seams" (paths of least importance) in an image and automatically removes them to reduce image size or inserts them to extend it. It also notes that manually defining areas in which pixels may not be modified is possible, and features the ability to remove whole objects from photographs.

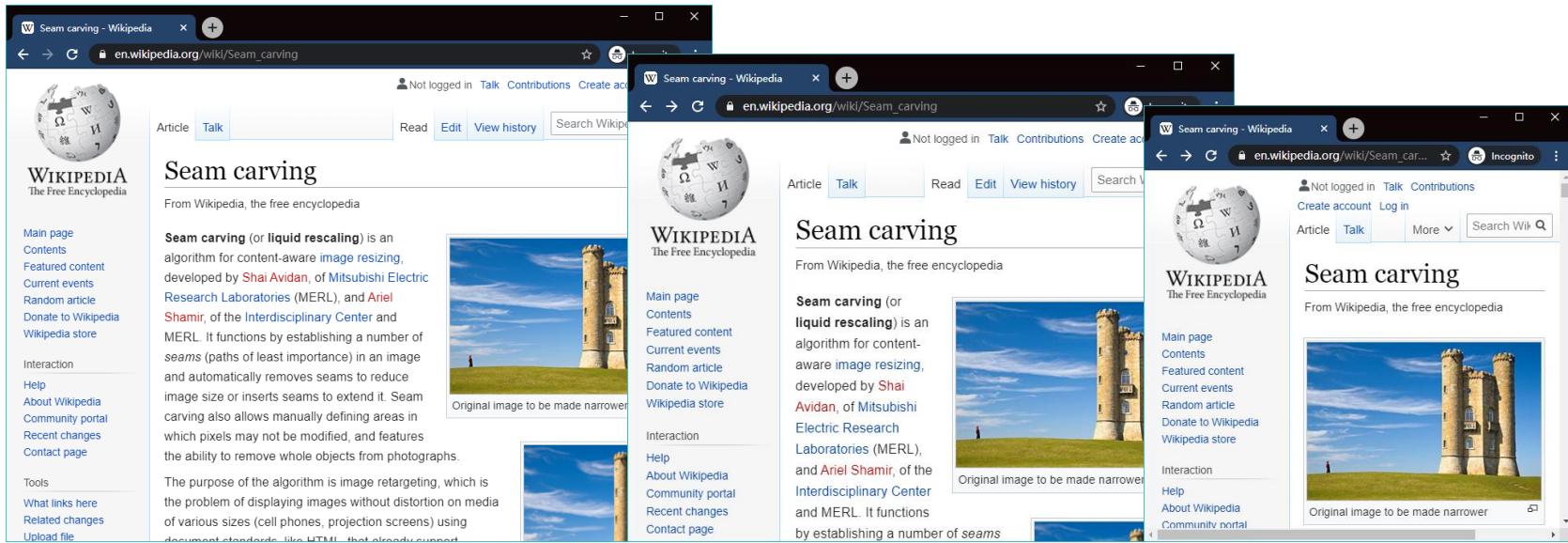
The purpose of the algorithm is image retargeting, which is the problem of displaying images without distortion on media of various sizes (cell phones, projection screens) using document standards, like HTML, that already support dynamic changes in page layout and text but not images [1].

Image Retargeting was invented by Vidya Setlur, Saeko Takage, Ramesh Raskar, Michael Gleicher and Bruce Gooch in 2005 [2]. The work by Setlur et al.

The screenshot shows a mobile browser displaying the same Wikipedia article on "Seam carving". The browser interface includes a header with the URL "en.m.wikipedia.org", a search bar, and a navigation bar at the bottom with icons for back, forward, and search. The article content is identical to the desktop version, featuring the same text and images. The images show a large stone tower (likely Broadway Tower) on a grassy hill under a blue sky with white clouds. The first image is labeled "Original image to be made narrower".

Background

- Text layout can be easily changed

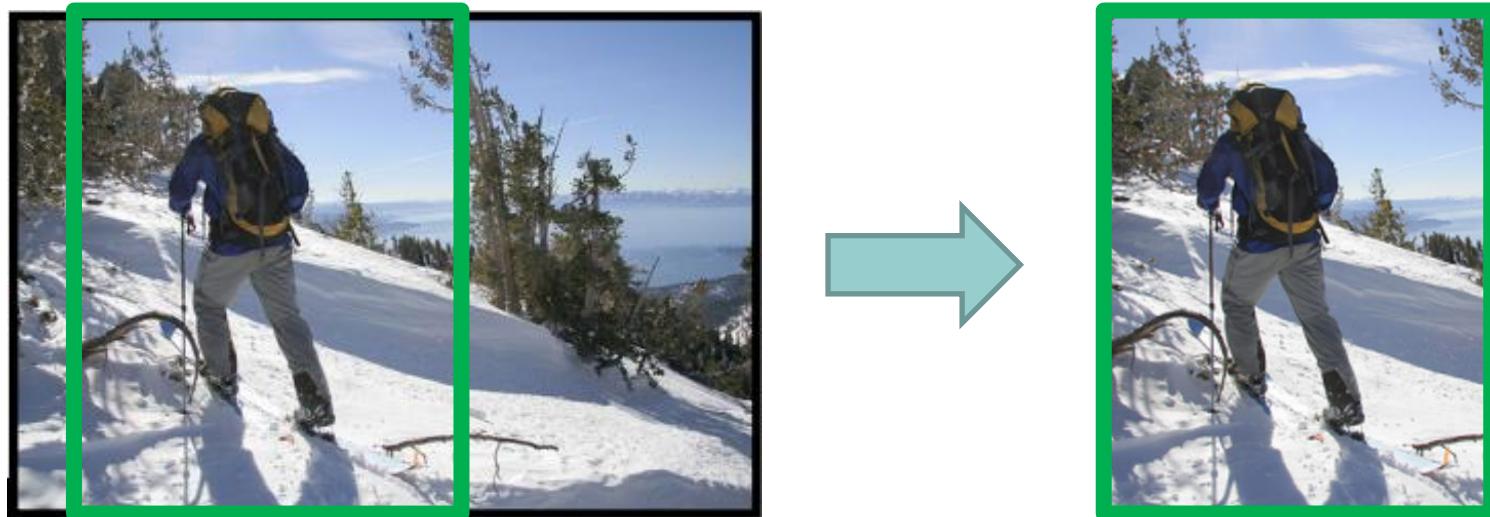


- But how about image?



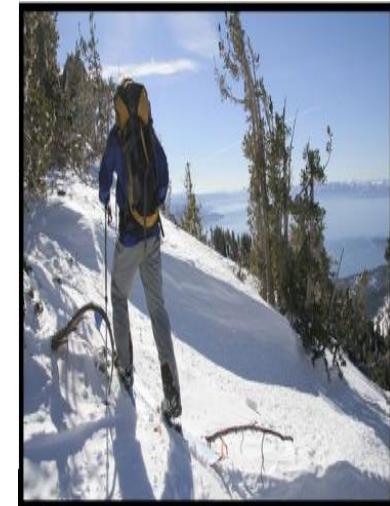
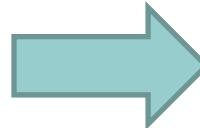
Background

