Lecture: Image Resizing

Juan Carlos Niebles and Ranjay Krishna

Stanford Vision and Learning Lab

CS 131 Roadmap

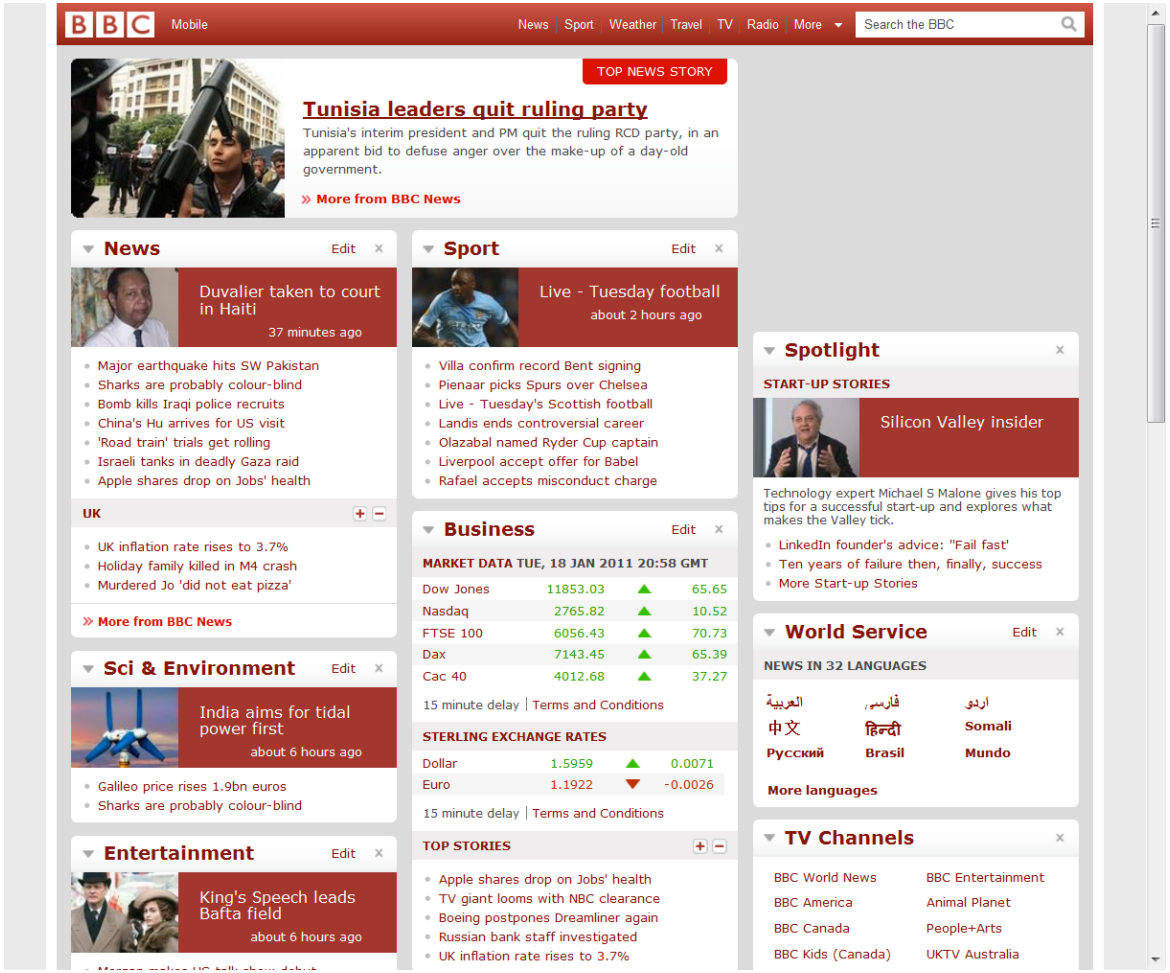
Pixels	Segments	Images	Videos	Web
Convolutions Edges Descriptors	Resizing Segmentation Clustering	Recognition Detection Machine learning	Motion Tracking	Neural networks Convolutional neural networks

Today’s agenda

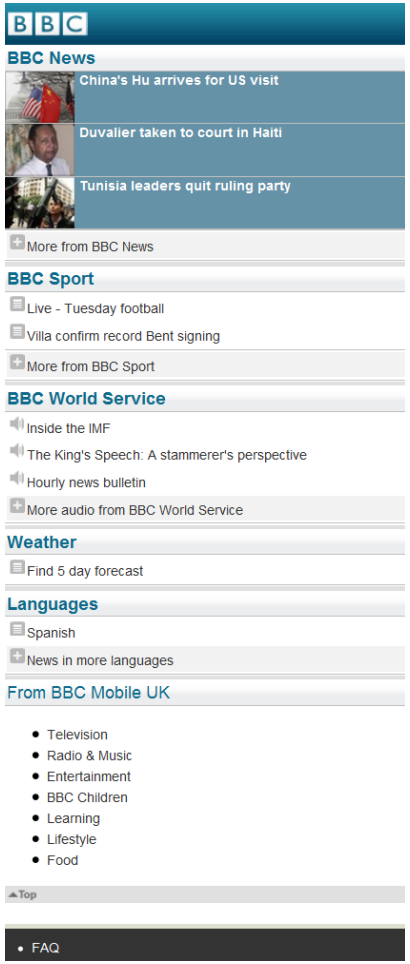
- Image resizing
  - Seam carving
  - Dynamic programming
- Applications

Display Devices

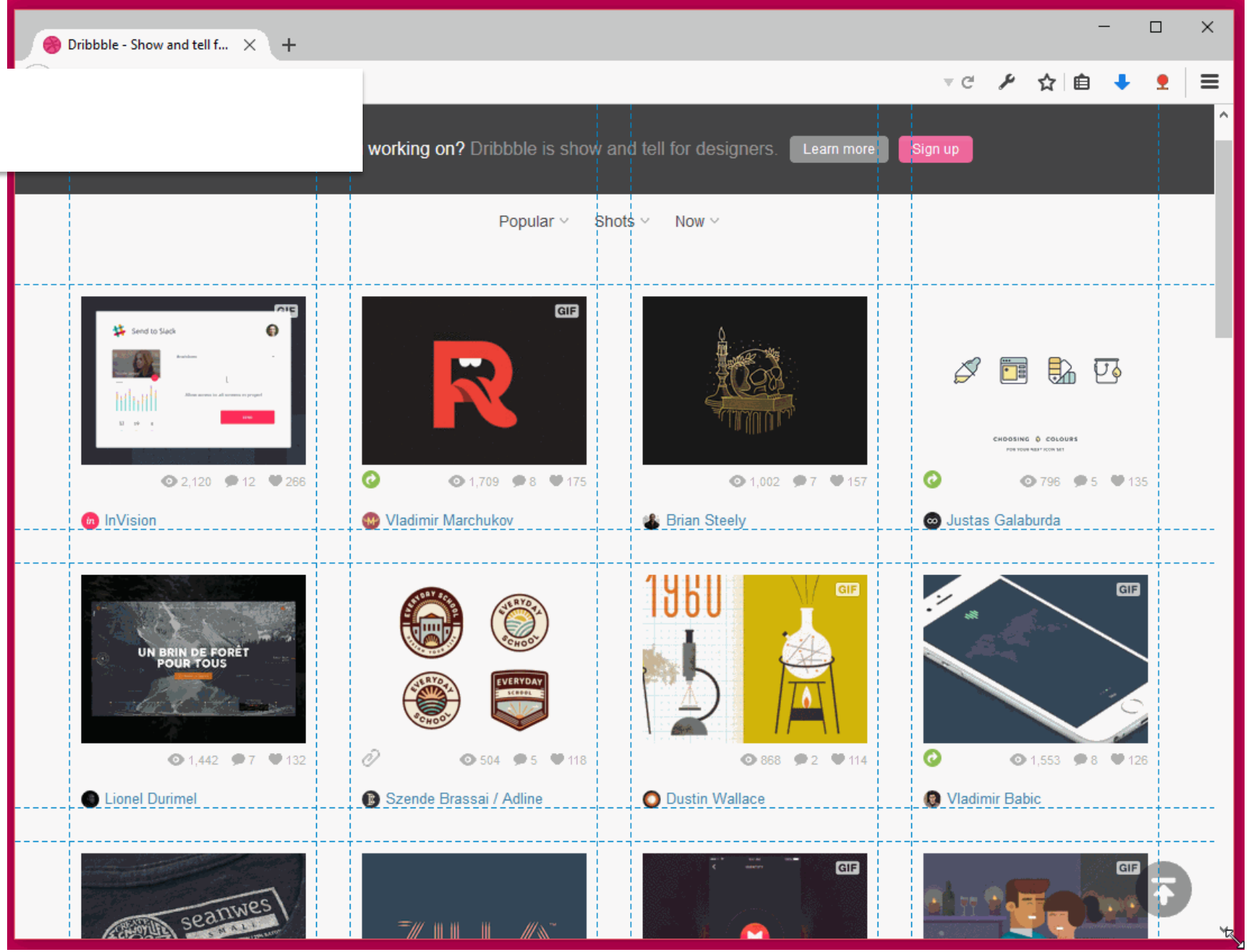
Content Retargeting



PC



iPhone


Dribbble - Show and tell f...
working on? Dribbble is show and tell for designers.
Learn more
Sign up
Popular
Shots
Now
InVision
Vladimir Marchukov
Brian Steely
Justas Galaburda
Lionel Durimel
Szende Brassai / Adline
Dustin Wallace
Vladimir Babic

Simple Media Retargeting Operators

Letterboxing Scaling

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'xm'
  - Output Image I' of size n'xm' 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

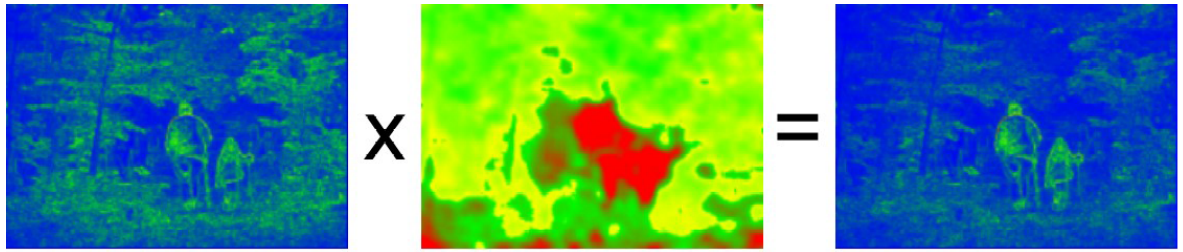
- In large, we would expect retargeting to:

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 → [0,1]



Wang et al. 2008

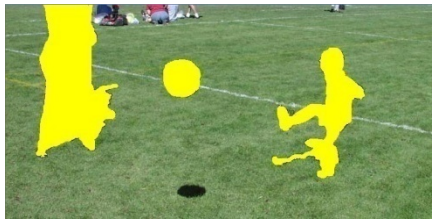
- More sophisticated: attention models, eye tracking (gazing studies), face detectors, …

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

General Retargeting Framework

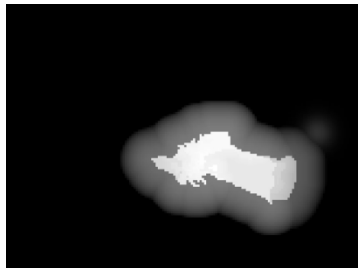
1. Define an energy function **E(I)** (interest, importance, saliency)

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



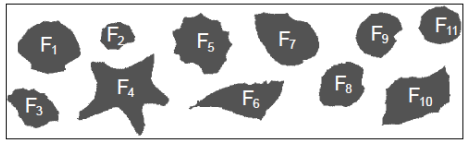
Recompose

Setlur et al. [2005]



Crop

Santella et al. [2005]

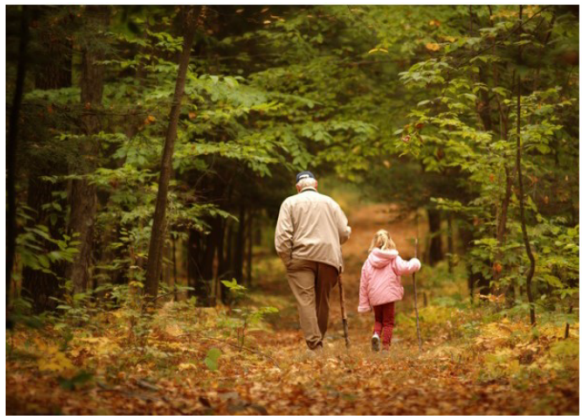


Warp

Gal et al. [2006]

