Computational Photography

Week 4, Spring 2014

Instructor: Lou Kratz

Last Week: HDR

Camera is not a photometer!

What does the eye see?

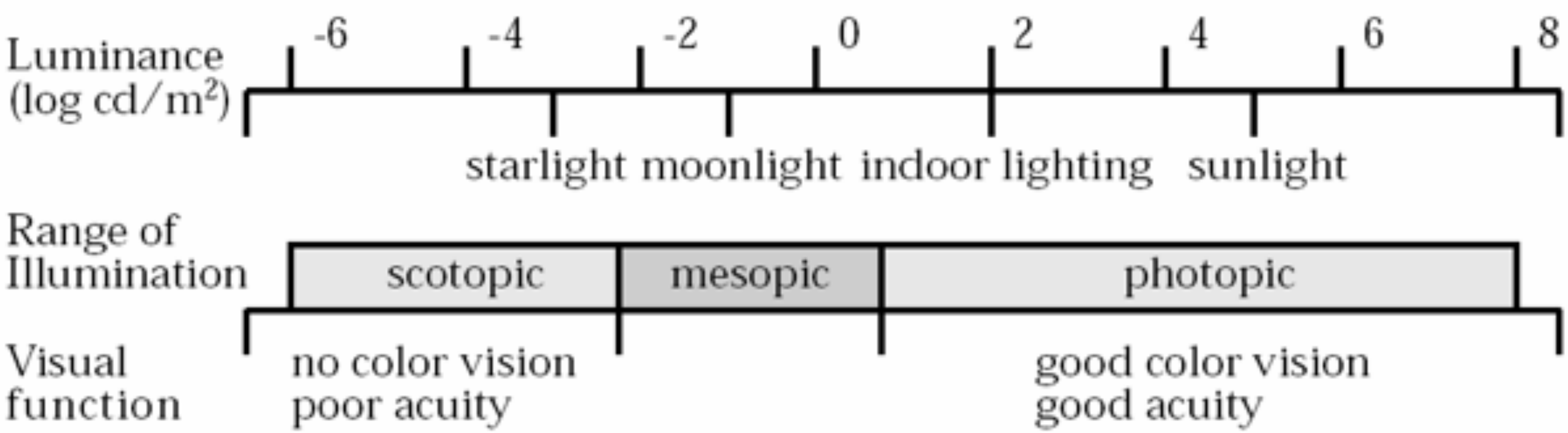


Figure 1: The range of luminances in the natural environment and associated visual parameters. After Hood (1986).

