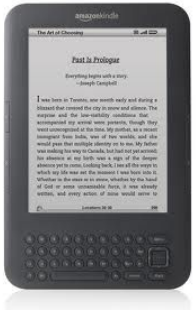
CSCE 448/748 - Computational Photography

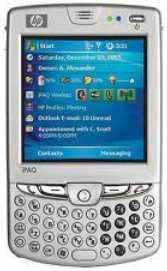
Image retargeting

Nima Kalantari

Slides by Michael Rubenstein, Ariel Shamir, and Shai Avidan

Display Devices



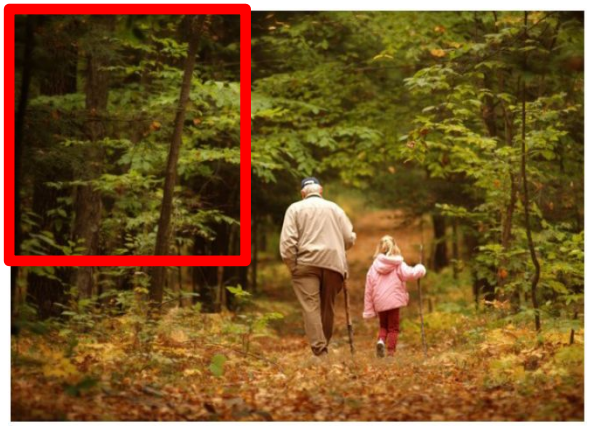


Simple Media Retargeting Operators

Letterboxing

Simple Media Retargeting Operators

- **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)**

Simple Media Retargeting Operators

Michael Rubinstein — MIT CSAIL – mrub@mit.edu

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!**

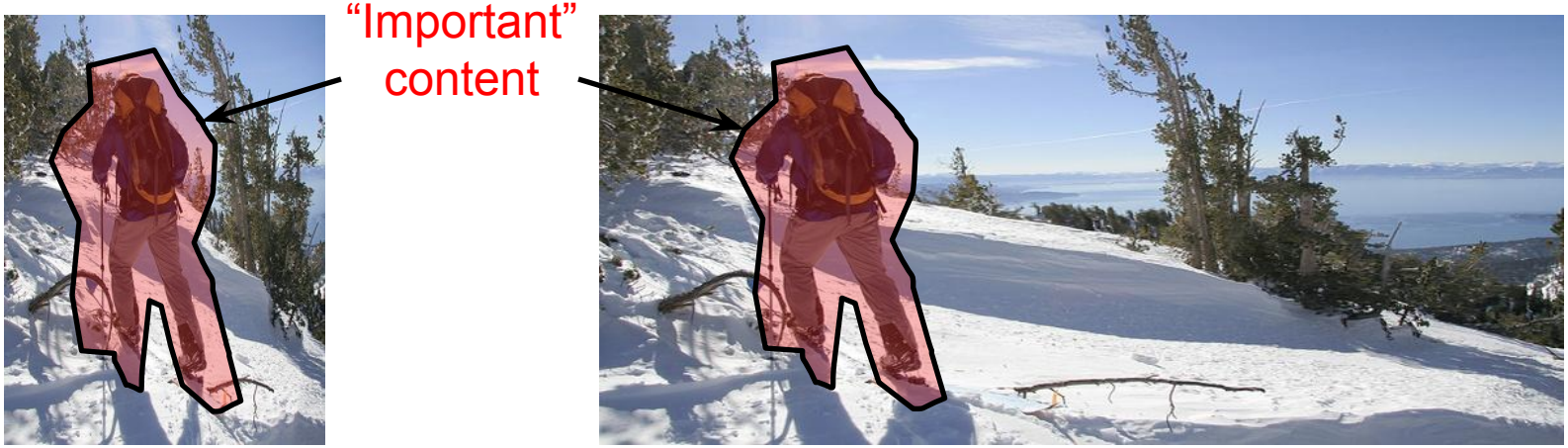
Example

Scaling

Content-aware Retargeting Operators

Content-*aware*

"Important" content



Content-*oblivious*

Michael Rubinstein — MIT CSAIL – mrub@mit.edu

Content-aware Retargeting

Input

Scale

Crop

Content-aware

“less-Important” content

Seam Carving

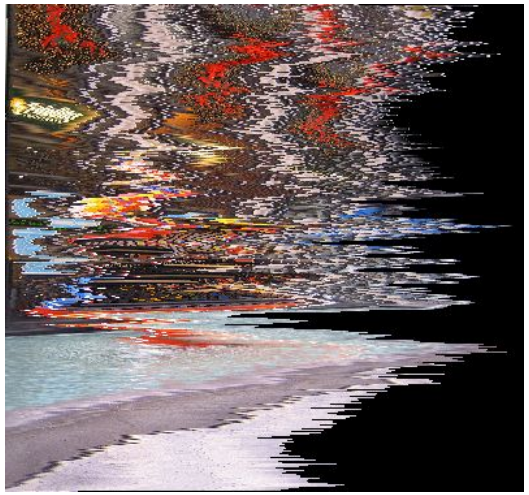
- **Assume n x m □ n x m', m'<m (summarization)**

- **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
  - See their paper for more measures they have used