Previous Retargeting Approaches

- Optimal Cropping Window



- For videos: “Pan and scan”

  Still done manually in the movie industry

Liu and Gleicher, **Video Retargeting: Automating Pan and Scan (2006)**

Cropping

Seam Carving

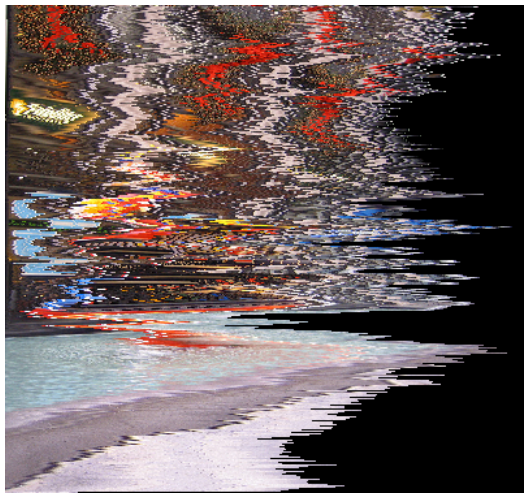
- Assume m x n → m x n', n'<n (summarization)

- Basic Idea: remove unimportant pixels from the image
  - Unimportant = pixels with less "energy"

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

- 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

Pixel Removal



Optimal

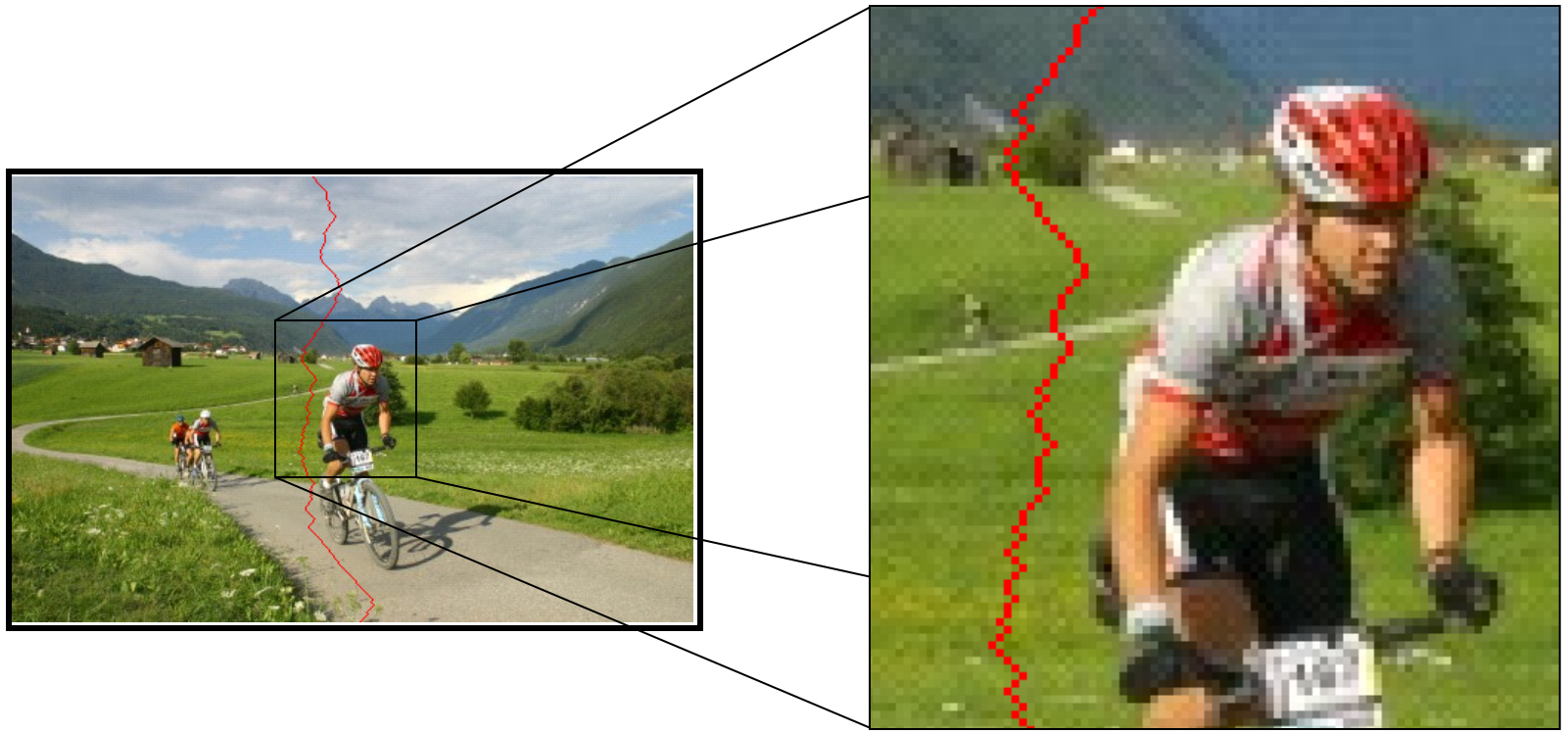
Least-energy pixels (per row)

Least-energy columns

A Seam

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

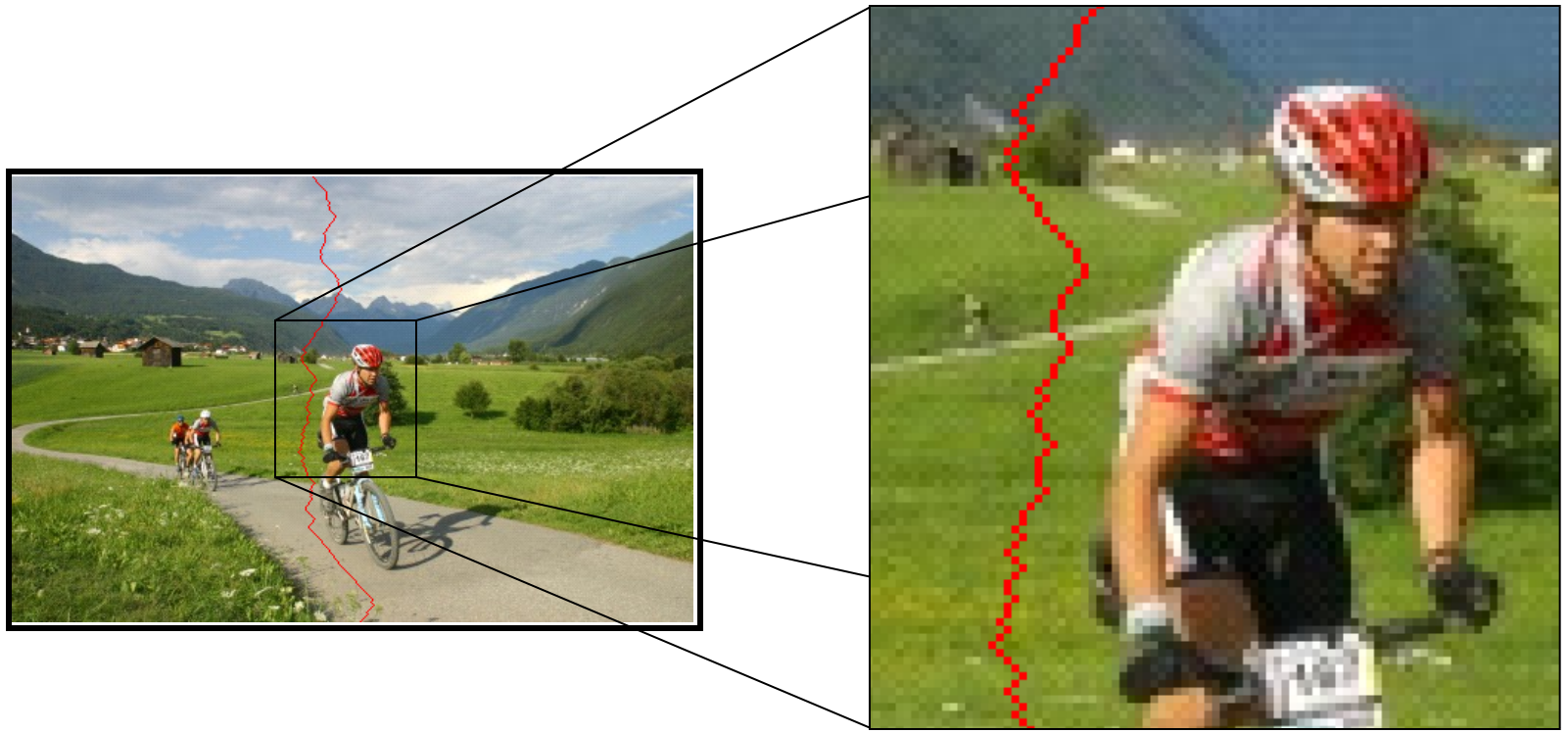
$$\mathbf{s}^{\mathbf{x}} = \{s_i^x\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n, \text{ s.t. } \forall i, |x(i) - x(i-1)| \leq 1$$