- Simple Media Retargeting Operators
- Crop



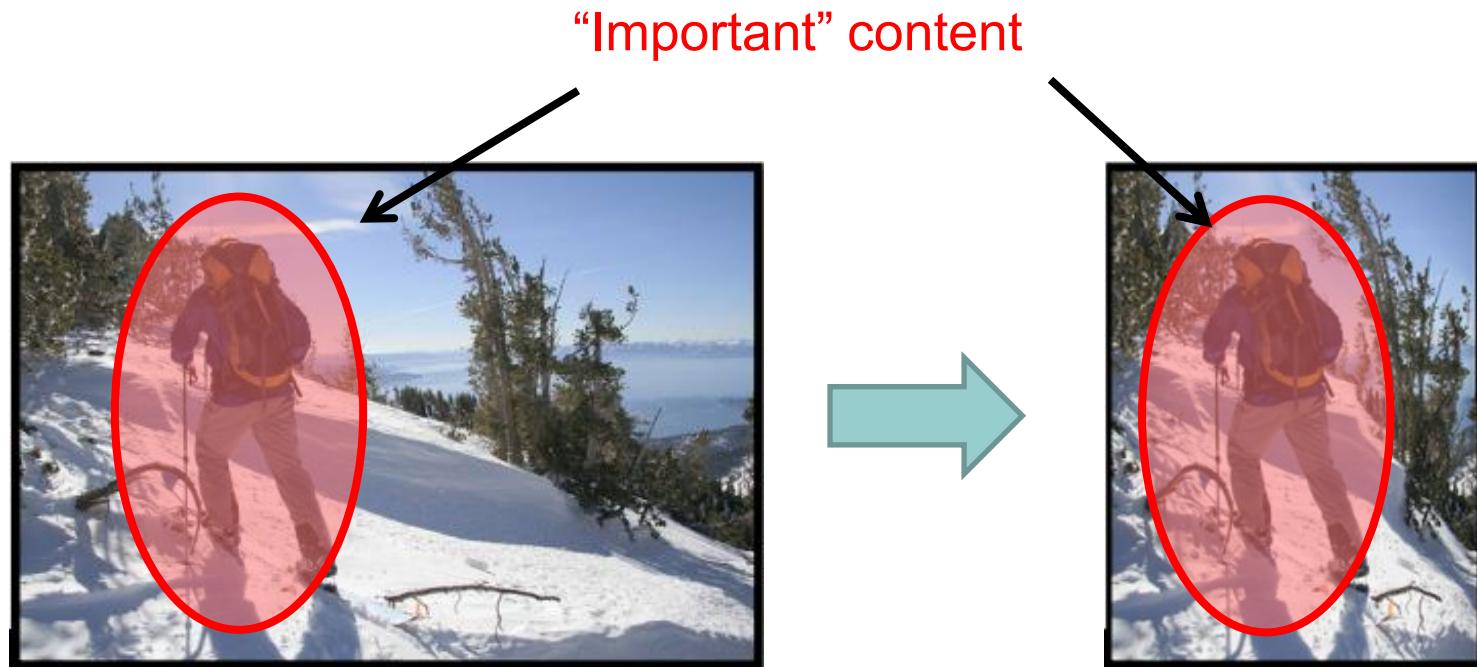
Background

- Simple Media Retargeting Operators
- Scale



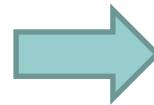
Background

- Content-aware Retargeting Operators



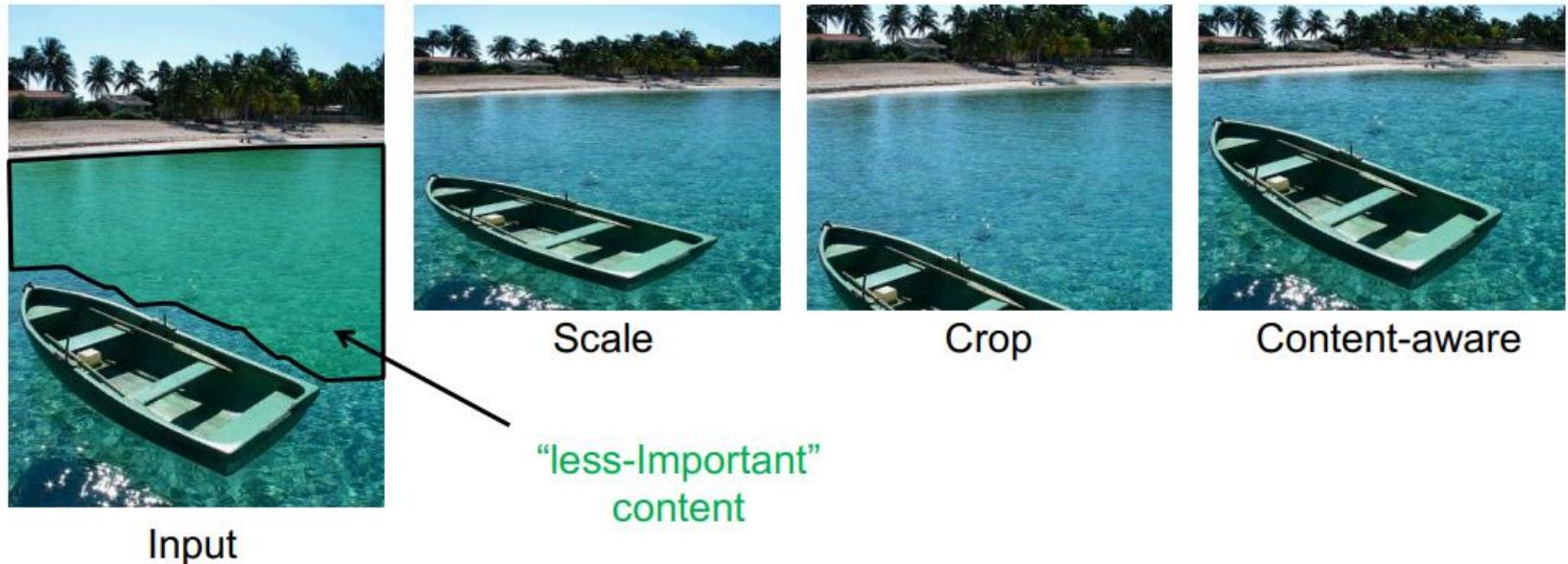
Background

- Content-aware Retargeting Operators



Background

- Content-aware Retargeting Operators



What we will learn today?

- Background
- What is image retargeting
- Seam carving
- Dynamic programming
- Applications

What is image retargeting

- Image Retargeting
- Problem statement:
 - Input Image I ($n * m$), and new size ($n' * m'$)
 - Output Image I' of size ($n' * m'$) which will be “good representative” of the original image I
- “Good representative”?
- To date, no agreed definition, or measure, as to what a good representative is in this context!

What is image 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?

What is image retargeting

- Importance (Saliency) Measures

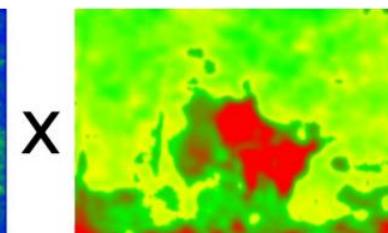
- gradient map $W = \left(\left(\frac{\partial}{\partial x} I \right)^2 + \left(\frac{\partial}{\partial y} I \right)^2 \right)^{1/2}$
- saliency map A function $S: p \rightarrow [0,1]$
- significance map



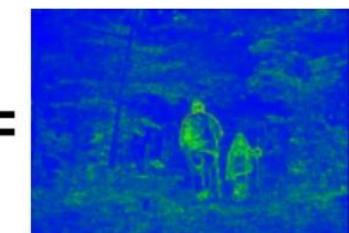
original image



Gradient Map



Saliency Map

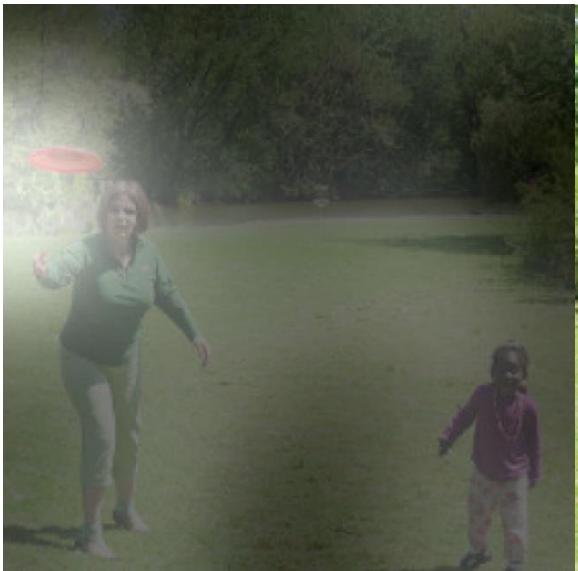


Significance Map

Wang et al. 2008

What is image retargeting

- Importance (Saliency) Measures



What is image retargeting

- Importance (Saliency) Measures
 - More sophisticated: attention models, eye tracking (gazing studies), face detectors, ...

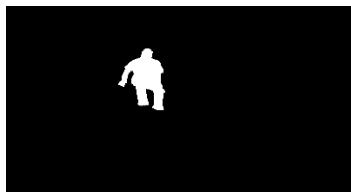


Judd et al. ICCV09 *Learning to predict where people look*

Image Saliency Definition



Salient Object



Eye Fixation

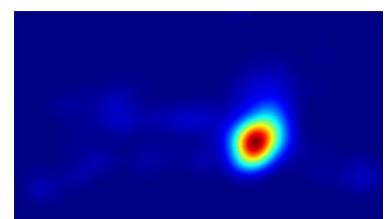
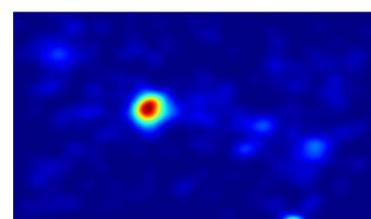
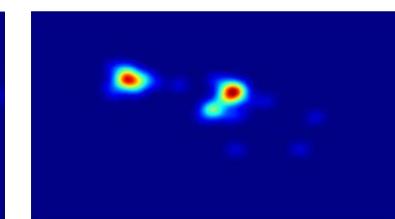
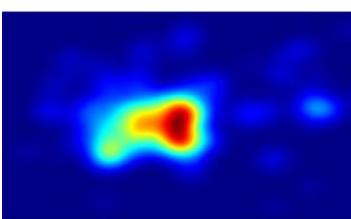
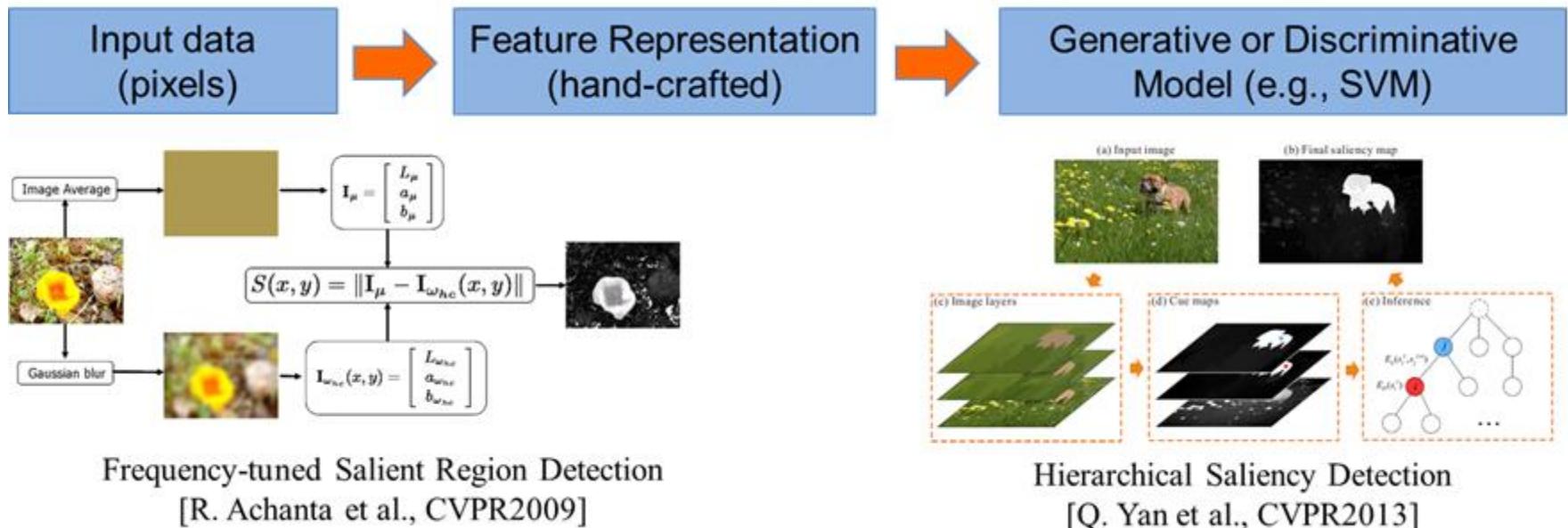


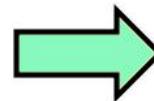
Image Saliency Calculation



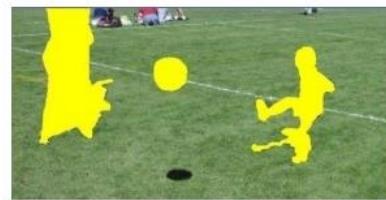
What is image retargeting

- General Retargeting Framework

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



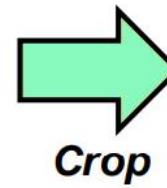
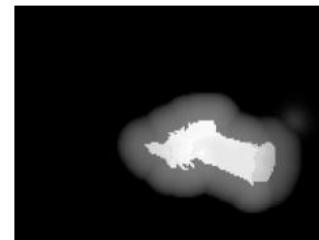
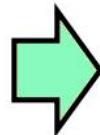
2. Use some operator(s) to
change the image I



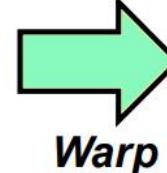
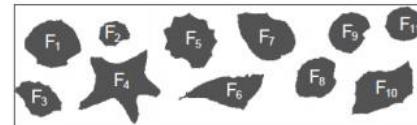
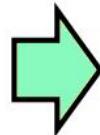
Recompose



Setlur et al.
[2005]



Santella et
al. [2005]



Gal et al.
[2006]

What we will learn today?