Pixel Removal



Optimal

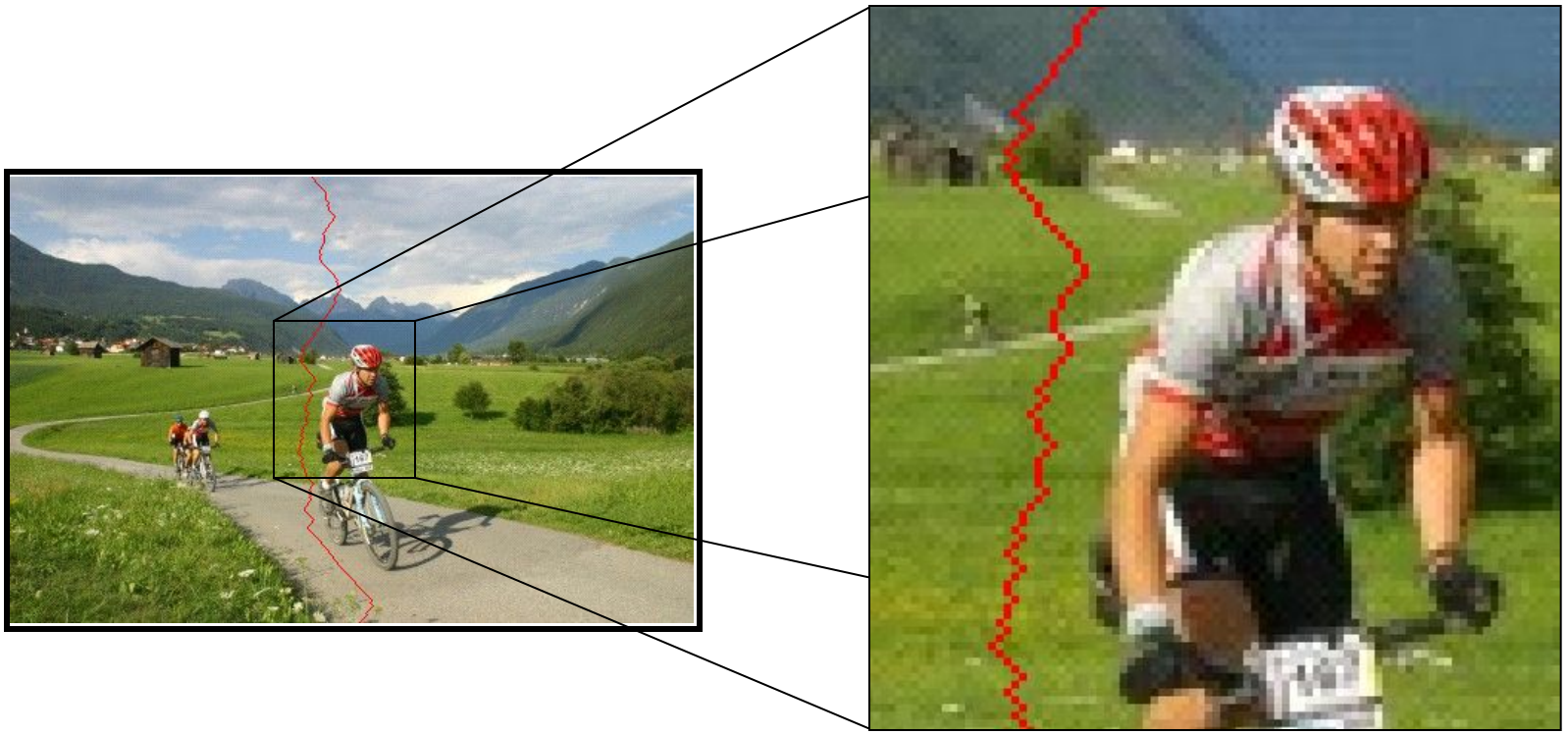
Least-energy pixels (per row)