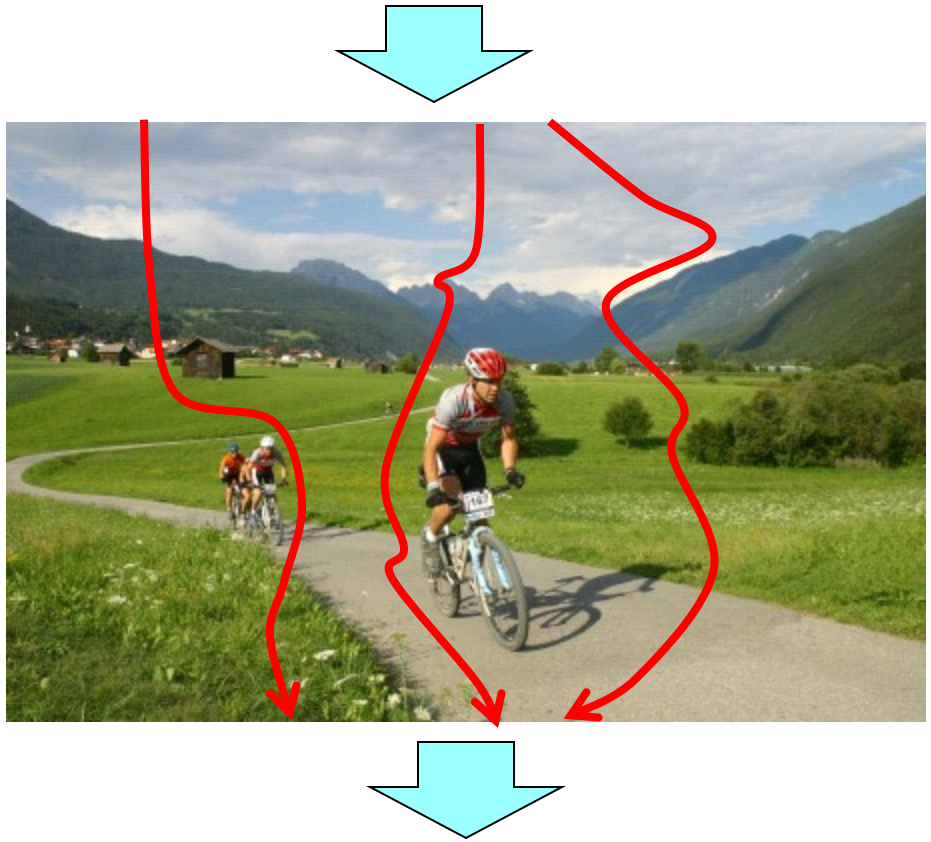
A Seam

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

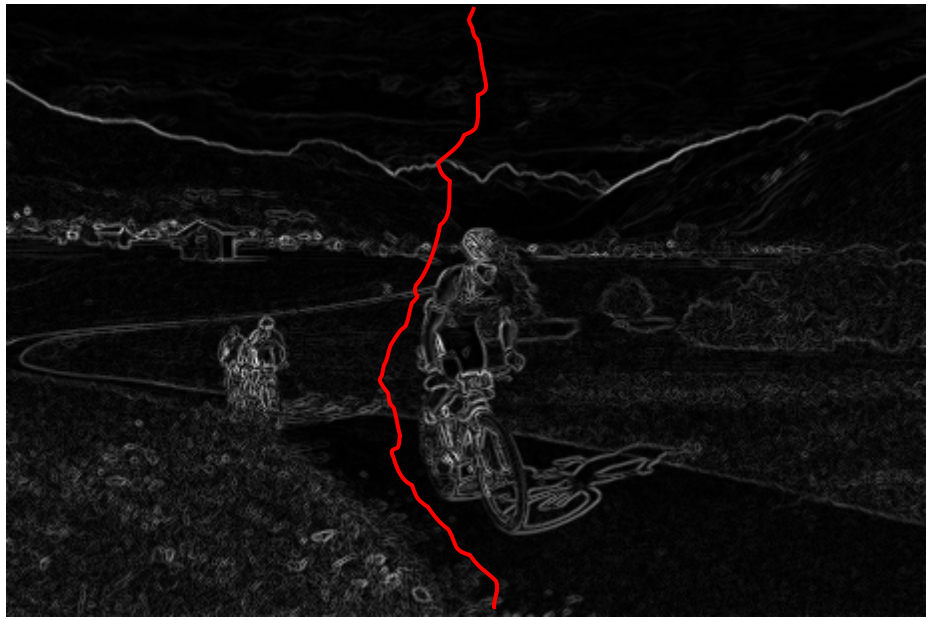
$$\mathbf{s^x} = \{s_i^x\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n, \text{ s.t. } \forall i, |x(i) - x(i-1)| \leq 1$$
$$\mathbf{s^y} = \{s_j^y\}_{j=1}^m = \{(j, y(j))\}_{j=1}^m, \text{ s.t. } \forall j |y(j) - y(j-1)| \leq 1$$



Finding the Seam?



The Optimal Seam



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

The Optimal Seam

- The recursion relation

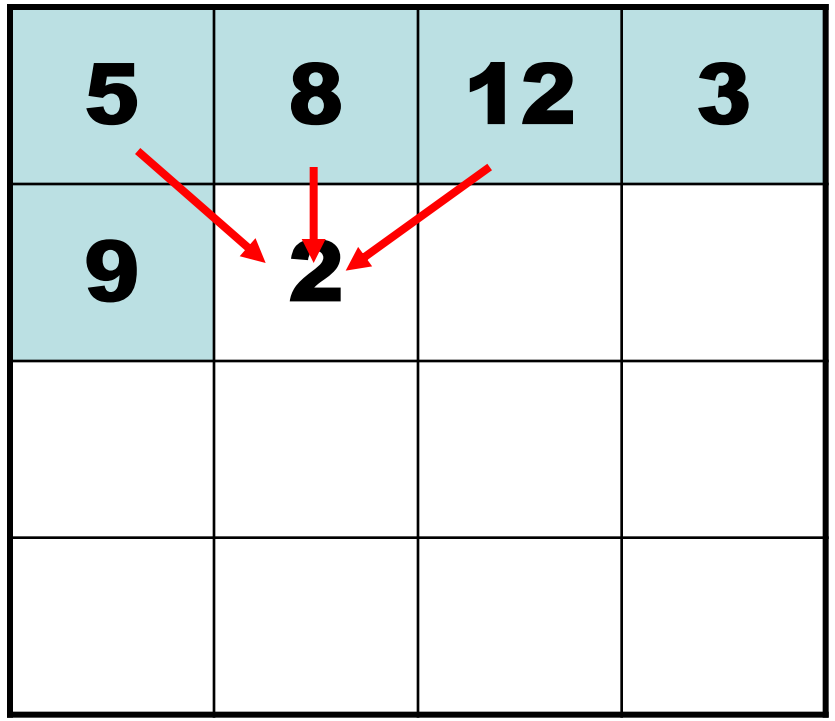
$$\mathbf{M}(i,j) = E(i,j) + \min\left(\mathbf{M}(i-1,j-1), \mathbf{M}(i-1,j), \mathbf{M}(i-1,j+1)\right)$$

- Can be solved efficiently using dynamic programming in $O(s \cdot n \cdot m)$

  (s=3 in the original algorithm)

Dynamic Programming

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



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

Energy - E(I,j)

Dynamic Programming

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

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

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

5	8	12	3
9	7	3+3	

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

Energy - E(I,j)

Dynamic Programming

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

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

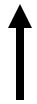
5	8	12	3
4	2	3	9
7	3	4	2
5	5	7	8

Energy - E(I,j)