- Background
- What is image retargeting
- Seam carving (接缝裁剪算法)
- Dynamic programming
- Applications

Seam carving

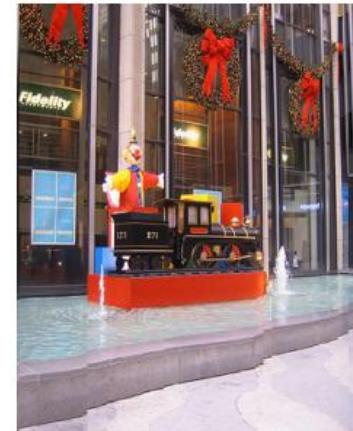
- Assume $m \times n \rightarrow m \times n'$, $n' < n$
- Basic Idea: remove unimportant pixels from the image
 - Unimportant = pixels with less “energy”
 - $E_1(I) = \left| \frac{\partial}{\partial x} I \right| + \left| \frac{\partial}{\partial y} I \right|$.
- Intuition for gradient-based energy:
 - Preserve strong contours
 - Human vision more sensitive to edges – so try remove content from smoother areas
 - Simple enough for producing some nice results

Seam carving

- Pixel Removal



Optimal

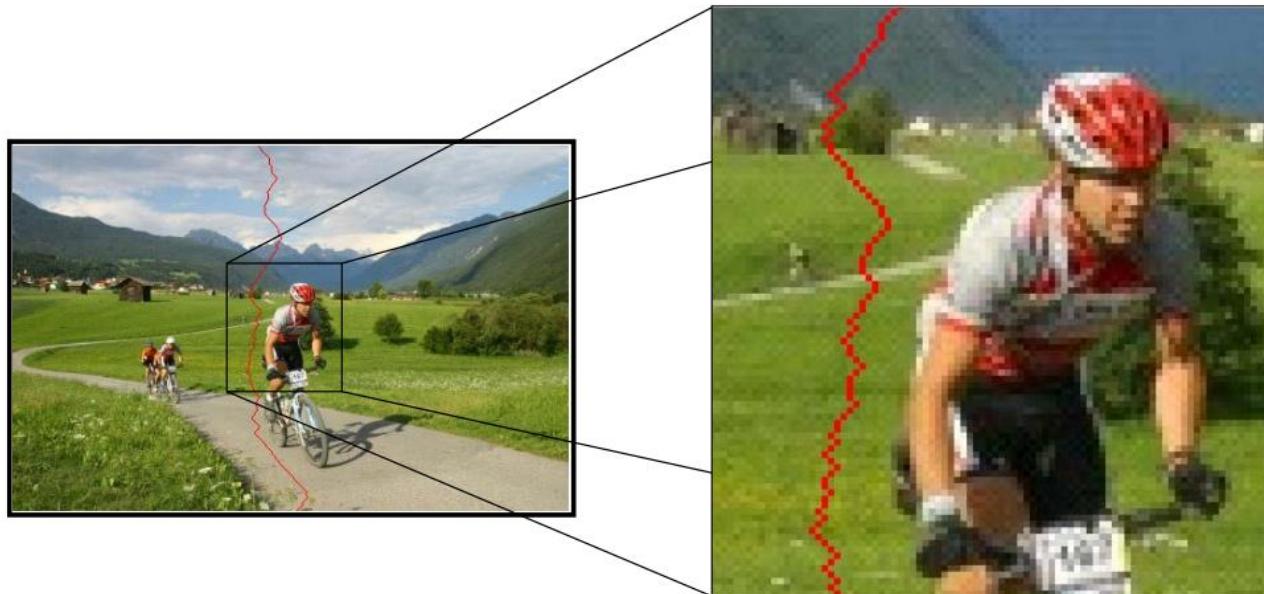
Least-energy pixels
(per row)

Least-energy columns

Seam carving

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

$$\mathbf{s}^x = \{s_i^x\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n, \text{ s.t. } \forall i, |x(i) - x(i-1)| \leq 1$$

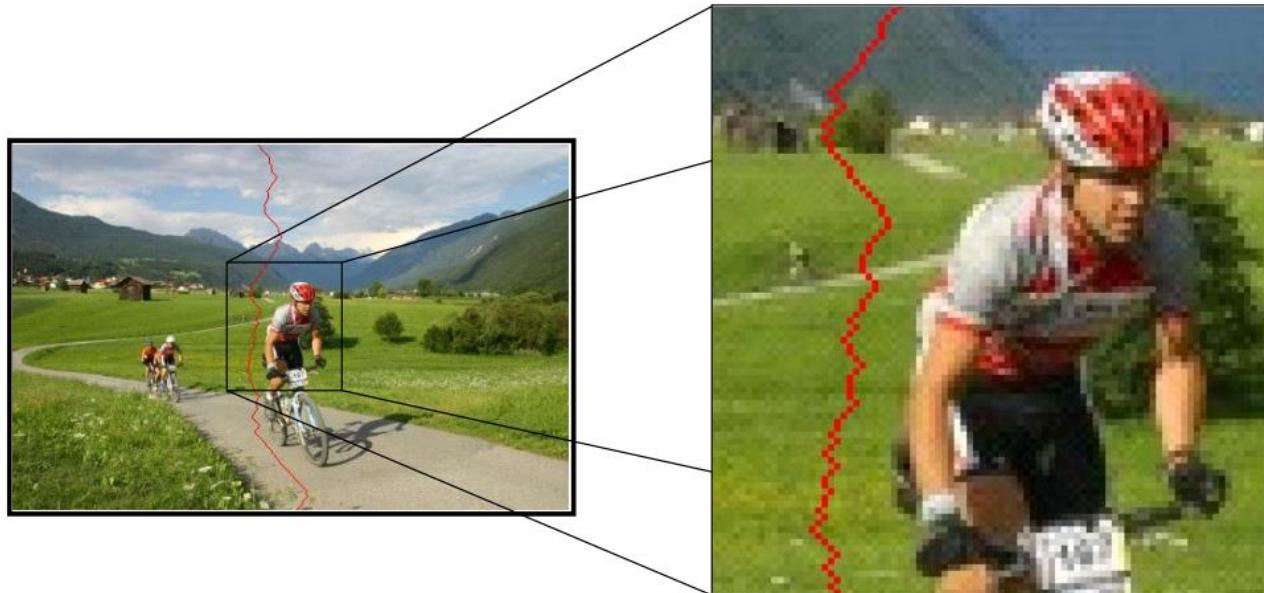


Seam carving

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

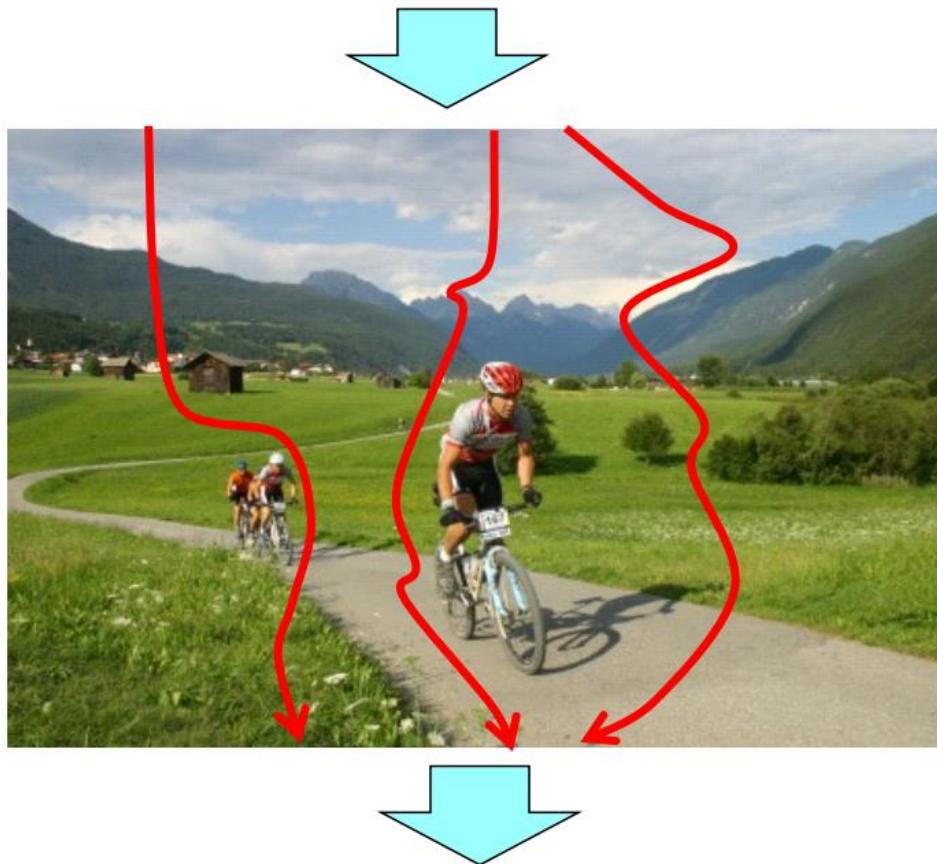
$$\mathbf{s}^x = \{s_i^x\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n, \text{ s.t. } \forall i, |x(i) - x(i-1)| \leq 1$$

$$\mathbf{s}^y = \{s_j^y\}_{j=1}^m = \{(j, y(j))\}_{j=1}^m, \text{ s.t. } \forall j, |y(j) - y(j-1)| \leq 1$$



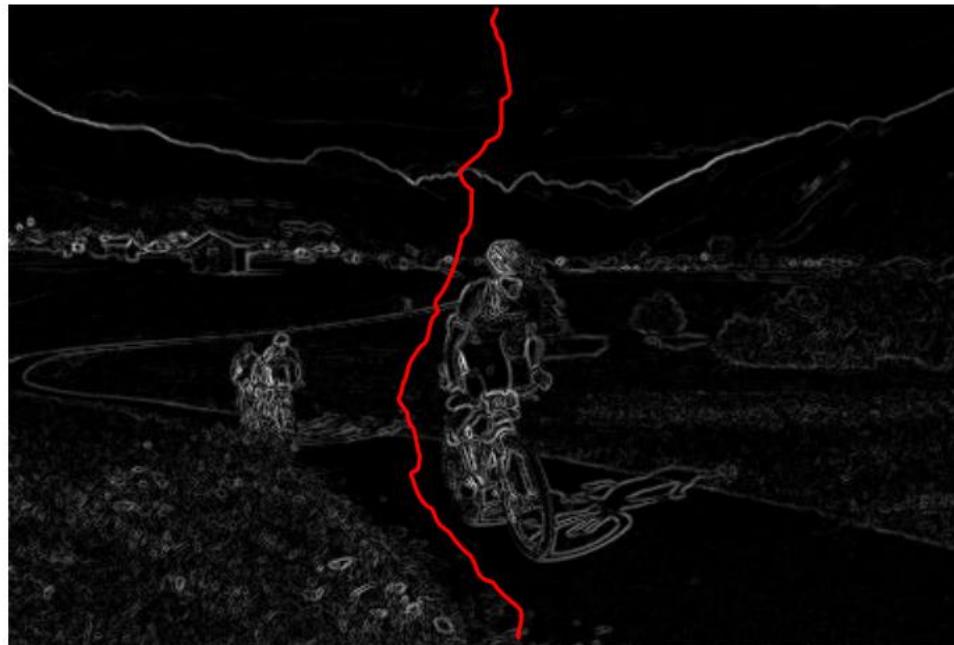
Seam carving

- Finding the Seam?