Least-energy columns

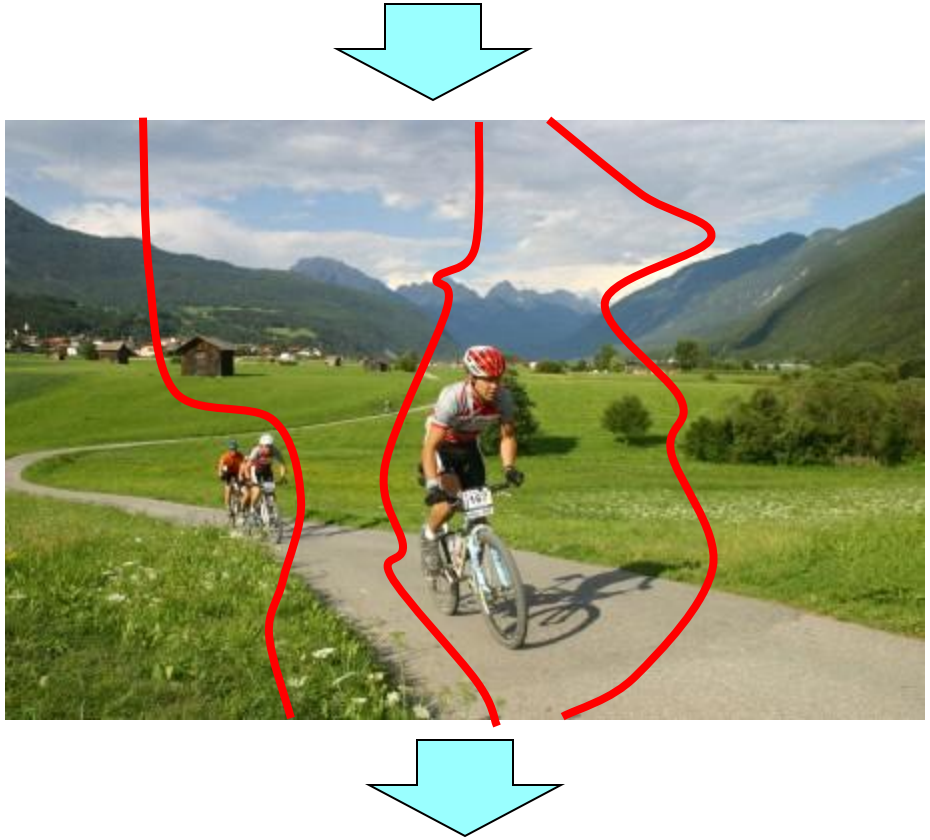
A Seam

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

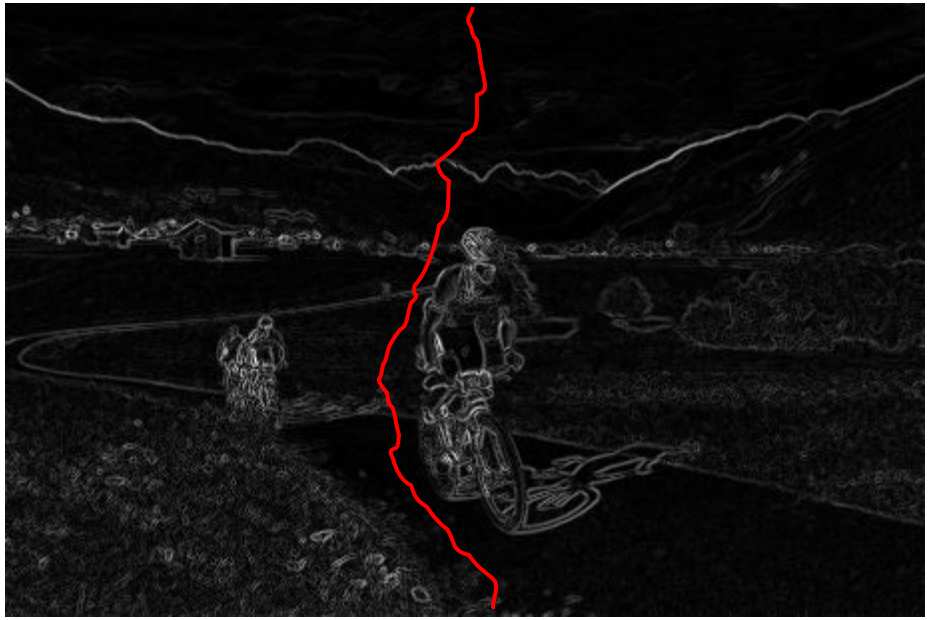
$$\mathbf{s^x} = \{s_i^x\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n, \text{ s.t. } \forall i, |x(i) - x(i-1)| \leq 1$$
$$\mathbf{s^y} = \{s_j^y\}_{j=1}^m = \{(j, y(j))\}_{j=1}^m, \text{ s.t. } \forall j |y(j) - y(j-1)| \leq 1$$



Finding the Seam?



The Optimal Seam



$$E(\mathbf{I}) = \left|\frac{\partial}{\partial x}\mathbf{I}\right| + \left|\frac{\partial}{\partial y}\mathbf{I}\right| \quad \Rightarrow \quad s^* = \underset{s}{\text{argmin}} E(s)$$

The Optimal Seam

- **The recursion relation**

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

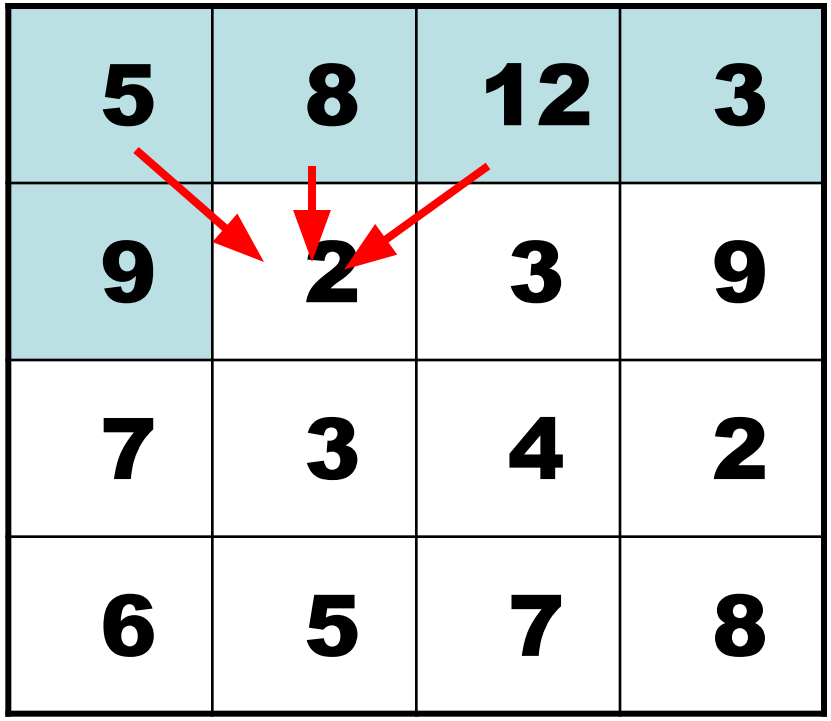
- **Can be solved efficiently using dynamic programming in**

$O(s \cdot n \cdot m)$

(s=3 in the original algorithm)