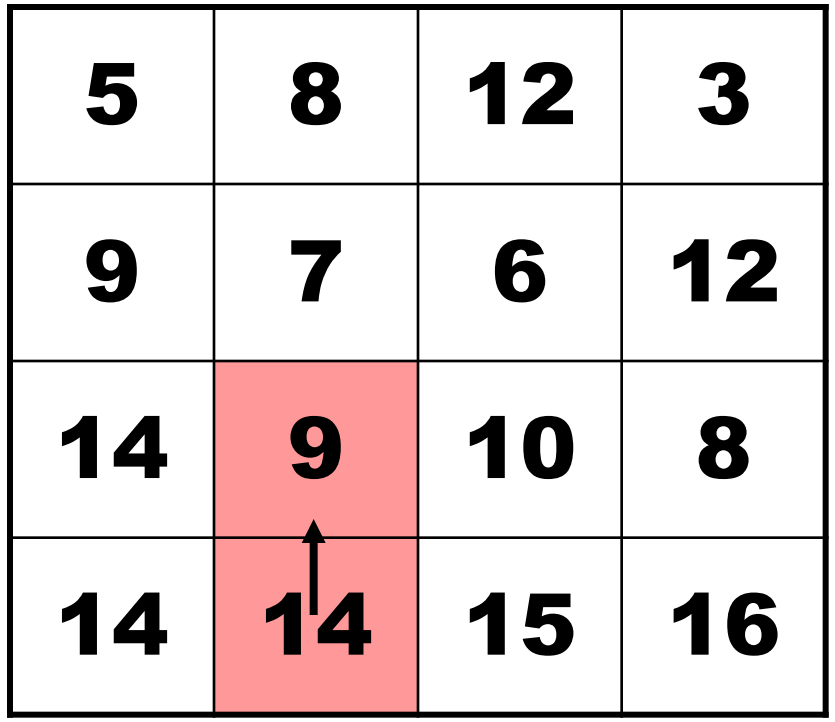
Searching for Minimum

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

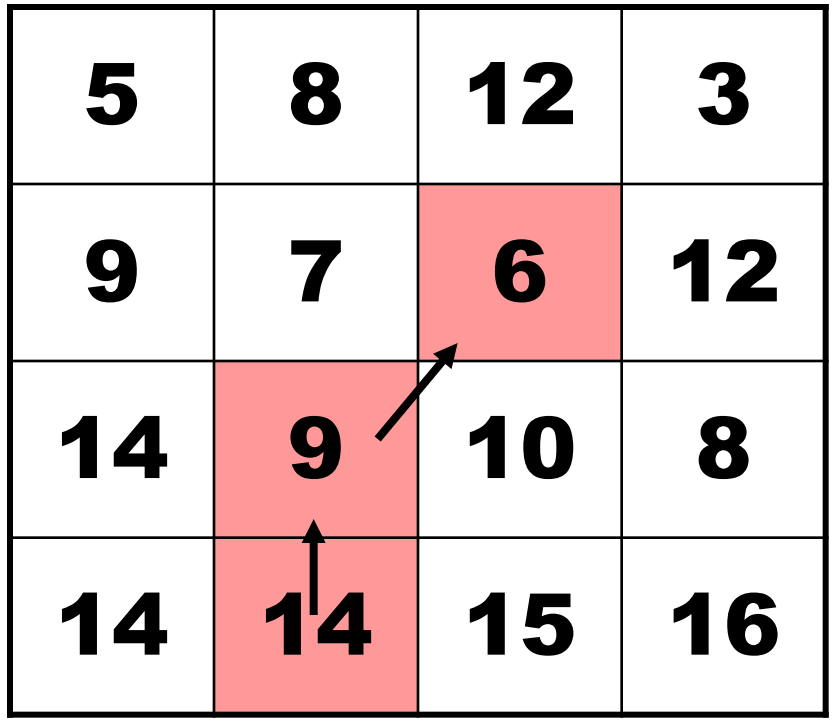
5	8	12	3
9	7	6	12
14	9	10	8
14	14	15	16



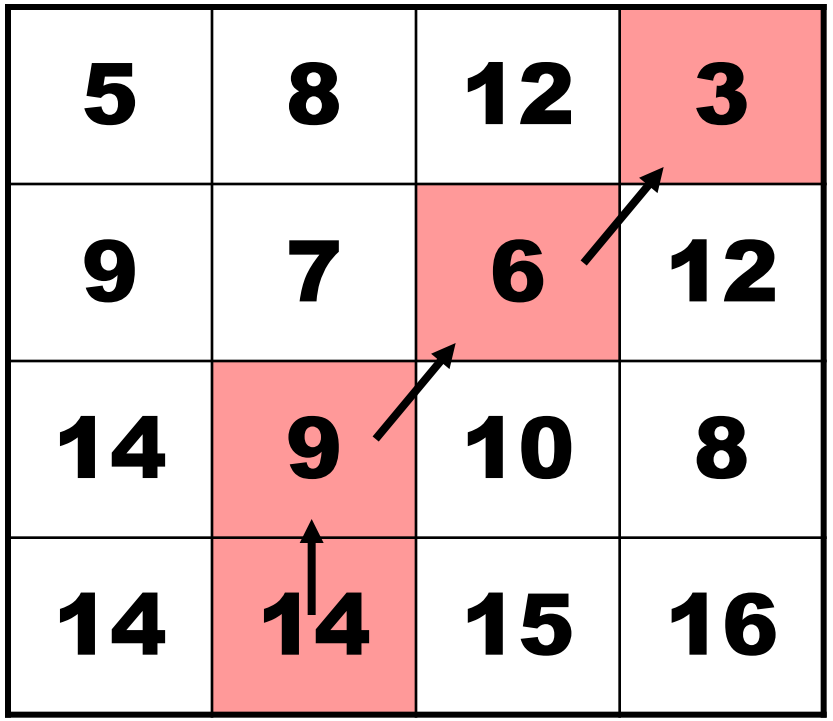
Backtracking the Seam



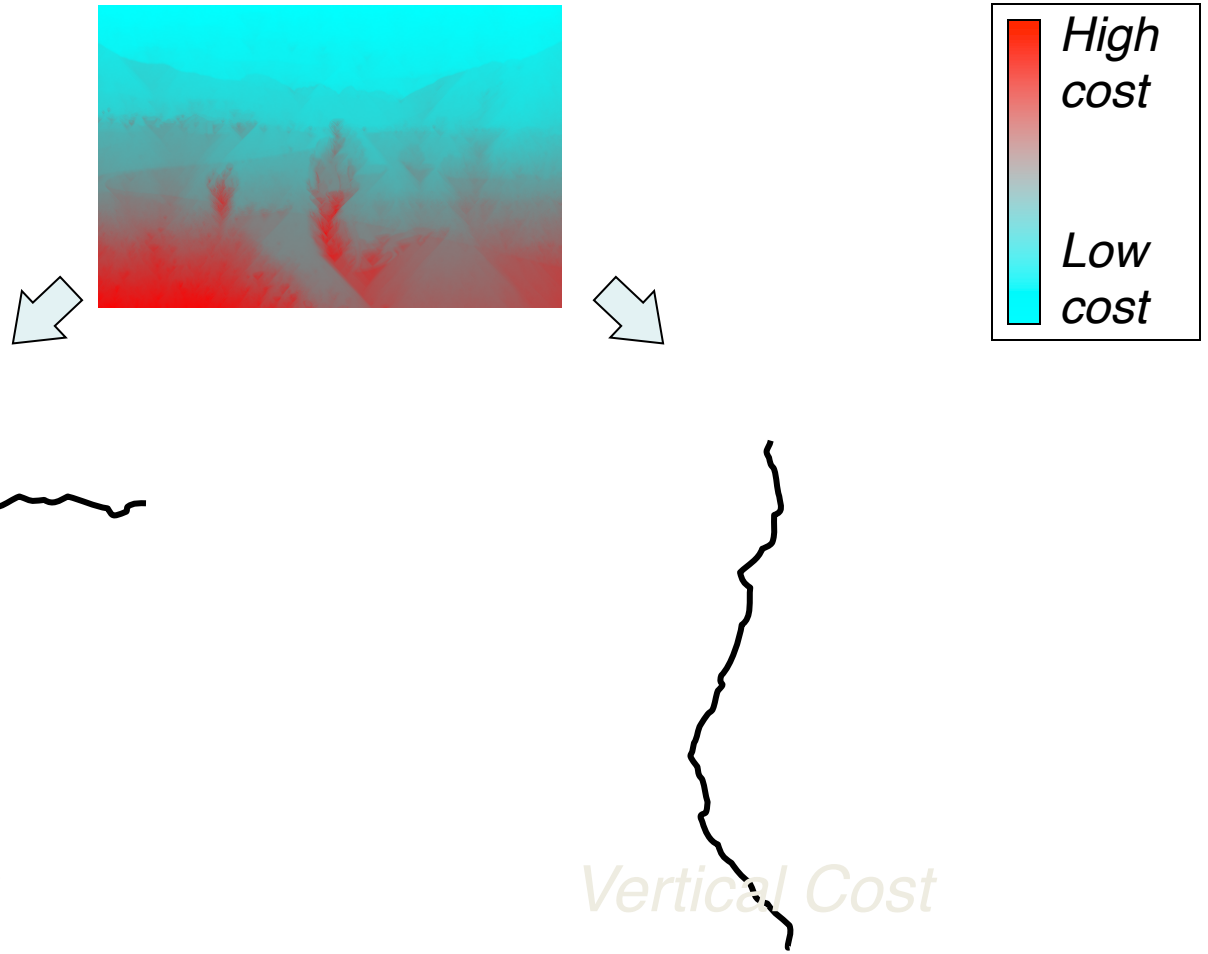
Backtracking the Seam



Backtracking the Seam



H & V Cost Maps



Seam Carving

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'

Changing Aspect Ratio

Changing Aspect Ratio

Original

Seam Carving

Scaling

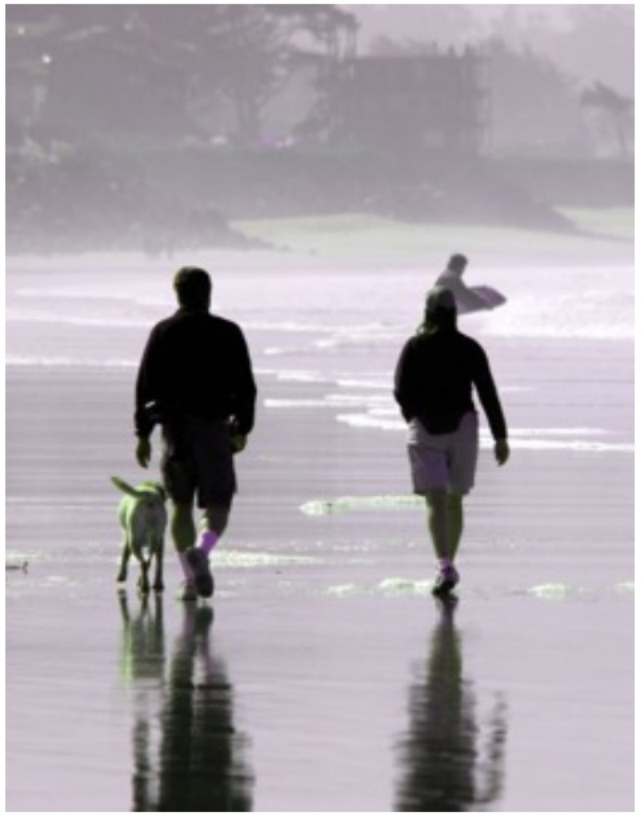
Changing Aspect ratio

Cropping

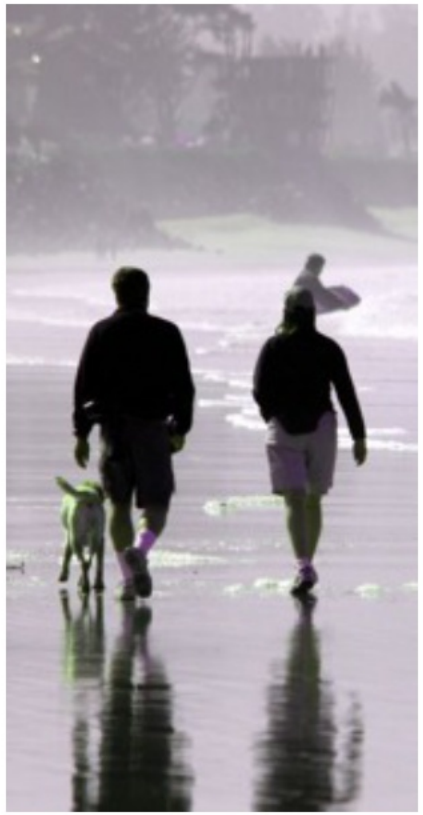
Seams

Scaling

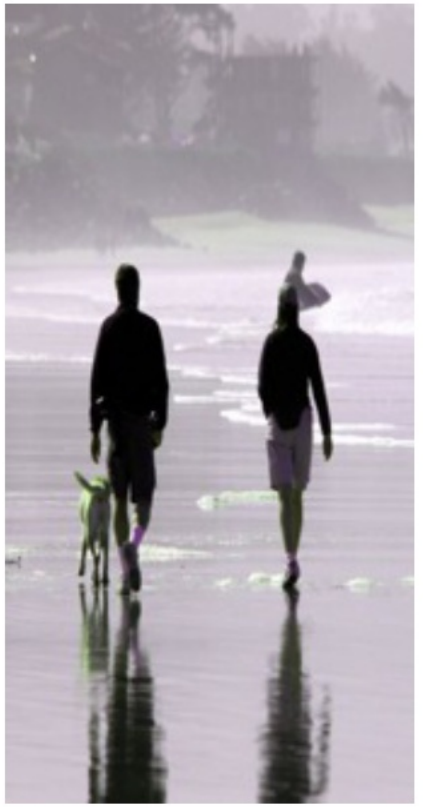
Changing Aspect Ratio



Original



Retarget

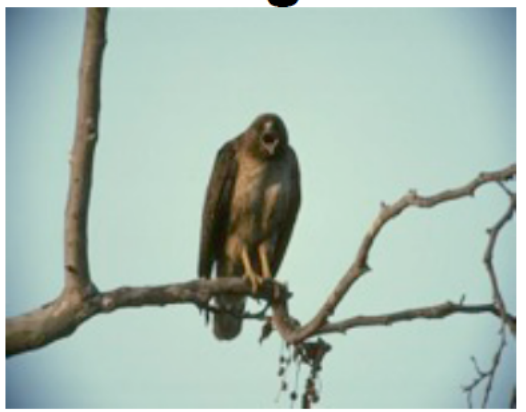


Scaling

Changing Aspect Ratio

Original

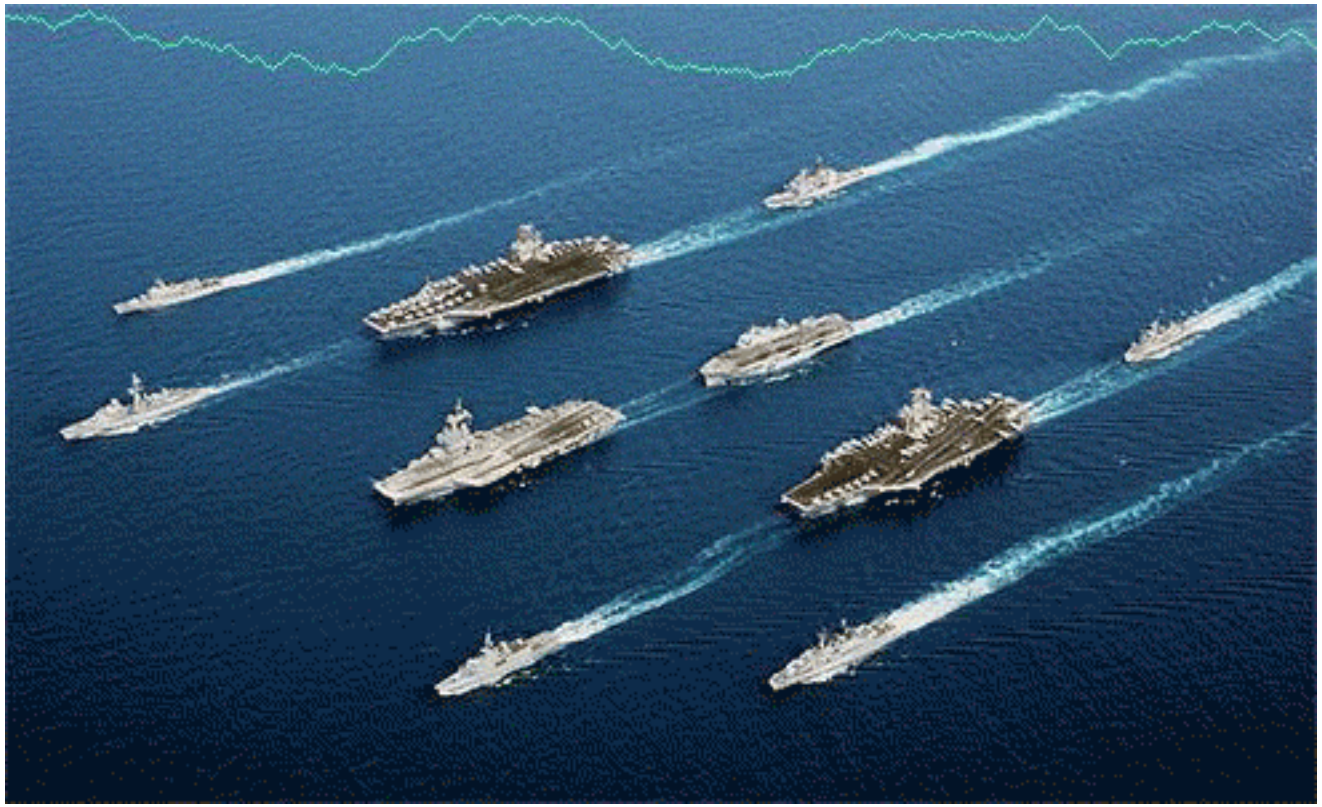
Retarget



Scaling

Example seam carving

Another example



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 accounting for locality of operations

A Local Operator

Questions?

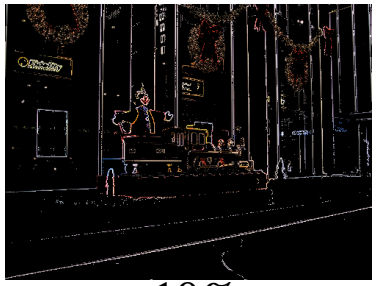
- Q: Will the result be the same if the image is flipped upside down?
- A: Yes (up to numerical stability)

- Q: Can we improve the running time?
- A: Yes, by accounting for locality of operations

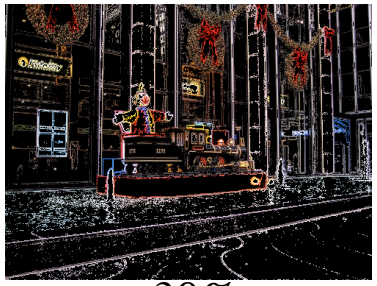
- Q: What happens to the overall energy in the image during seam carving?