Seam carving

- The Optimal Seam



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

Seam carving

- The Optimal Seam
- The recursion relation

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

- Can be solved efficiently using dynamic programming in $O(s \cdot n \cdot m)$
($s=3$ in the original algorithm)

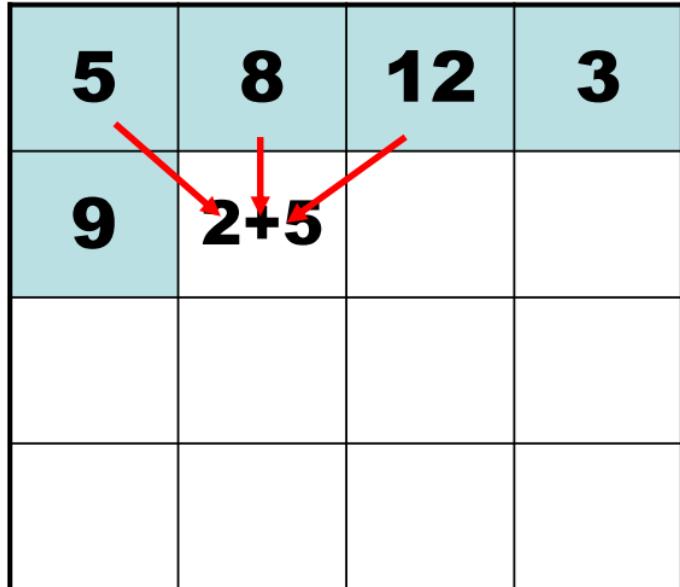
What we will learn today?

- Background
- What is image retargeting
- Seam carving
- Dynamic programming
- Applications

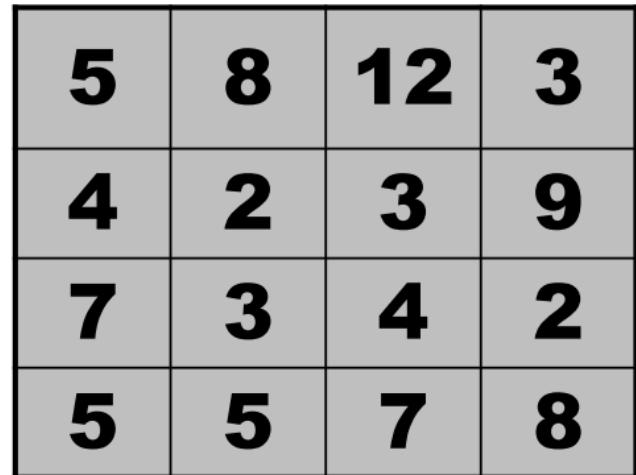
Dynamic programming

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

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



5	8	12	3
9	2+5		



5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - $E(I,j)$

Dynamic programming

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

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

5	8	12	3
9	2+5		

5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - $E(i,j)$

Dynamic programming

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

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

5	8	12	3
9	7	12	3

Diagram illustrating the computation of the minimal cost of a seam. Red arrows point from the value 3 in the second row to the value 3+3 in the third row, indicating the addition of the energy term $E(i, j)$ to the previous minimum value $M(i-1, j)$.

5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - $E(i, j)$

Dynamic programming

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

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

5	8	12	3
9	7	6	12
14	9	10	8
14	14	15	8+8

5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - $E(i,j)$

Dynamic programming

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

5	8	12	3
9	7	6	12
14	9	10	8
14	14	15	16

↑

Dynamic programming

- Searching for Minimum
 - Backtracking the Seam

5	8	12	3
9	7	6	12
14	9	10	8
14	14	15	16

Dynamic programming

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

5	8	12	3
9	7	6	12
14	9	10	8
14	14	15	16

A 4x4 grid of numbers. The bottom-left cell (14) is highlighted in red. An arrow points from the bottom-left cell to the cell above it (9), which is also highlighted in red. The other cells contain the following values: Row 1: 5, 8, 12, 3; Row 2: 9, 7, 6, 12; Row 3: 14, 9, 10, 8; Row 4: 14, 14, 15, 16.

Dynamic programming

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

5	8	12	3
9	7	6	12
14	9	10	8
14	14	15	16

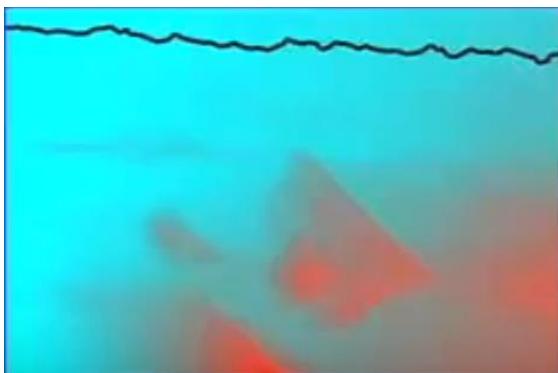
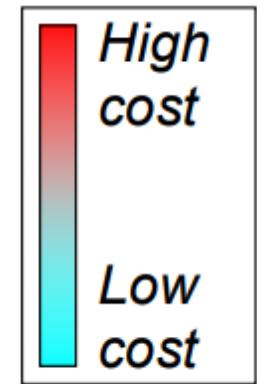
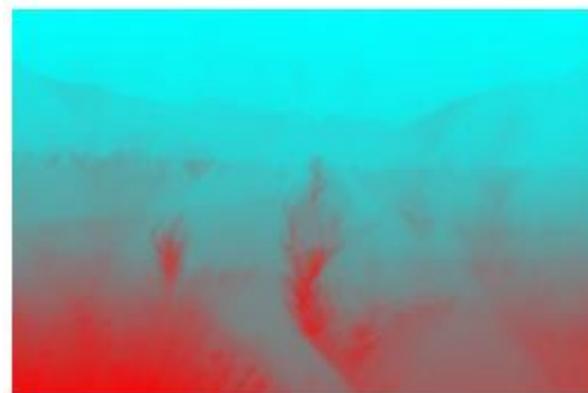
The diagram shows a 4x4 grid of numbers. The numbers are arranged as follows:

5	8	12	3
9	7	6	12
14	9	10	8
14	14	15	16

Cells (3,1), (3,2), (3,3), and (3,4) are highlighted in red. Arrows point from the bottom-left cell (14) to the cell above it (9), and from the cell above (9) to the cell to its right (10).

Dynamic programming

- H & V Cost Maps



Horizontal Seam

Vertical Seam

Dynamic programming

- Seam Carving (GIF)

Sandipan Dev (UMBC)

Removing V-Seam, image shape = (413x627), size = 5.93 MB



Dynamic programming

- The Seam-Carving Algorithm

```
SEAM -CARVING(im, n') // size(im) = m x n
```

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'

Dynamic programming

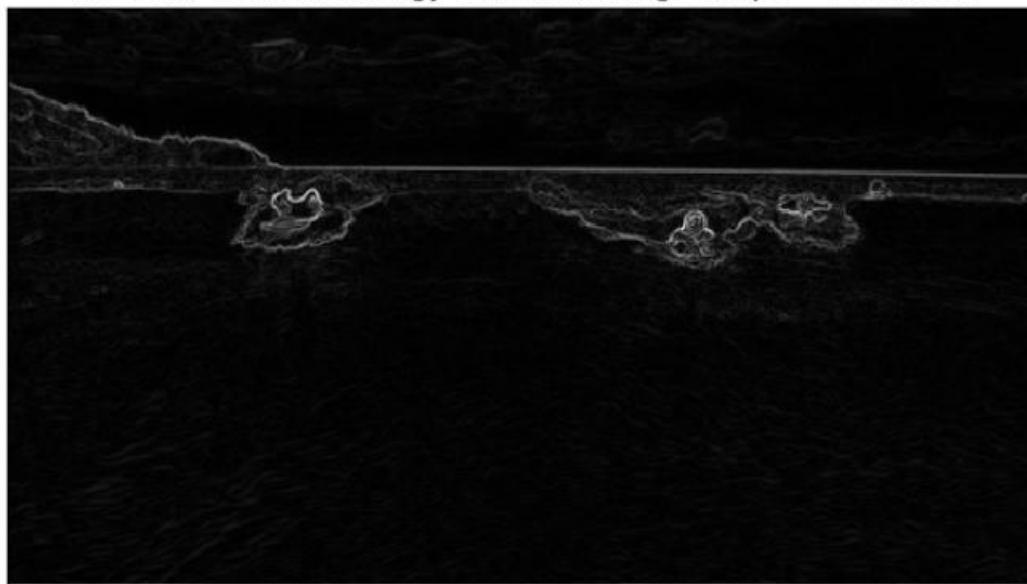
- Changing Aspect Ratio



Dynamic programming

- Changing Aspect Ratio

The Dual-Gradient Energy Function, image shape = (284x507)



Dynamic programming

- Changing Aspect Ratio (GIF)

Sandipan Dey (UMBC)