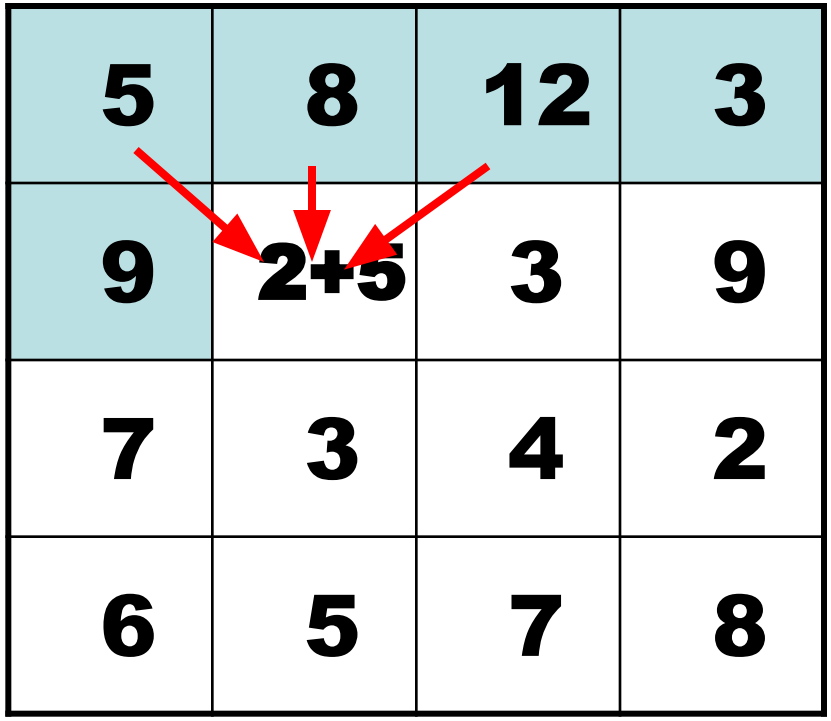
Dynamic Programming

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



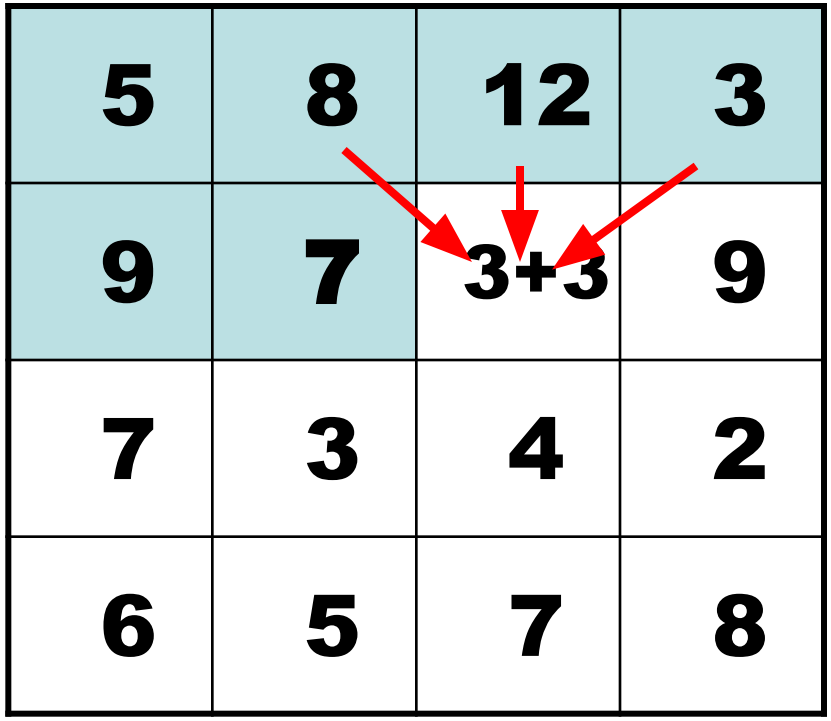
Dynamic Programming

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



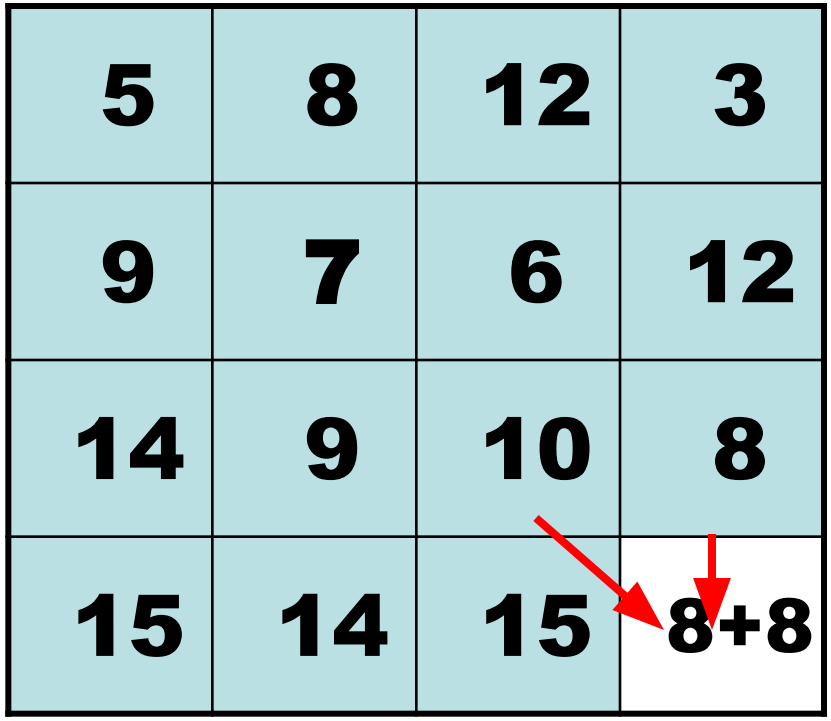
Dynamic Programming

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