Preserved Energy

Energy



10%



30%

40%

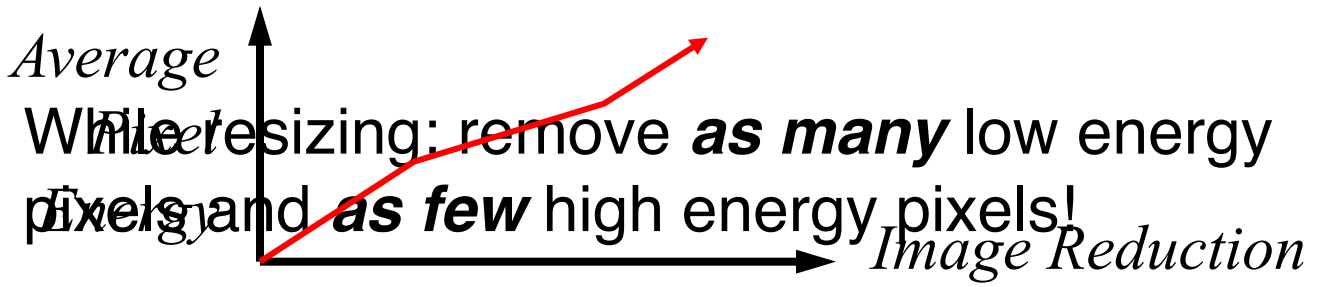
75%

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 ↑

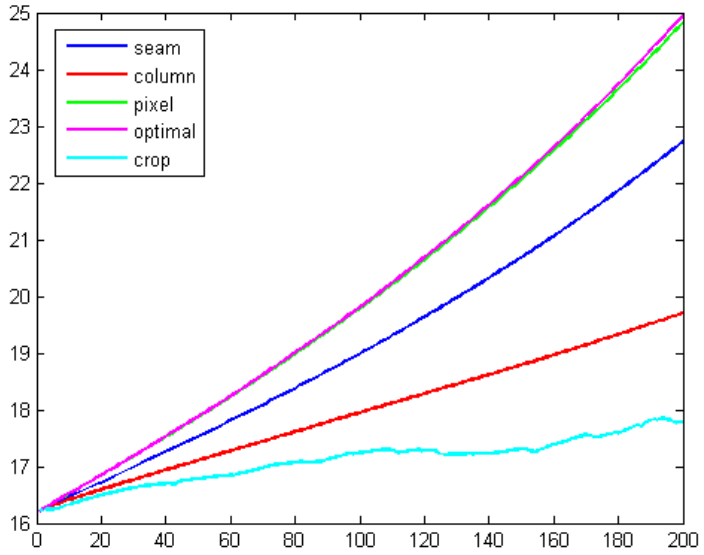


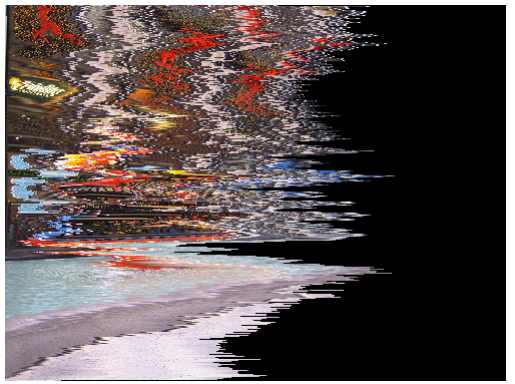
Image Reduction →

crop

column

seam

pixel



optimal

Extension to Both Dimensions?

- m x n → m’ x n’

- Remove horizontal seam first?
- Remove vertical seams first?
- Alternate between the two?
- The optimal order can be found! → Dynamic Prog (again)

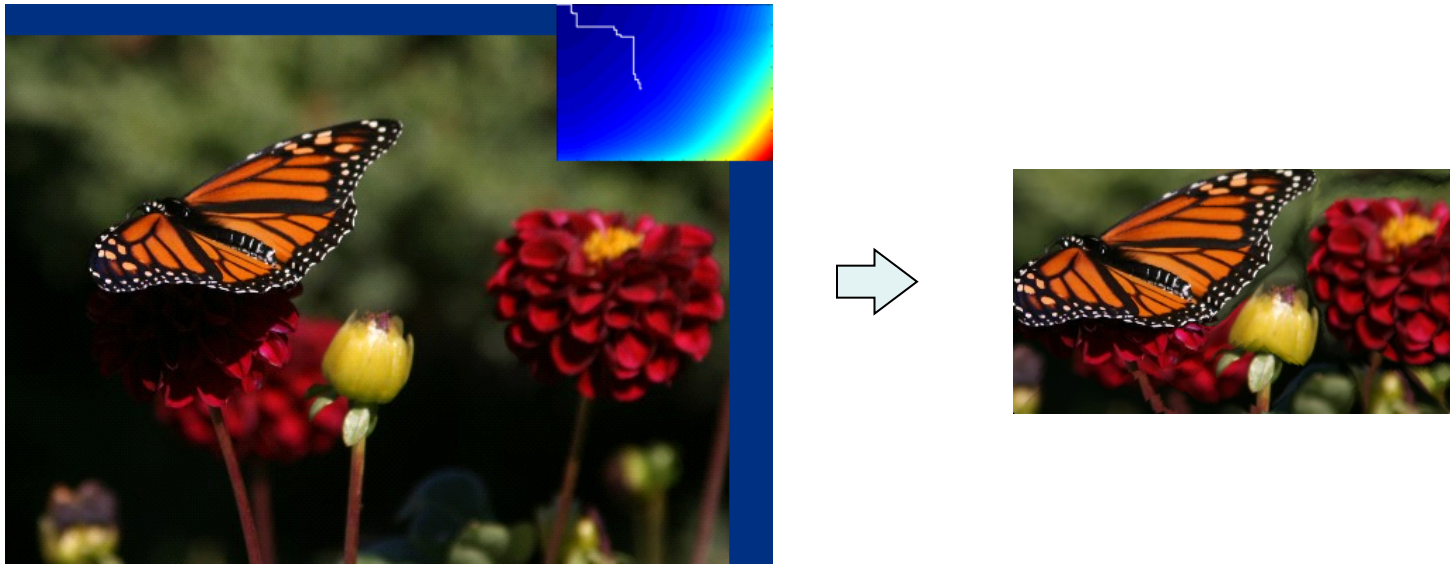


Image Expansion (Synthesis)

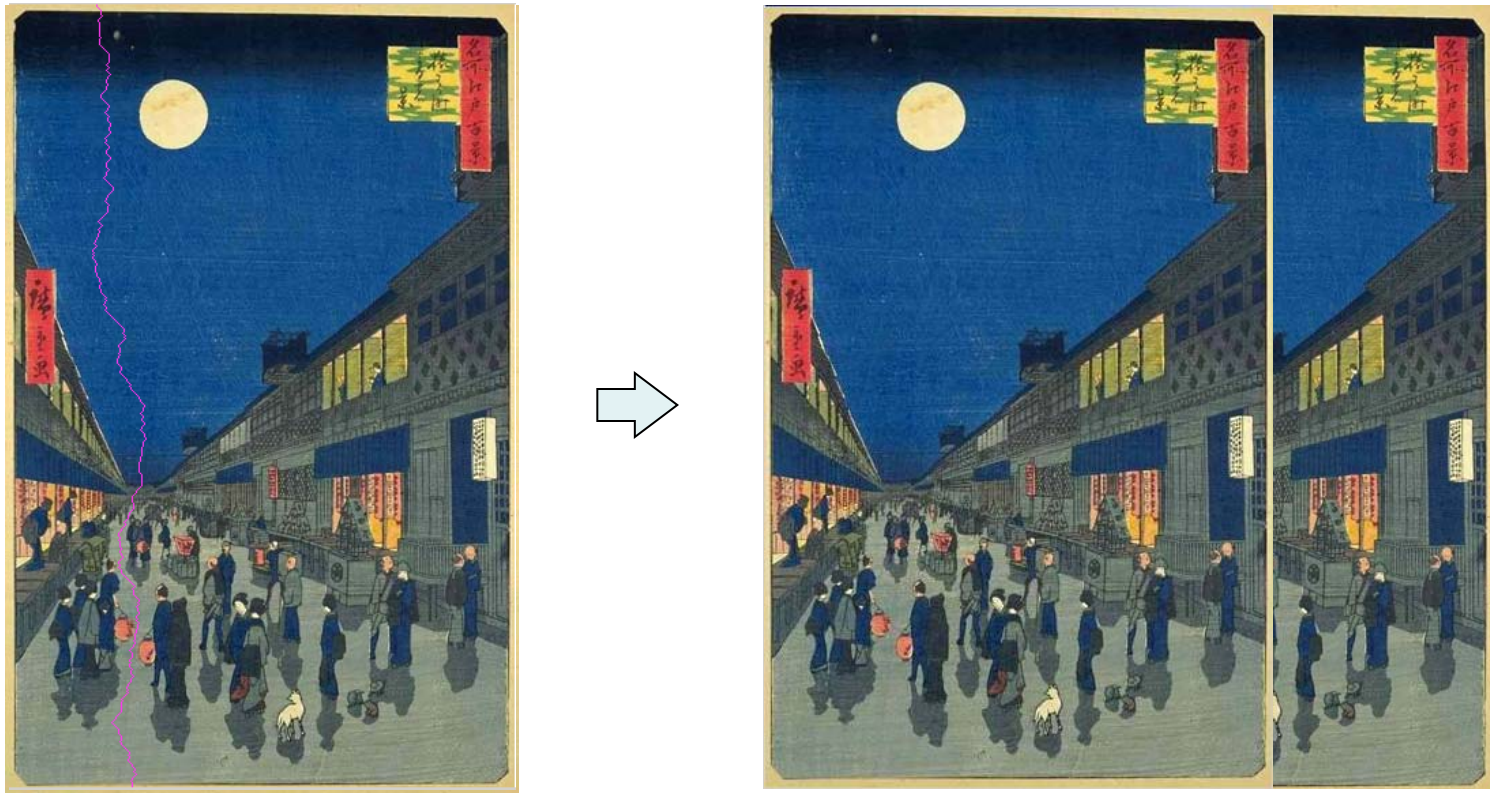
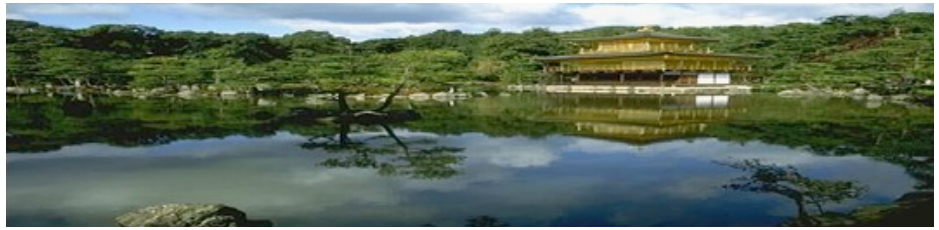


Image Expansion – take 2

Scaling

Combined Insert and Remove

Insert & remove seams



Scaling

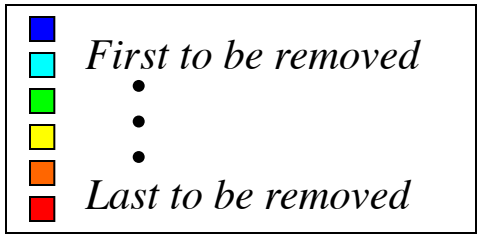
Questions?

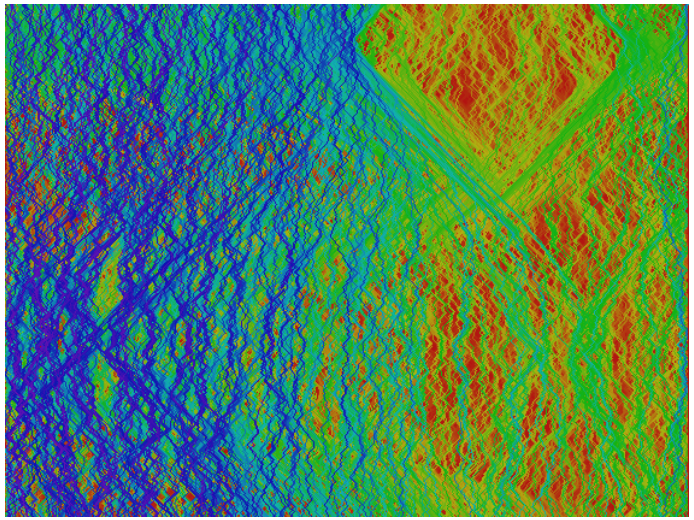
Multi-Size Images

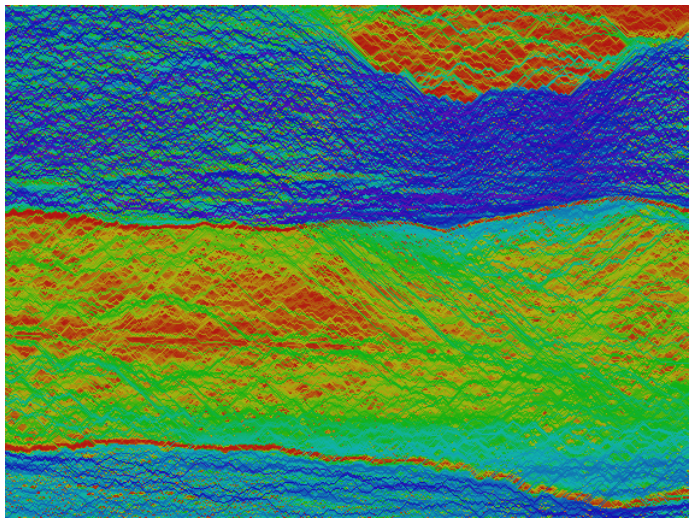
- We can create a new *representation* of an image that will allow adapting it to different sizes!