Removing Vertical Seam, image shape = (285x506), size = 3.31 MB

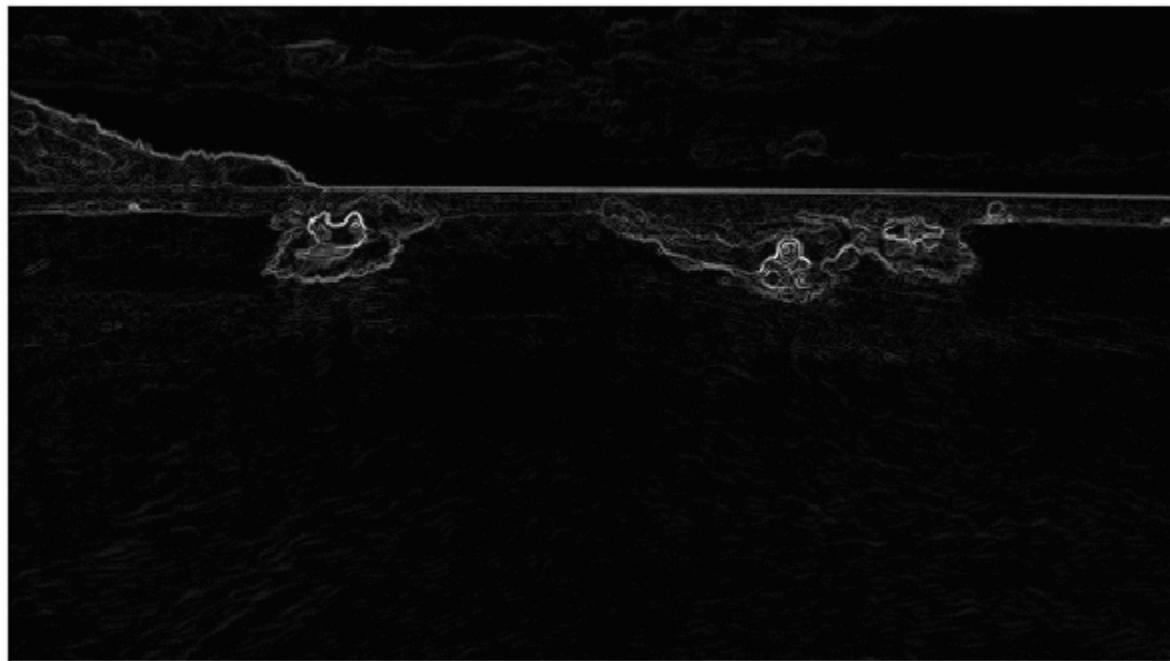


Dynamic programming

- Changing Aspect Ratio (GIF)

Sandipan Dey (UMBC)

The Dual-Gradient Energy Function, image shape = (285x507)



Dynamic programming

- Another example



Dynamic programming

- Changing Aspect Ratio



Original



Seam Carving



Scaling

Dynamic programming

- Changing Aspect Ratio



Cropping



Seams



Scaling

Dynamic programming

- Changing Aspect Ratio



Original



Retarget



Scaling

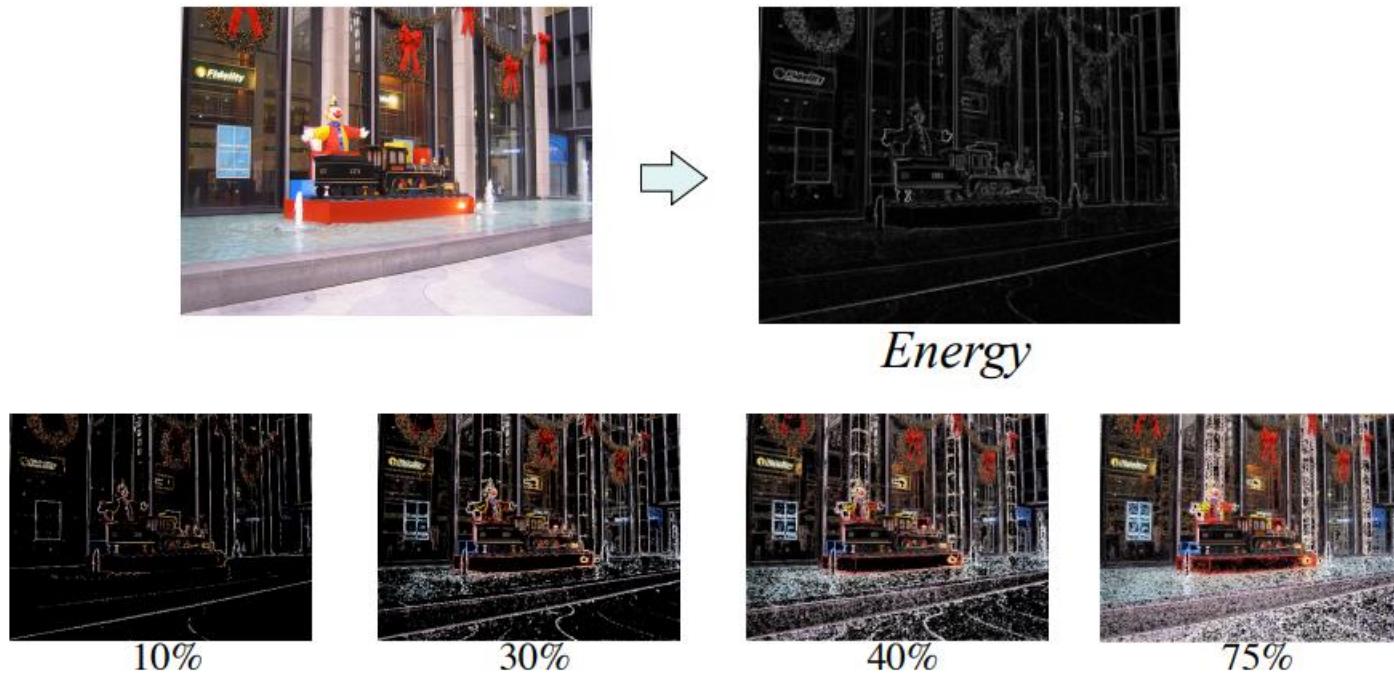
Dynamic programming

- Content Enhancement



Dynamic programming

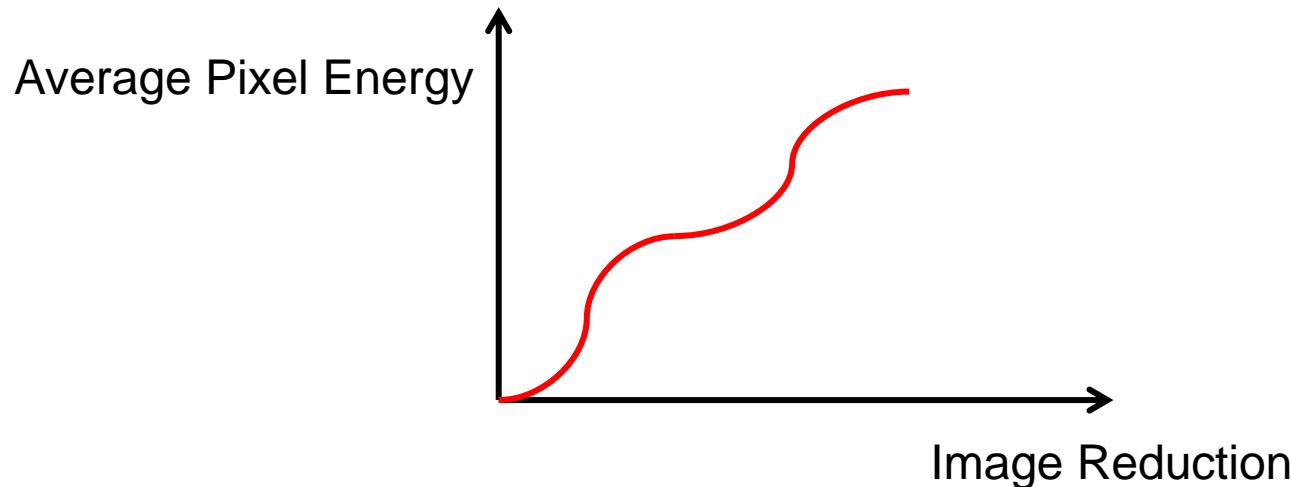
- Q: What happens to the overall energy in the image during seam carving?



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

Dynamic programming

- Preserved Energy
- While resizing: remove **as many** low energy pixels and **as few** high energy pixels!
- If we measure the average energy of pixels in the image after applying a resizing operator...
- ...the average should increase!



Dynamic programming

- Preserved Energy



Average
Pixel
Energy

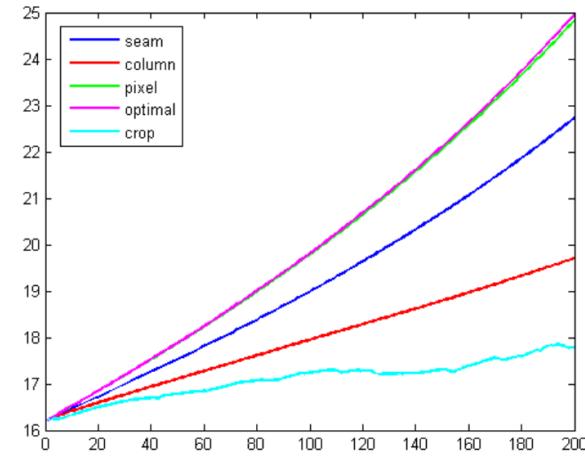
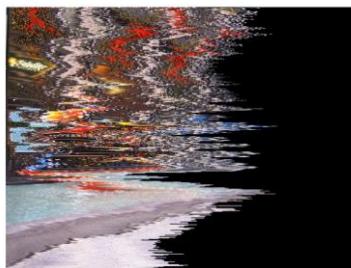