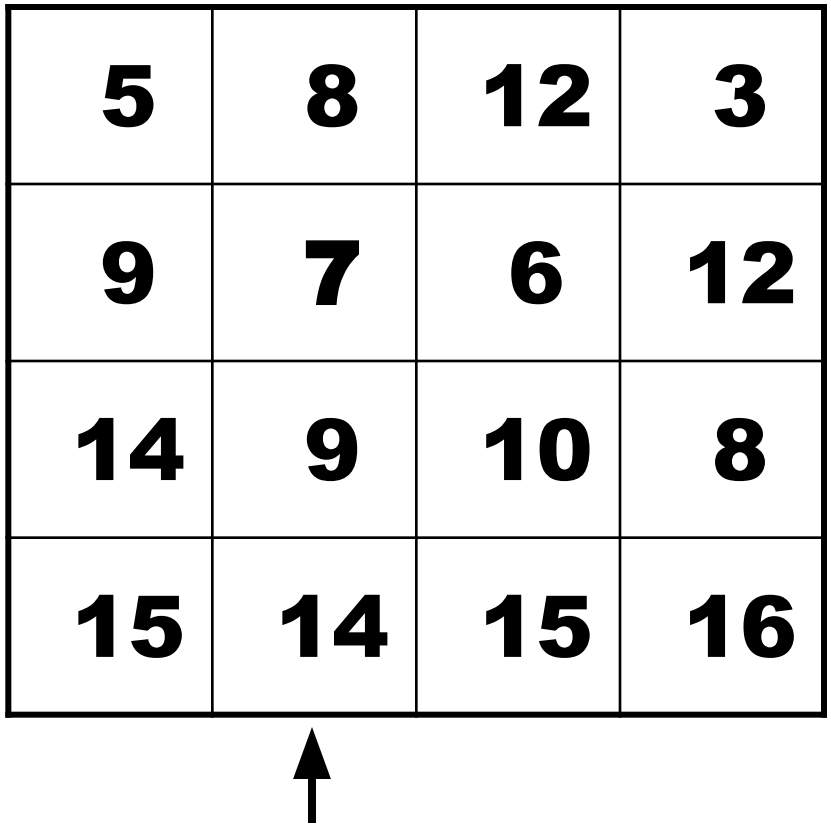
Dynamic Programming

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

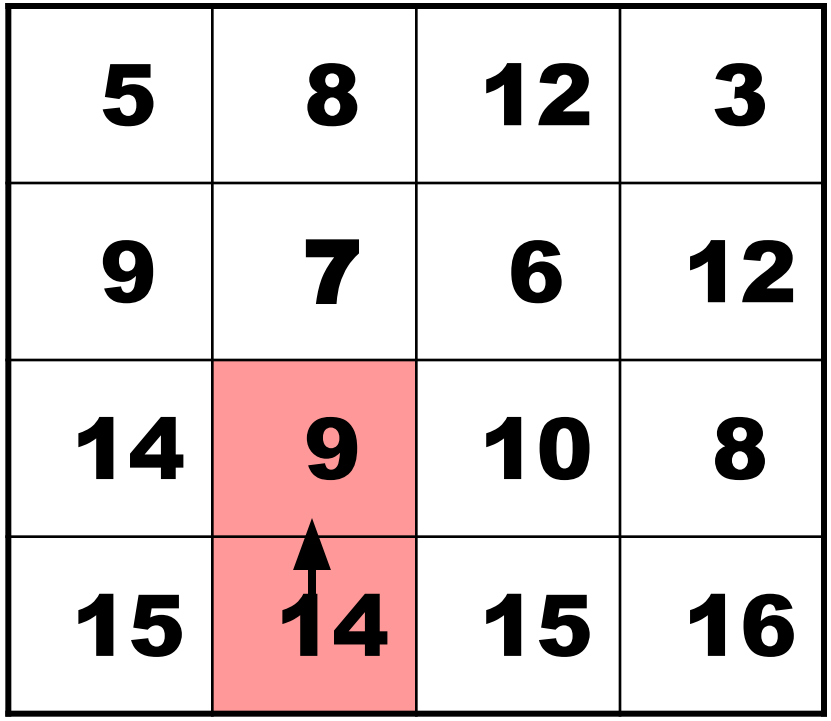


Searching for Minimum

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

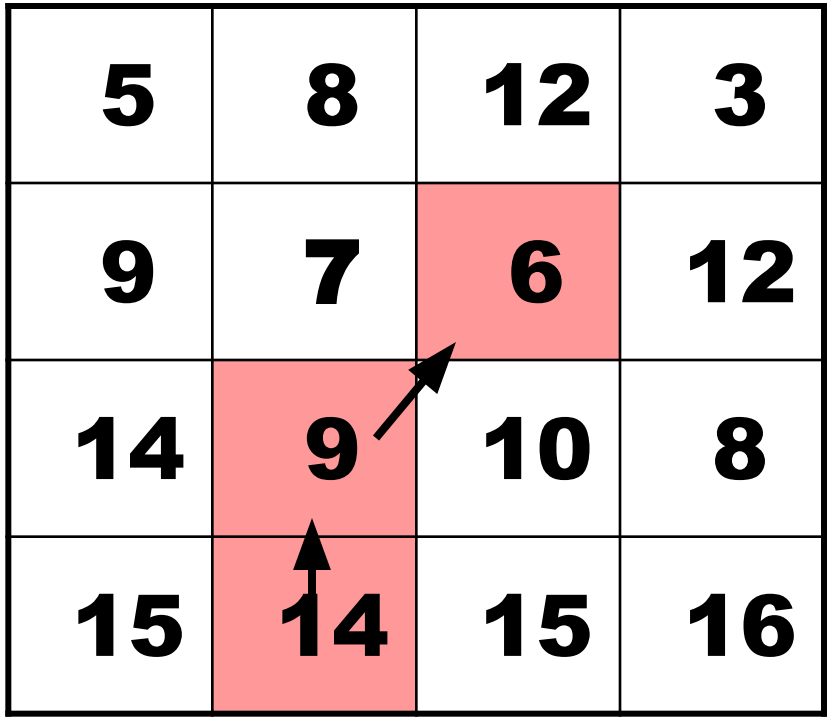


Backtracking the Seam



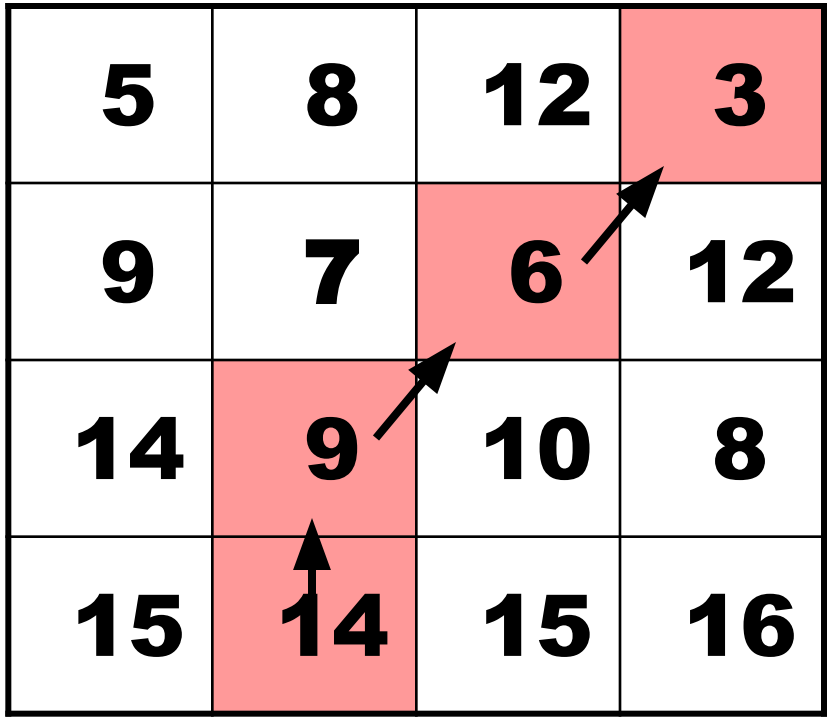
Michael Rubinstein — MIT CSAIL – mrub@mit.edu