1. Precompute all seams once
2. Realtime resizing / transmit with content

Multi-Size Images

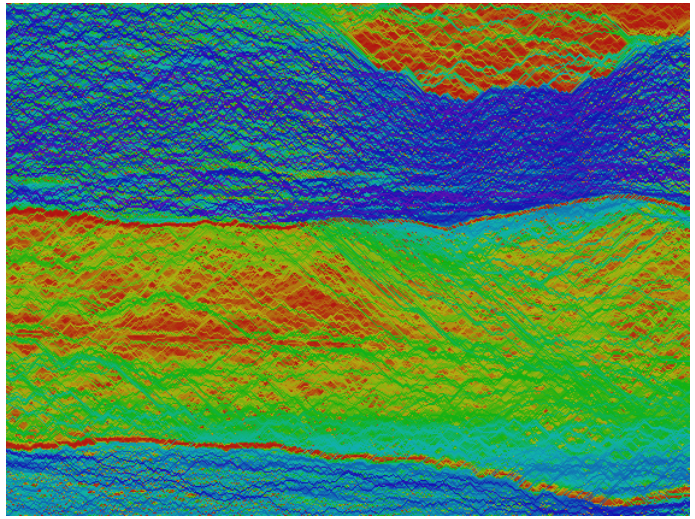




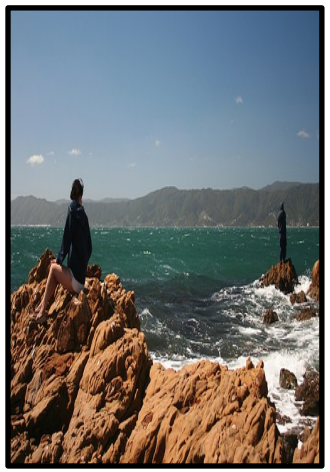


Multi-Size Image Representation

+



Multi-Size Image Representation



Object Removal

Auxiliary Energy

- Recall our seam equation

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

Object Removal

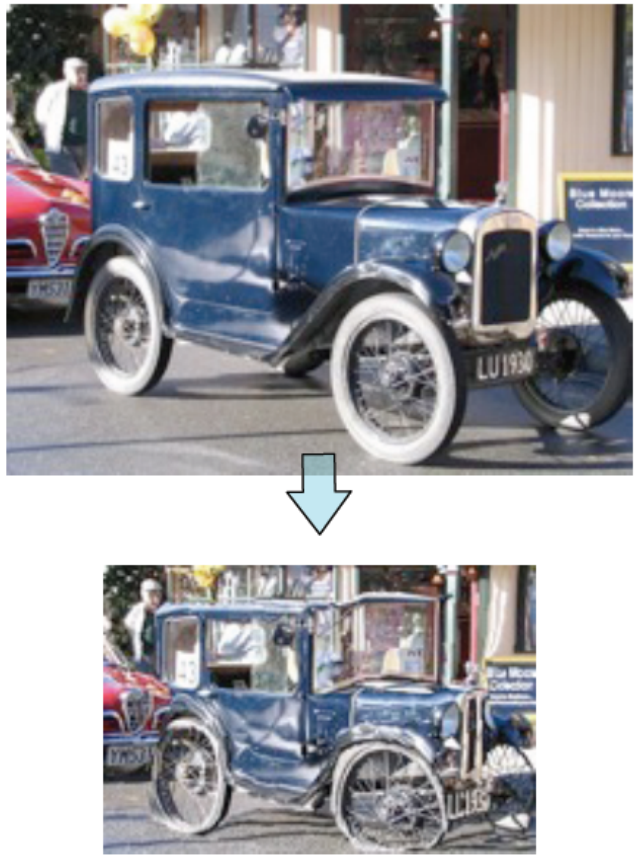
Input | Retargeted | Pigeon Removed | Girl Removed

Find the Missing Shoe!