Image Reduction



optimal



pixel



crop



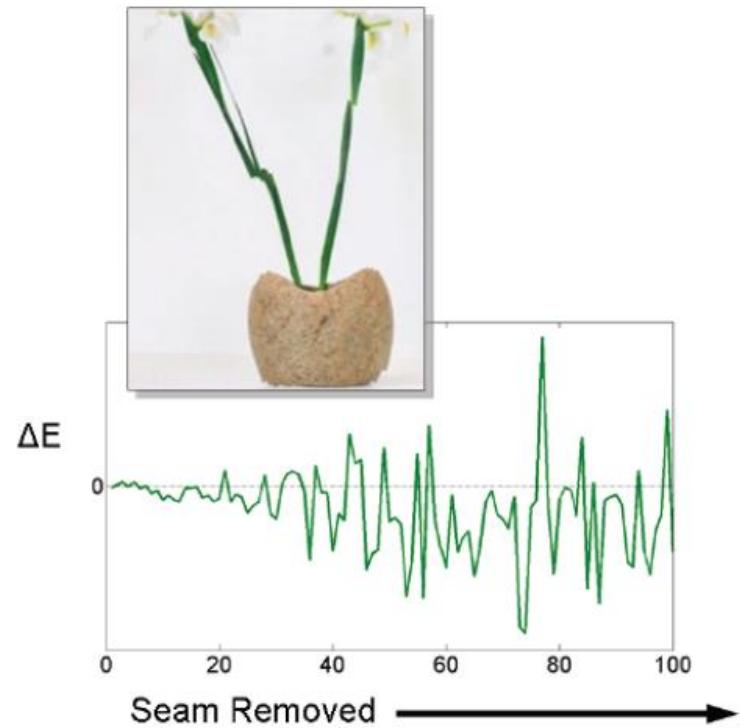
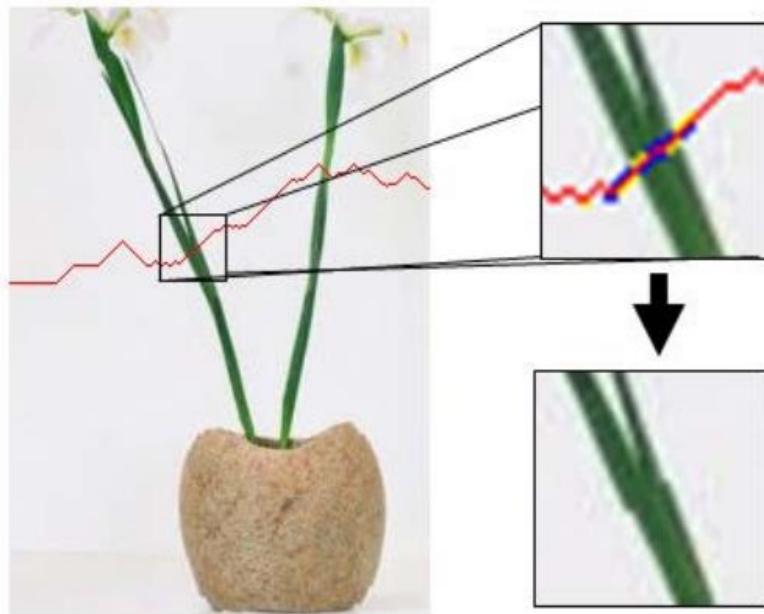
column



seam

Dynamic programming

- Inserted Energy
 - Limitation



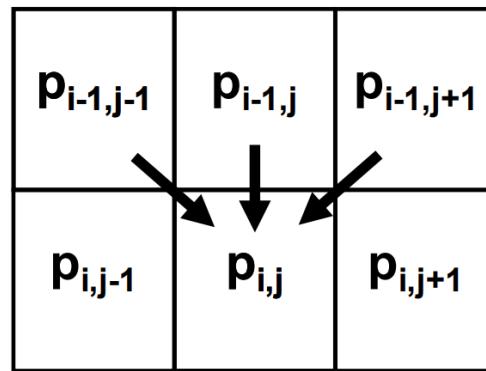
Dynamic programming

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

Dynamic programming

- Old “Backward” Energy

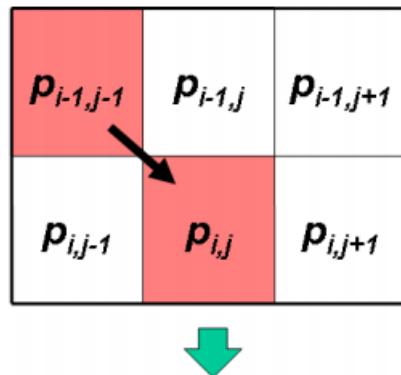
$$M(i, j) = E(i, j) + \min \begin{cases} M(i - 1, j - 1) \\ M(i - 1, j) \\ M(i - 1, j + 1) \end{cases}$$



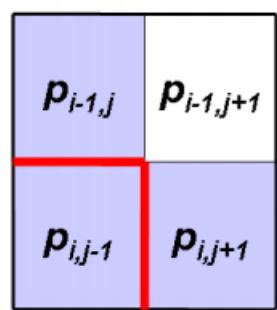
- Three possibilities when removing pixel $P_{i,j}$

Dynamic programming

- Left Seam

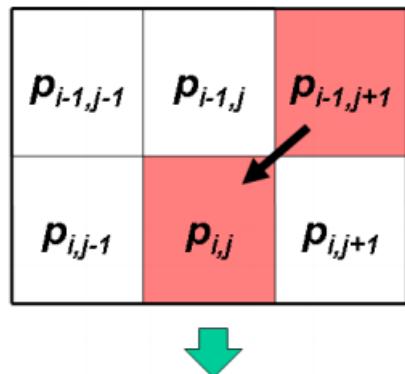


$$C_L(i, j) = |I(i, j + 1) - I(i, j - 1)| + |I(i - 1, j) - I(i, j - 1)|$$

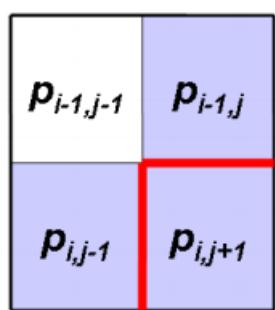


Dynamic programming

- Right Seam

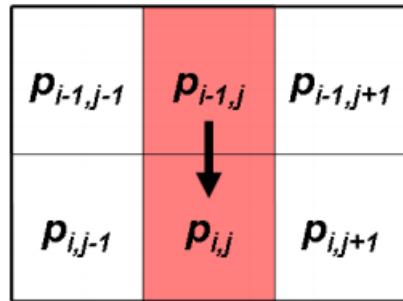


$$C_R(i, j) = |I(i, j + 1) - I(i, j - 1)| + |I(i - 1, j) - I(i, j + 1)|$$

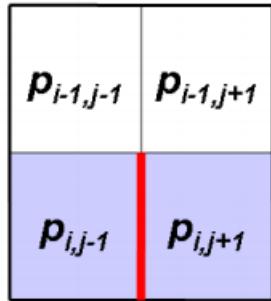


Dynamic programming

- Vertical Seam



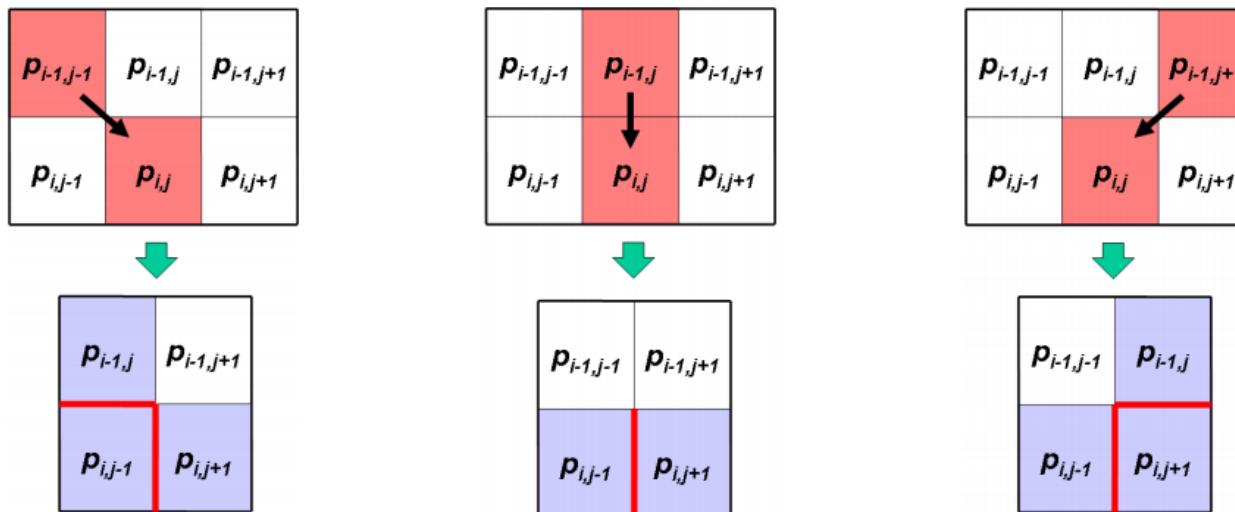
$$C_V(i, j) = |I(i, j + 1) - I(i, j - 1)|$$



Dynamic programming

- New “Forward” Looking Energy

$$M(i, j) = \min \begin{cases} M(i - 1, j - 1) + |I(i, j + 1) - I(i, j - 1)| + |I(i - 1, j) - I(i, j - 1)| \\ M(i - 1, j) + |I(i, j + 1) - I(i, j - 1)| \\ M(i - 1, j + 1) + |I(i, j + 1) - I(i, j - 1)| + |I(i - 1, j) - I(i, j + 1)| \end{cases}$$



Dynamic programming

- Backward vs. Forward
 - removing the seam of least energy

$$M(i, j) = E(i, j) + \min \begin{cases} M(i - 1, j - 1) \\ M(i - 1, j) \\ M(i - 1, j + 1) \end{cases}$$

- remove the seam that inserts the least energy to the image

$$M(i, j) = \min \begin{cases} M(i - 1, j - 1) + |I(i, j + 1) - I(i, j - 1)| + |I(i - 1, j) - I(i, j - 1)| \\ M(i - 1, j) + |I(i, j + 1) - I(i, j - 1)| \\ M(i - 1, j + 1) + |I(i, j + 1) - I(i, j - 1)| + |I(i - 1, j) - I(i, j + 1)| \end{cases}$$