Backtracking the Seam



Michael Rubinstein — MIT CSAIL – mrub@mit.edu

Backtracking the Seam



Michael Rubinstein — MIT CSAIL – mrub@mit.edu

The Seam-Carving Algorithm

SEAM-CARVING(im,m') // size(im) = nxm

1. **Do (m-m') 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)

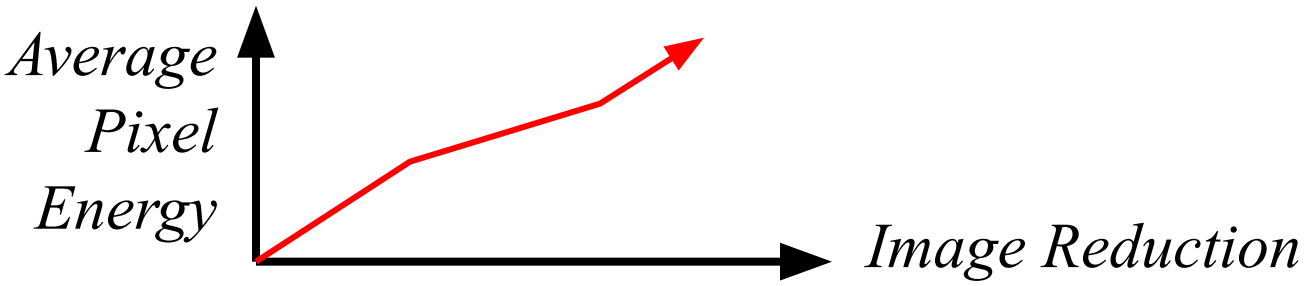
  □ O(dmn) d=m-m'

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!



Preserved Energy

Average Pixel Energy

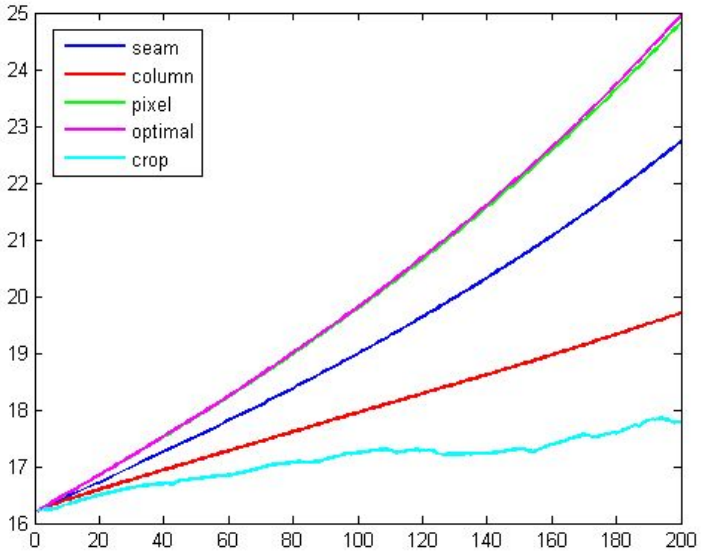


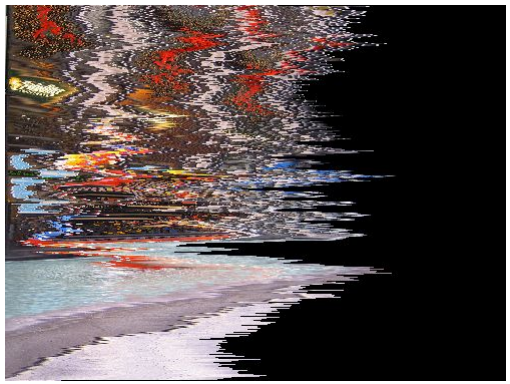
Image Reduction →

crop

column

seam

pixel



optimal

Michael Rubinstein — MIT CSAIL – mrub@mit.edu

Changing Aspect Ratio

Changing Aspect Ratio

Original

Seam Carving

Scaling

Changing Aspect Ratio

Original

Retarget

Scaling

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)