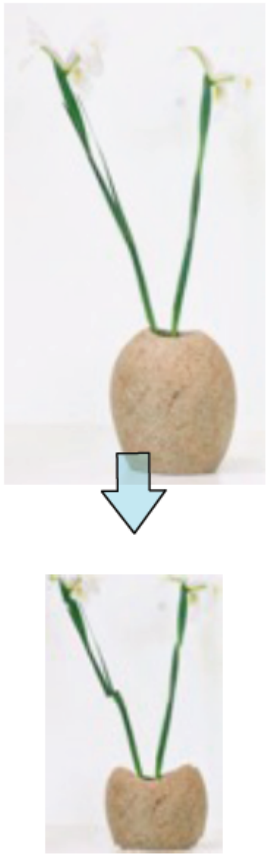
Solution

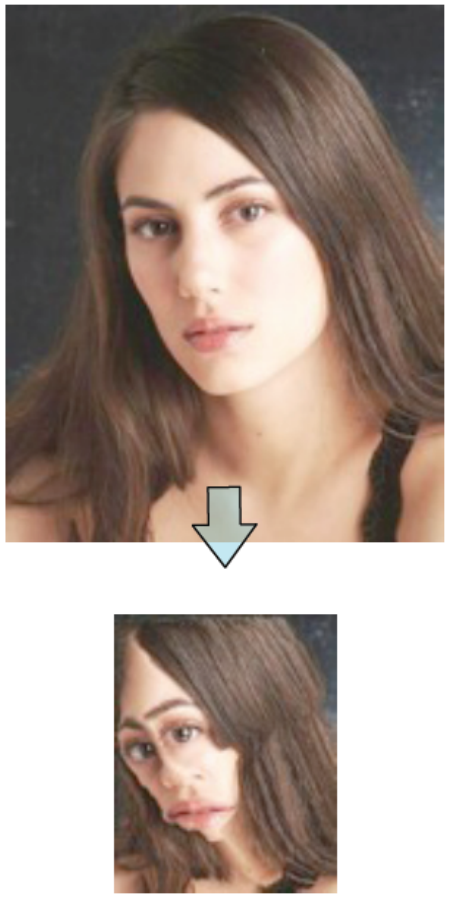
Limitations

Content



Structure





With face detector

With User Constraints

Preserved Energy - Revisited

Average Pixel Energy ↑

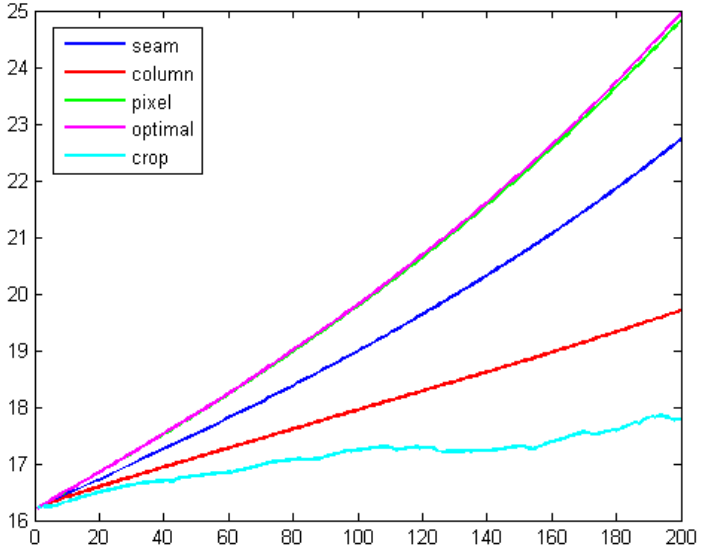


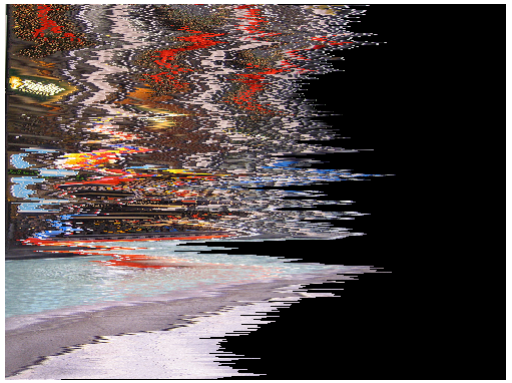
Image Reduction →

crop

column

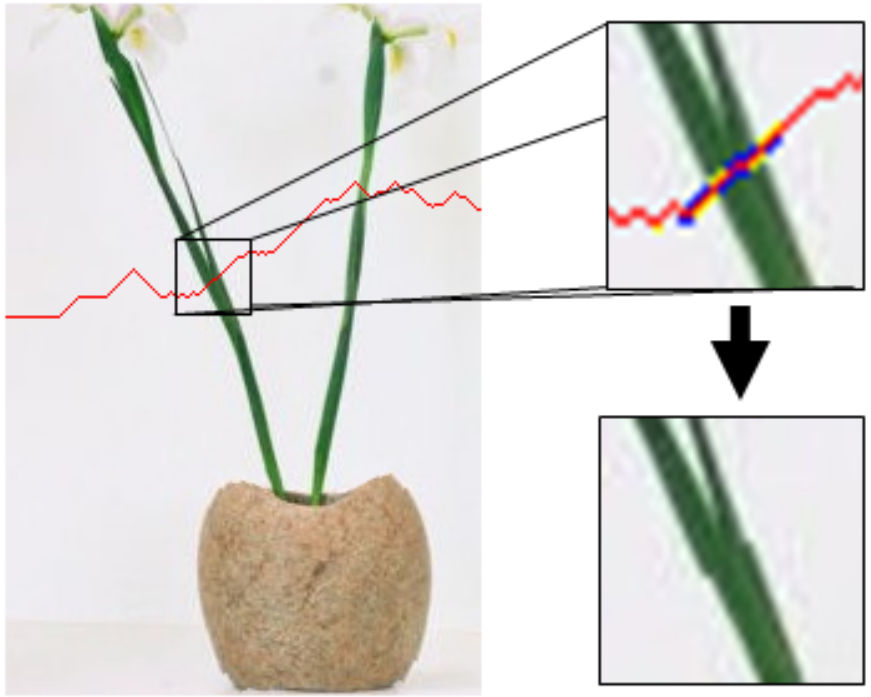
seam

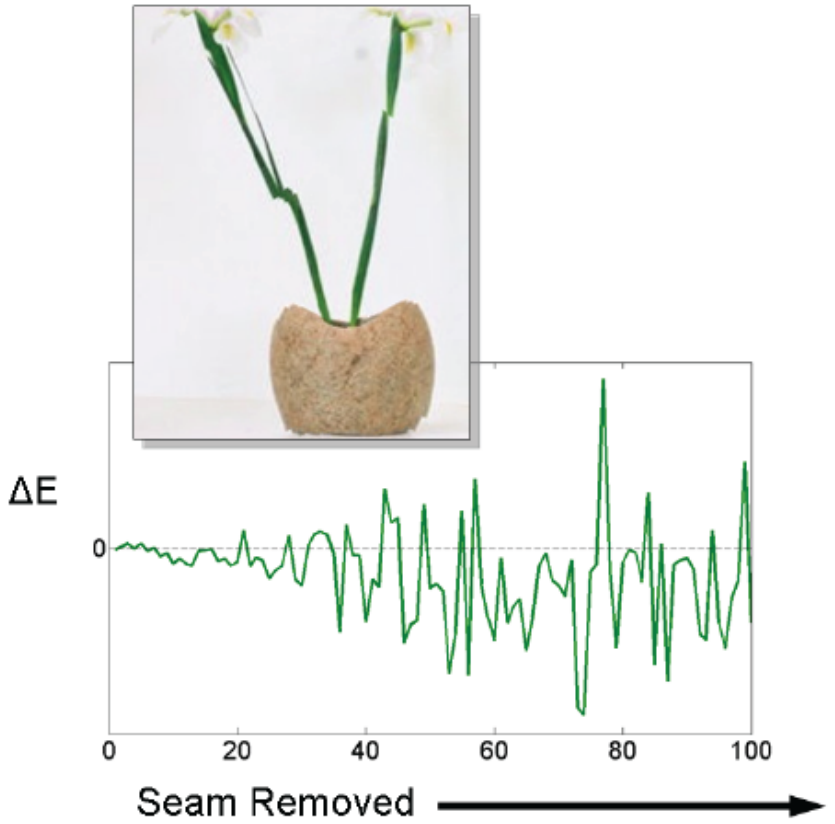
pixel



optimal

Inserted Energy

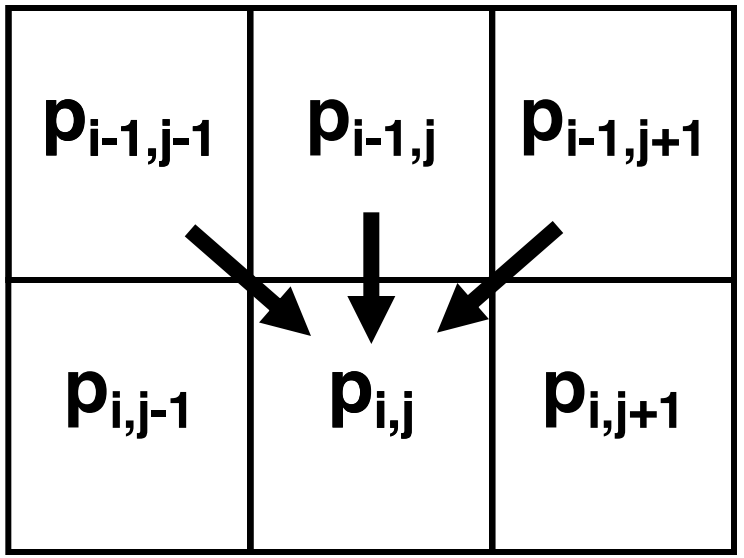




Minimize Inserted Energy

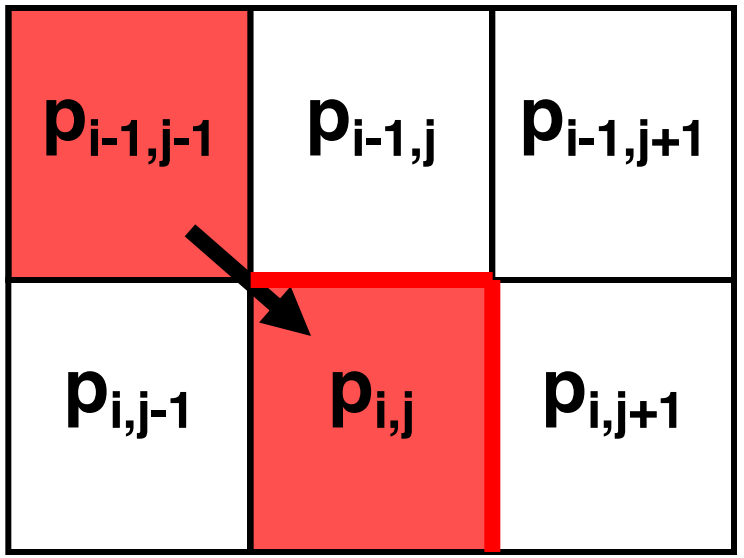
- Instead of removing the seam of least energy, remove the seam that *inserts the least energy* to the image !

Tracking Inserted Energy



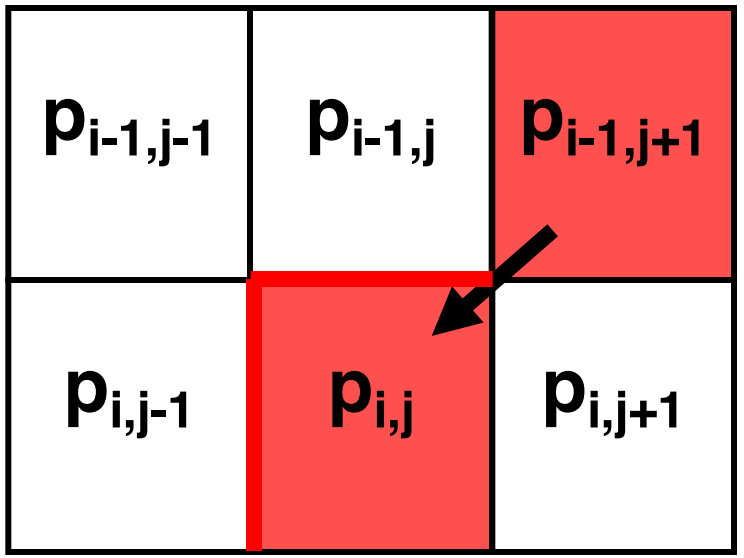
- Three possibilities when removing pixel $P_{i,j}$

Pixel $P_{i,j}$: Left Seam



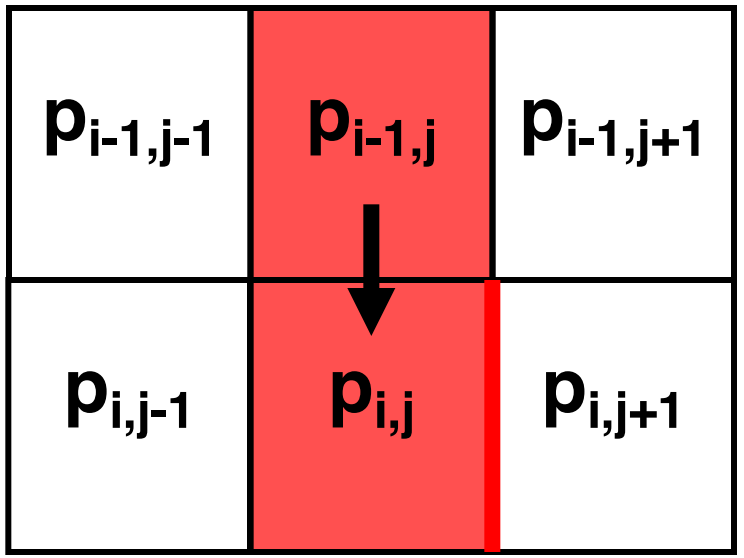
$$C_L(i,j) = |I(i,j+1) - I(i,j-1)\| + |I(i-1,j) - I(i,j-1)|$$

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



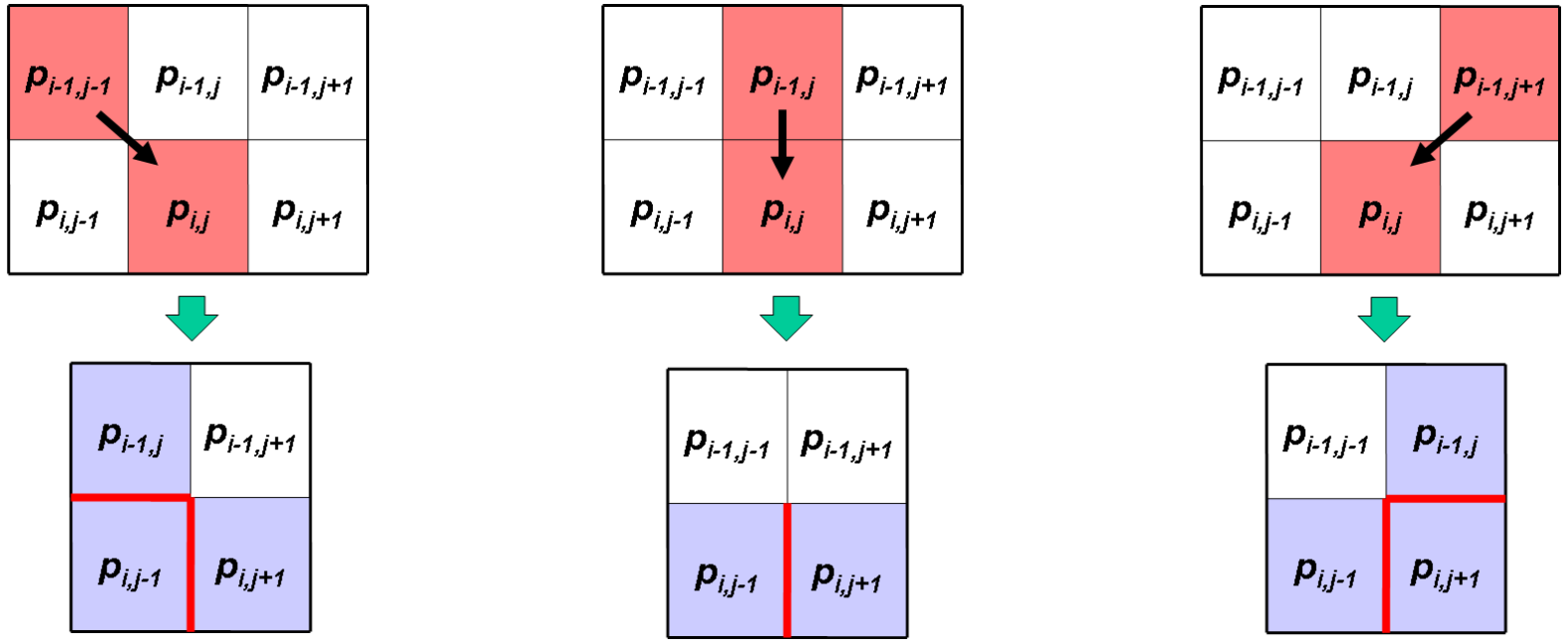
$$C_V(i,j) = |I(i,j+1) - I(i,j-1)|$$

Old “Backward” Energy

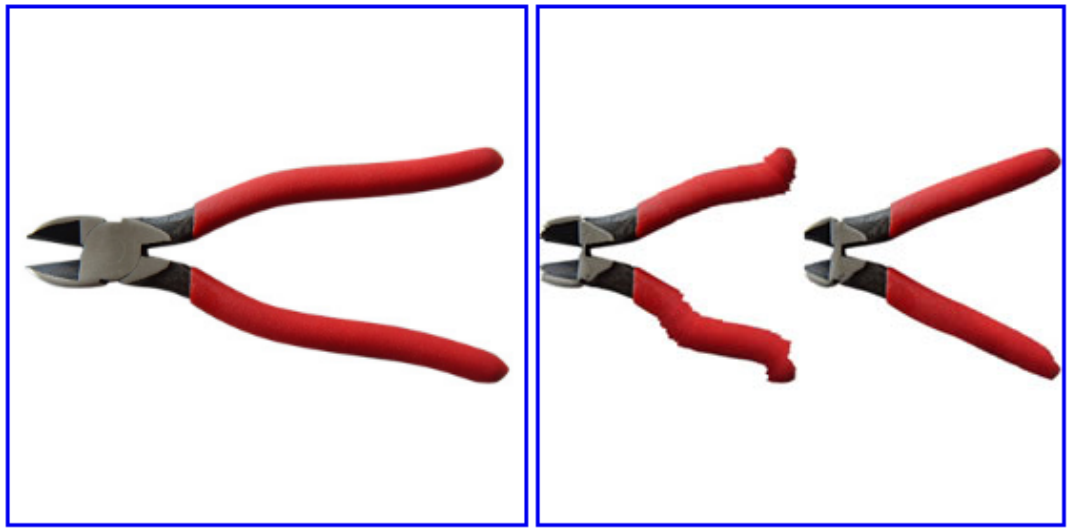
$$M(i,j) = E(i,j) + \min \begin{cases} M(i-1,j-1) \\ M(i-1,j) \\ M(i-1,j+1) \end{cases}$$