Applications

- Results



Input



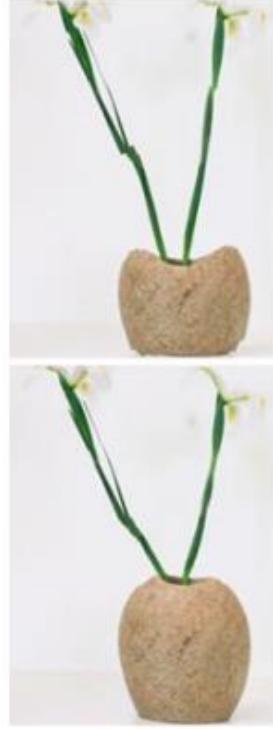
Backward



Forward



Input



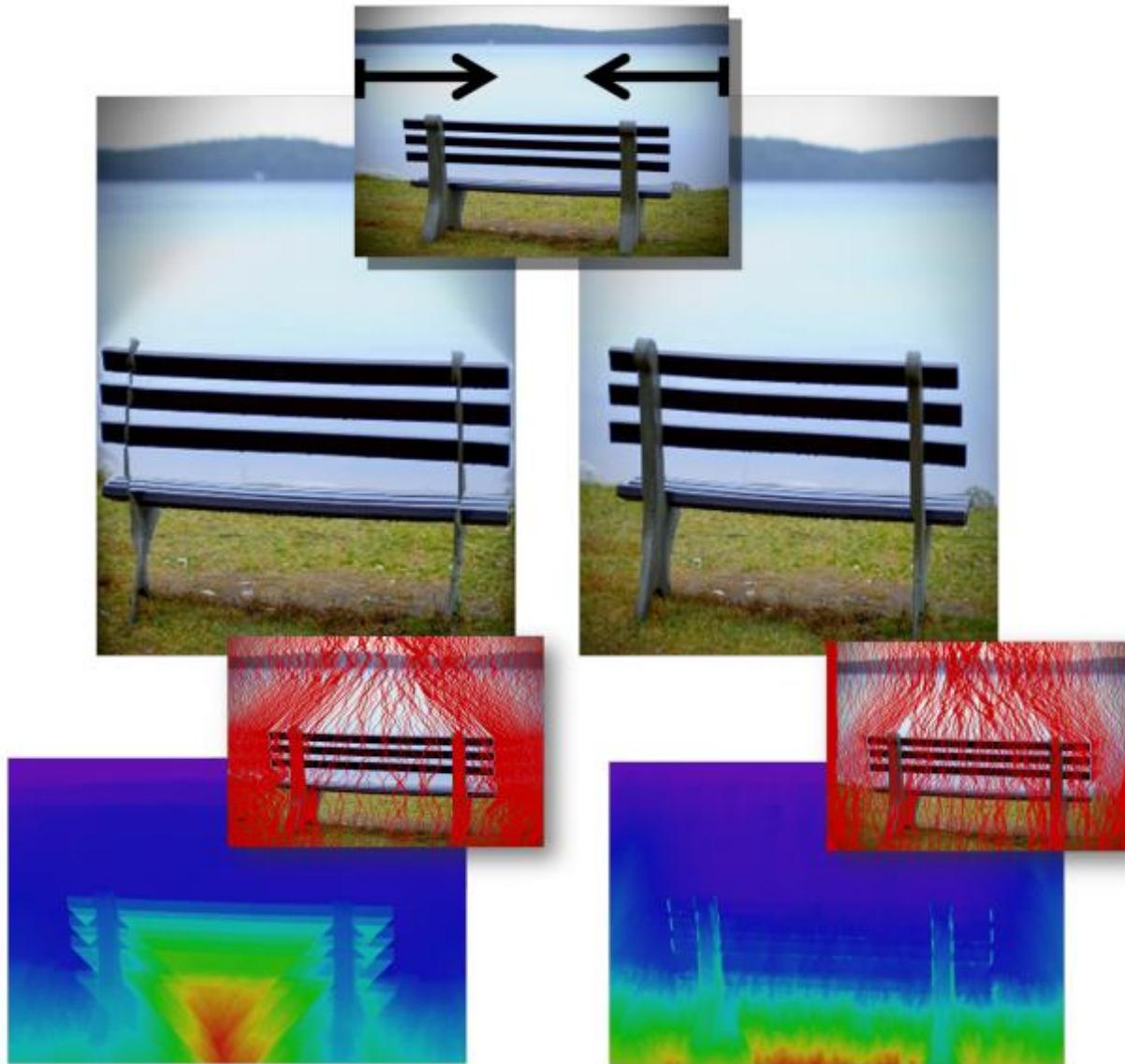
Backward



Forward

Dynamic programming

- Results

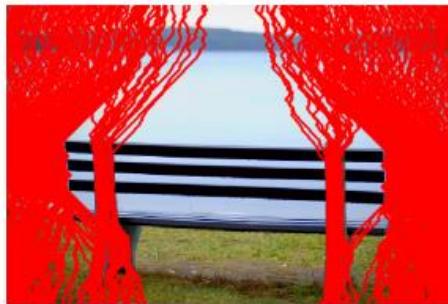


Dynamic programming

- Backward vs. Forward



(a) Original image.



(b) Backward energy seams.



(c) BE Resizing.



(d) Forward energy seams.



(e) FE Resizing.

Applications

- Backward vs. Forward



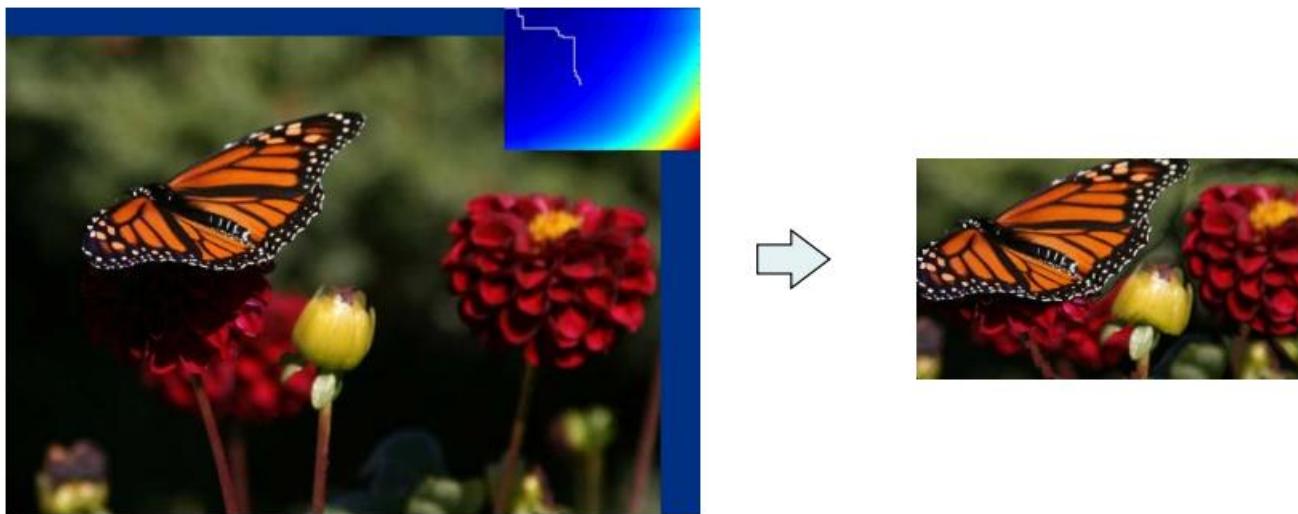
Backward

Forward

Dynamic programming

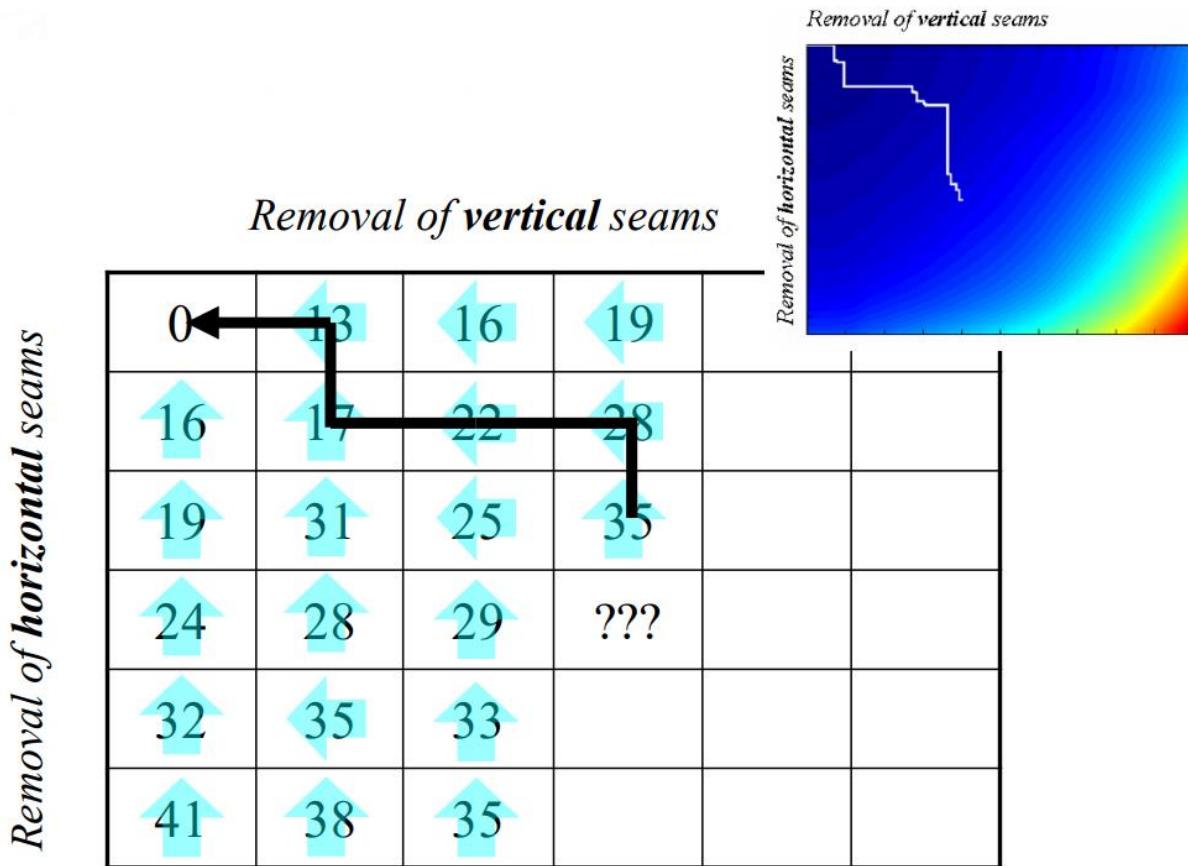
- Extension to Both Dimensions

- $m \times n \rightarrow m' \times n'$
- Remove horizontal seam first?
- Remove vertical seams first?
- Alternate between the two?
- The optimal order can be found! \rightarrow Dynamic Prog (again)



Dynamic programming

- Extension to Both Dimensions

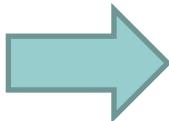


What we will learn today?