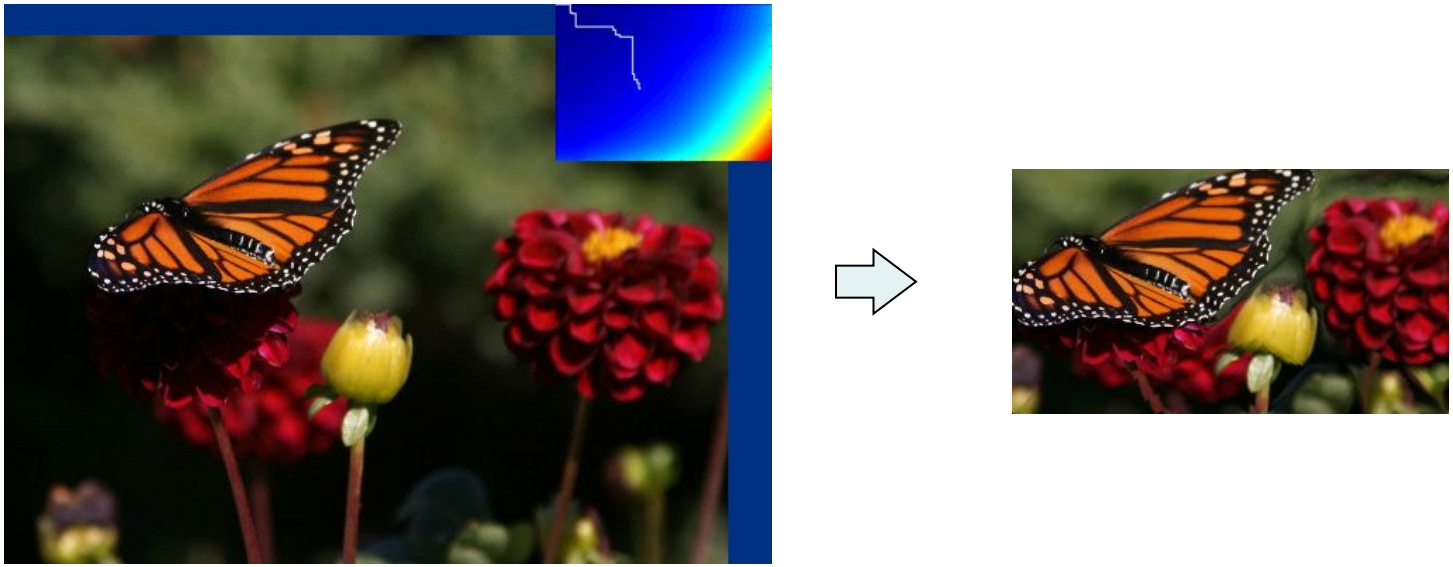


Image Expansion (Synthesis)

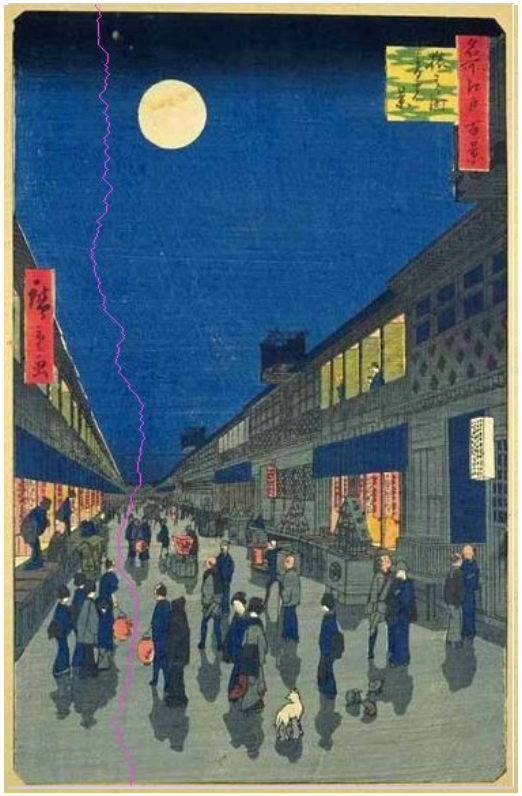
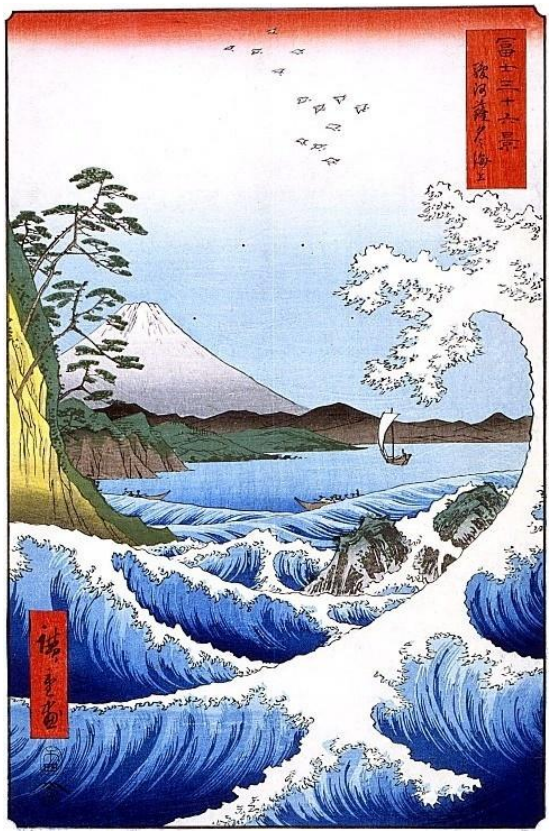


Image Expansion – take 2

Scaling

Enlarged or Reduced?