New Forward Looking Energy

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



Results



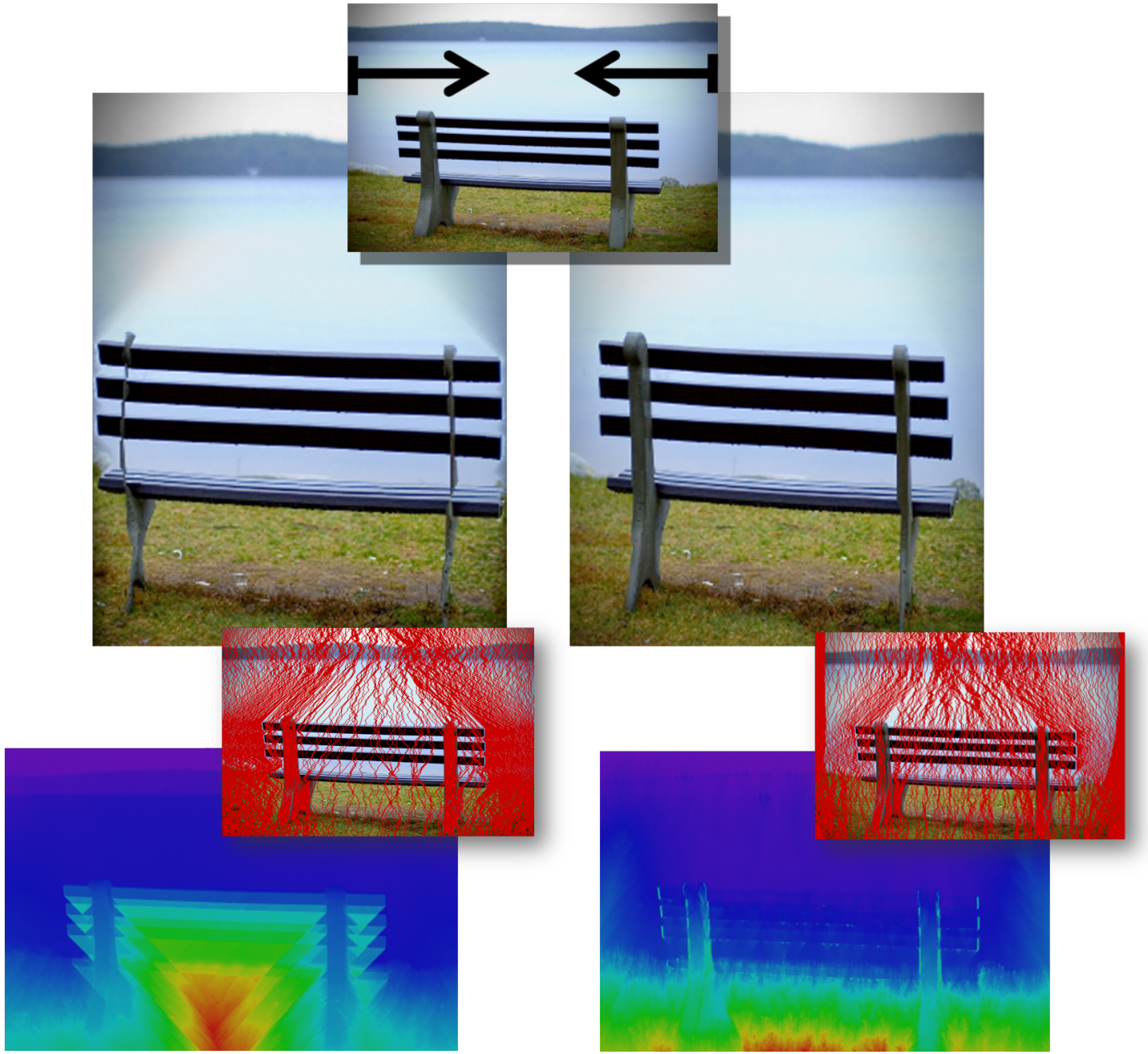
Input Backward Forward

Input

Backward

Forward

Results

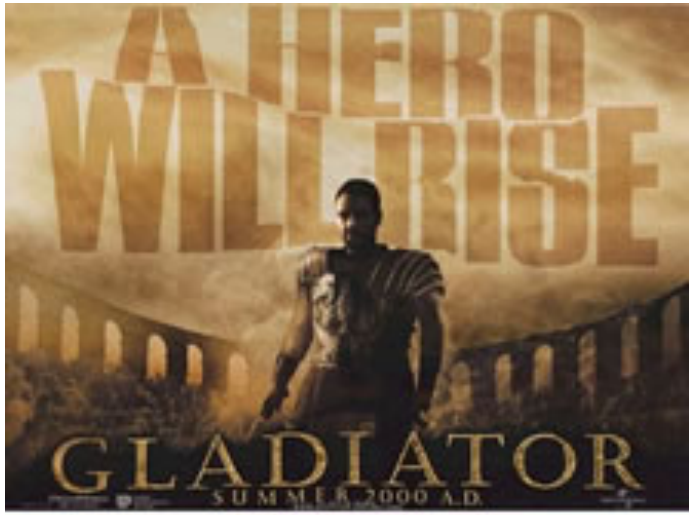


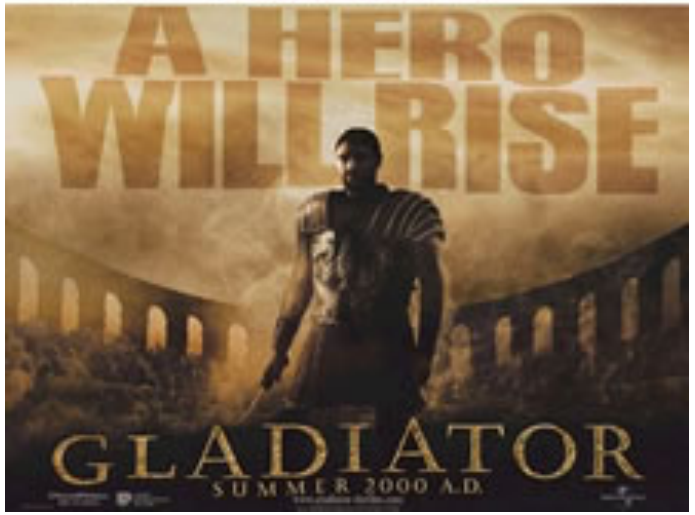
Backward vs. Forward

Backward

Forward

Results


A HERO WILL RISE
GLADIATOR
SUMMER 2000 A.D.


A HERO WILL RISE
GLADIATOR
SUMMER 2000 A.D.

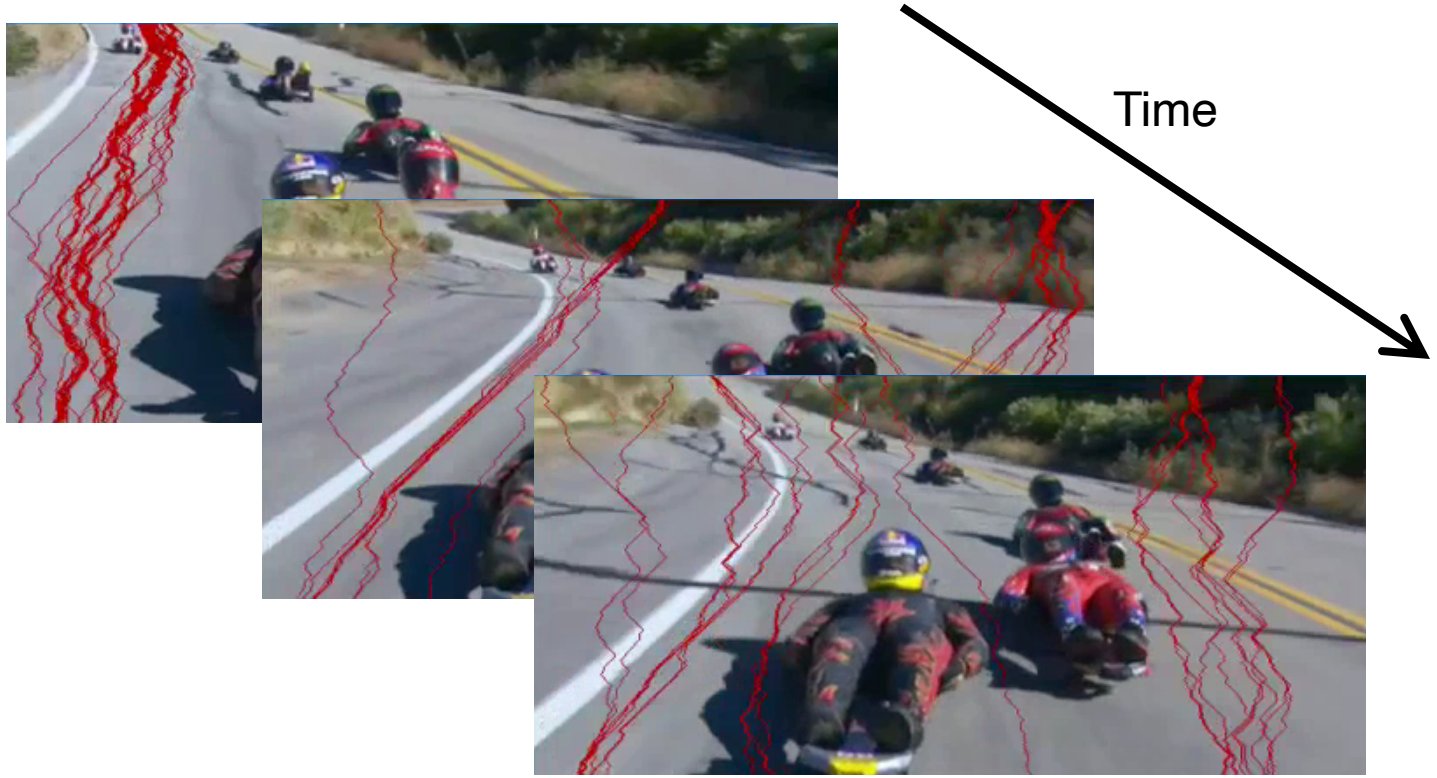
From Images to Videos

- 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!

Seam-Carving Video?

- Naive… frame by frame independently

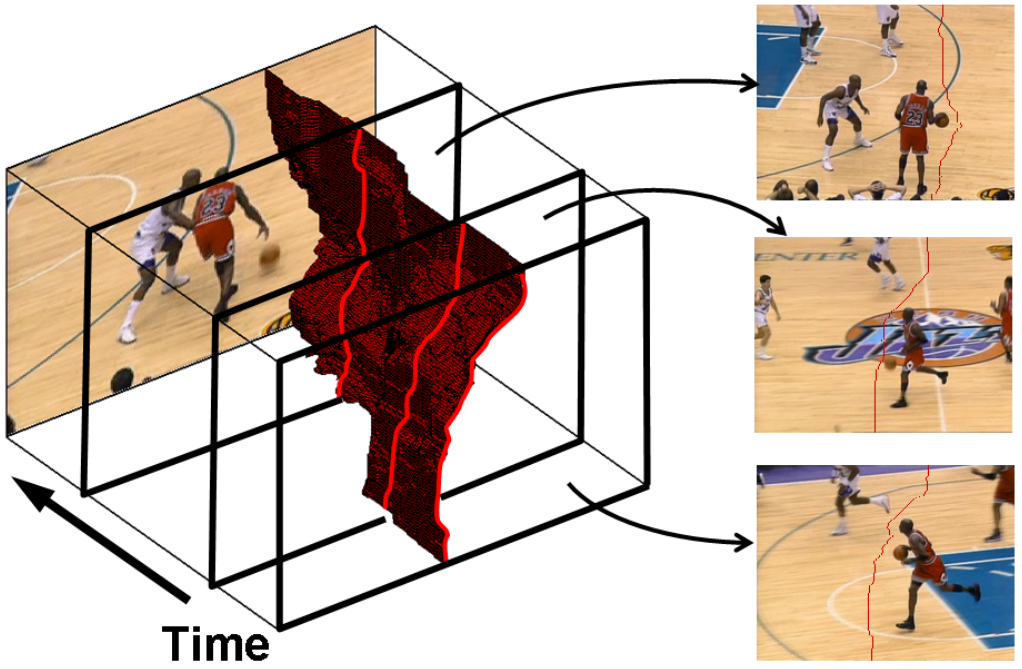


Frame-by-frame Seam-Carving

Let’s check out this video

From 2D to 3D

1D paths in images



2D manifolds in video cubes

Example video retargeting

Object Remove

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: http://help.adobe.com/en_US/Photoshop/11.0/WS6F81C45F-2AC0-4685-8FFD-DBA374BF21CD.html