- Background
- What is image retargeting
- Seam carving
- Dynamic programming
- Applications

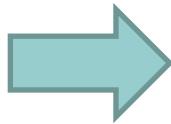
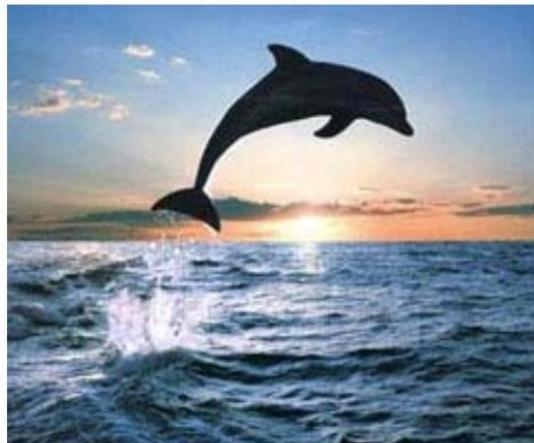
Applications

- Image Expansion (Synthesis)



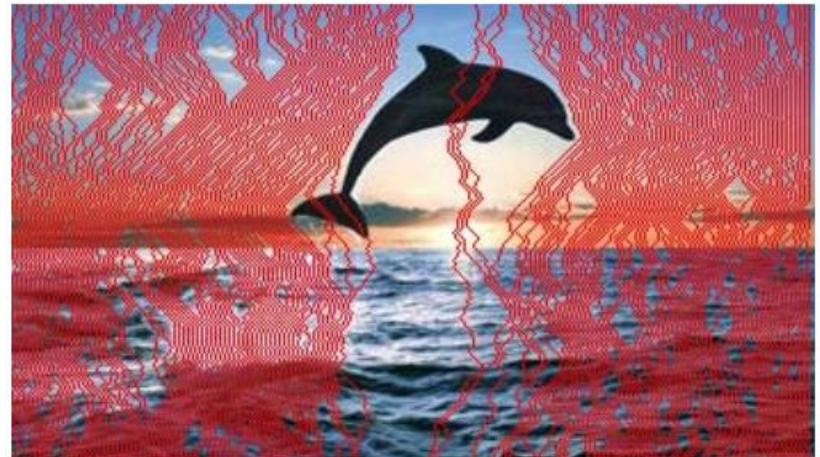
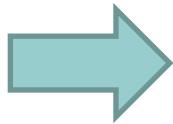
Applications

- Image Expansion



Applications

- Image Expansion (Synthesis)



Applications

- Combined Insert and Remove



Insert & remove seams



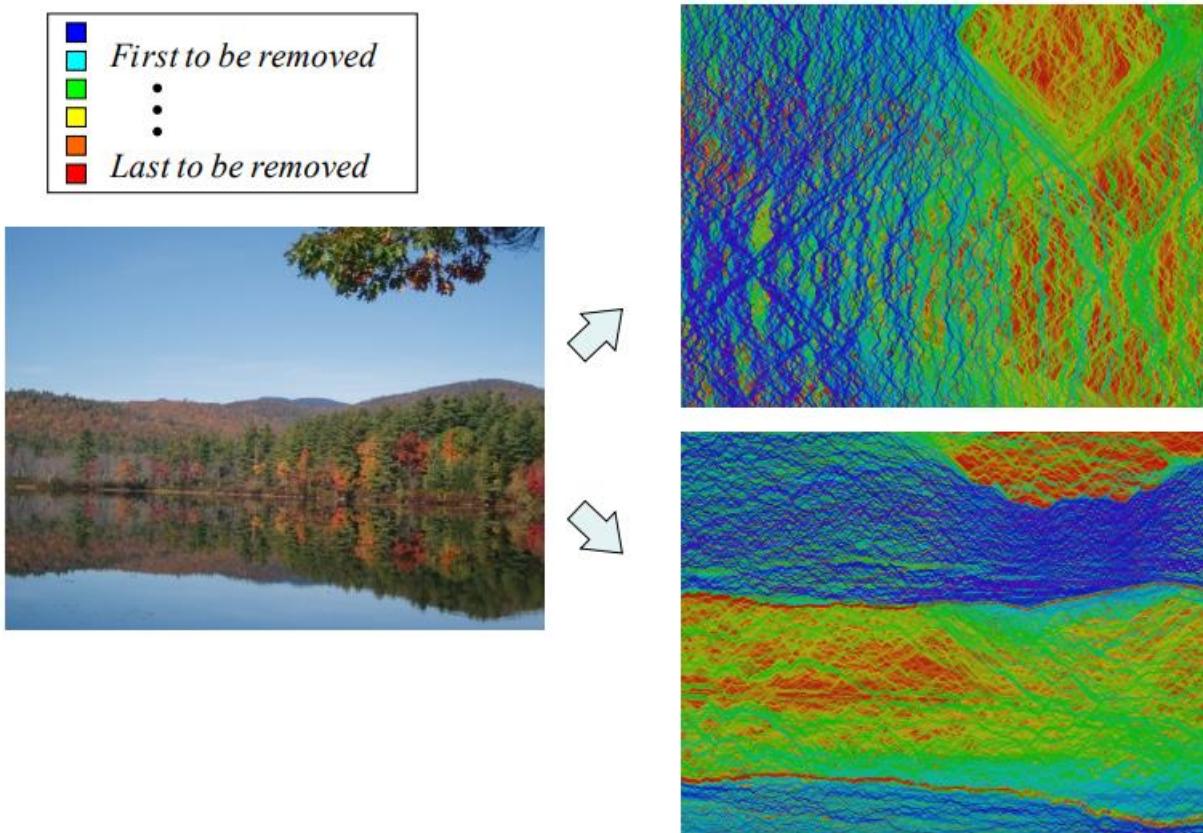
Scaling

Applications

- Multi-Size Image
 - 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

Applications

- Multi-Size Images

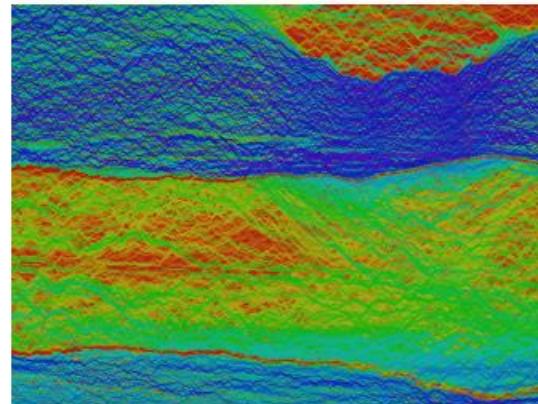


Applications

- Multi-Size Image Representation



+

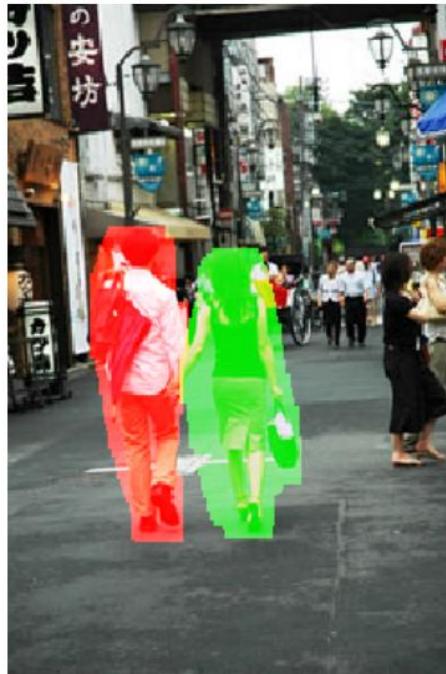


Applications

- Object Removal



(a) Original image.



(b) Highlighted regions.



(c) Resized image with object removed.

Applications

- Auxiliary Energy
 - Recall our seam equation (Backward)

$$M(i, j) = E(i, j) + \min \begin{cases} M(i - 1, j - 1) \\ M(i - 1, j) \\ M(i - 1, j + 1) \end{cases}$$

Applications

- Object Removal



Input

Retargeted

Pigeon Removed

Girl Removed

Applications

- Limitations

Content



Structure



Applications

- With User Constraints



(a) Original image.



(b) Failed seam carved image.



(c) Highlighted original image.



(d) Successful seam carved image.

Applications

- With User Constraints



Applications

- With face detector



Applications

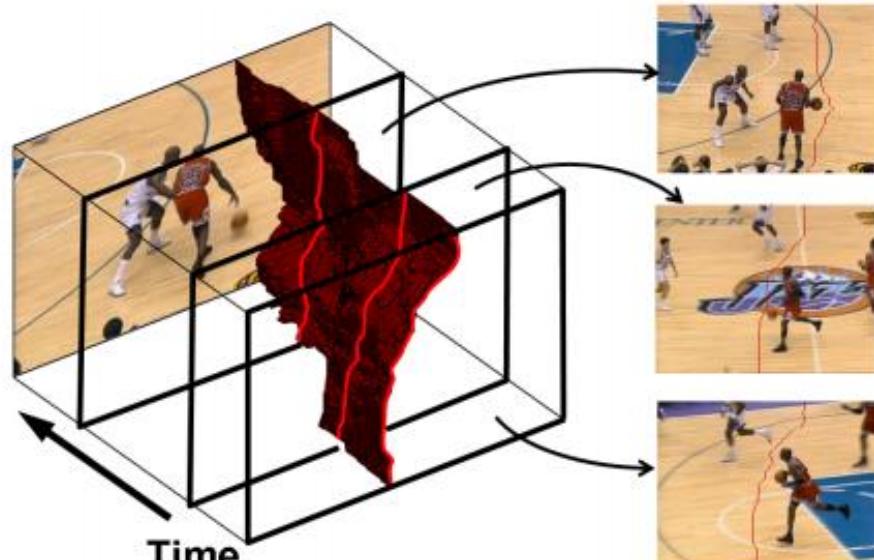
- From Images to Videos
 - In general, video processing is a much (much!) harder problem
 - 1. Cardinality
 - Suppose 1min of video x 30 fps = 1800 frames
 - Say your algorithm processes an image in 1 minute -> 30 hours !!
 - 2. Dimensionality/algorithmic
 - Temporal coherency: human visual system is highly sensitive to motion

Applications

- From 2D to 3D



1D paths in images



2D manifolds in video cubes

Seam Carving Demo



Shai Avidan
Mitsubishi Electric Research Lab
Ariel Shamir
The interdisciplinary Center & MERL

What we have learned today?

- Background
- What is image retargeting
 - $I(n * m) \rightarrow I'(n' * m')$ with “good representation”



- Seam carving
 - remove unimportant pixels from the image

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

- the Optimal Seam
 - $E(\mathbf{I}) = \left| \frac{\partial}{\partial x} \mathbf{I} \right| + \left| \frac{\partial}{\partial y} \mathbf{I} \right| \Rightarrow s^* = \arg \min_s E(s)$

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

What we have learned today?

- Dynamic programming

- algorithm

```
SEAM -CARVING(im, n') // size(im) = m x n
```

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

- preserved energy
- backward vs. forward
- Extension to Both Dimensions

- Applications

- Image Expansion, Multi-Size Image, Object Removal, video retargeting ...