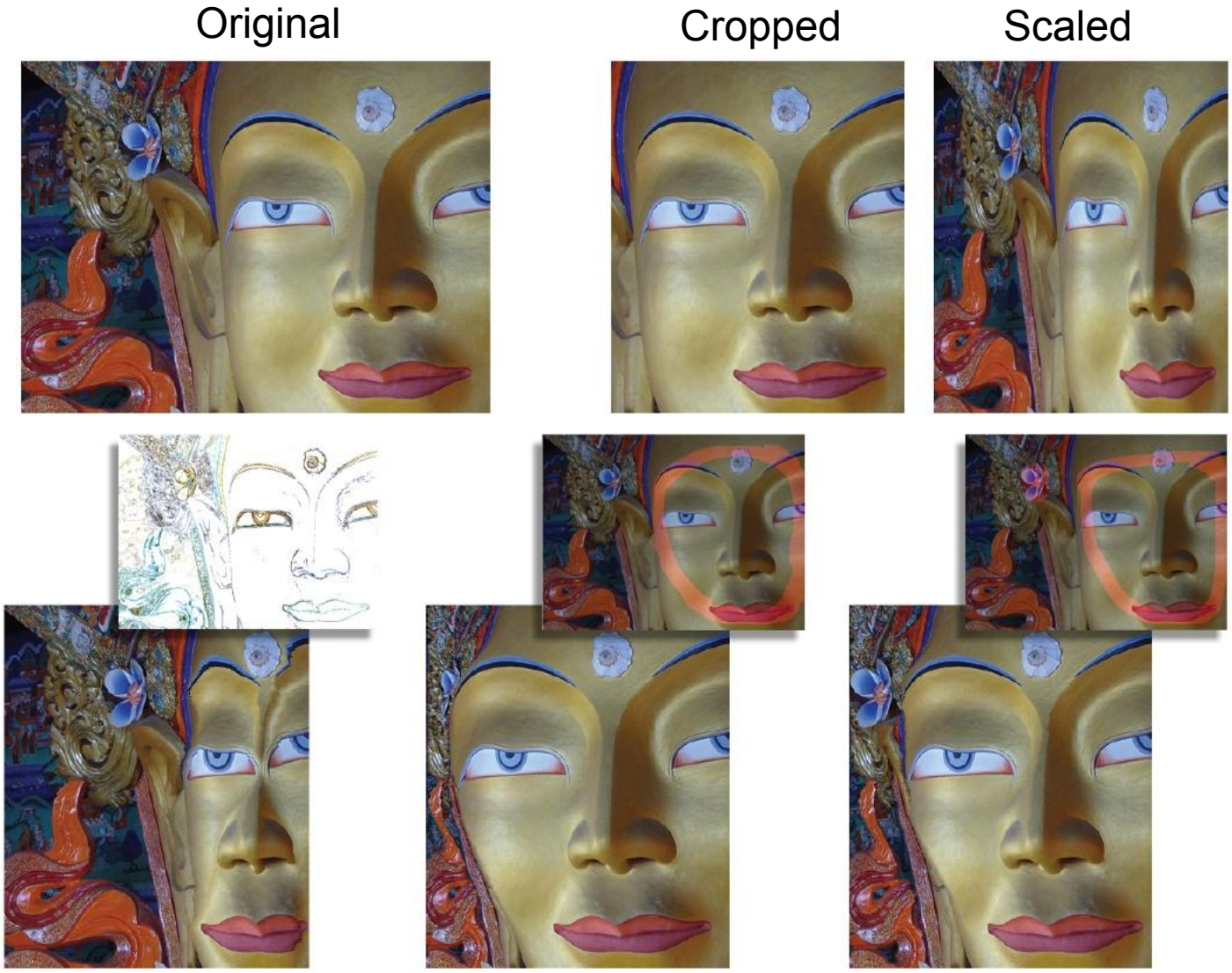
Combined Insert and Remove

Insert & remove seams

Scaling

With face detector

With User Constraints



Object Removal

Object Removal

Input | Retargeted | Pigeon Removed | Girl Removed

Find the Missing Shoe!

Michael Rubinstein — MIT CSAIL – mrub@mit.edu

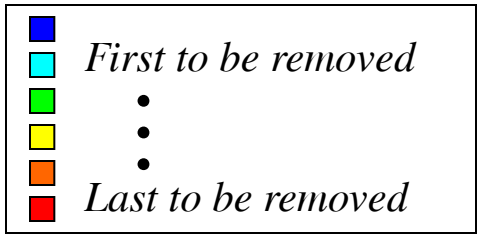
Solution

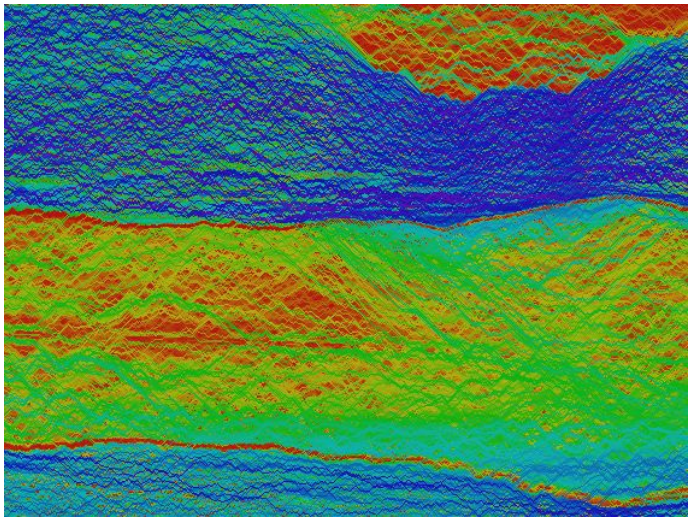
Multi-Size Images

- **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

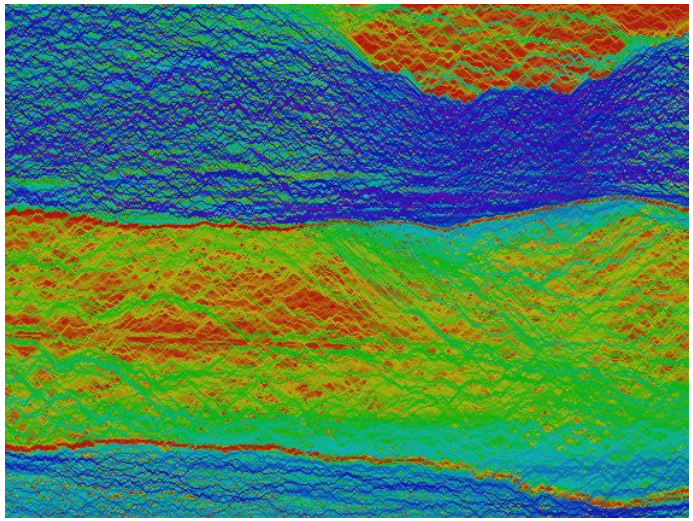
Multi-Size Images





Multi-Size Image Representation

+



Limitations

Content

Structure