The eye has a huge dynamic range

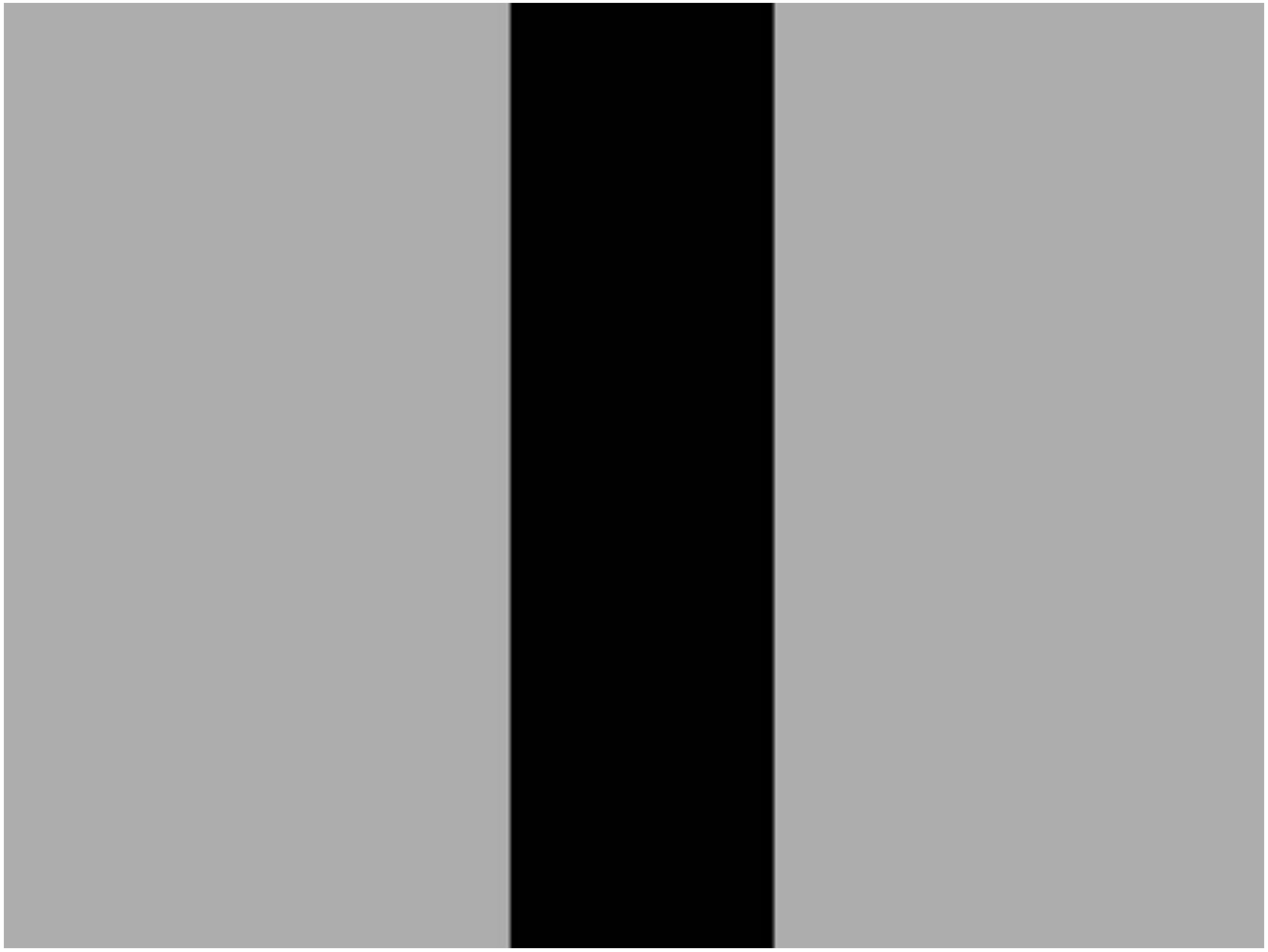
Do we see a true radiance map?

Eye is not a photometer!

"*Every light is a shade, compared to the higher lights, till you come to the sun; and every shade is a light, compared to the deeper shades, till you come to the night*."

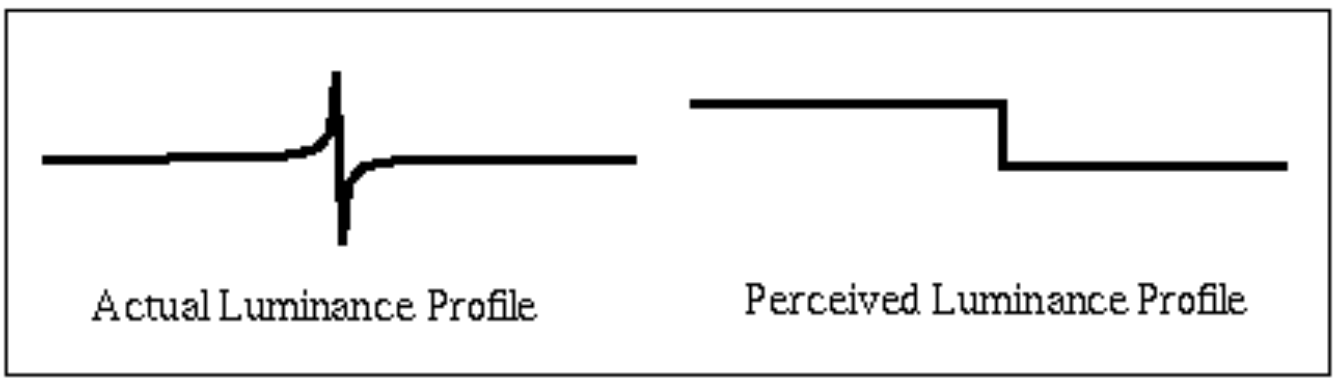
— John Ruskin, 1879

Cornsweet Illusion



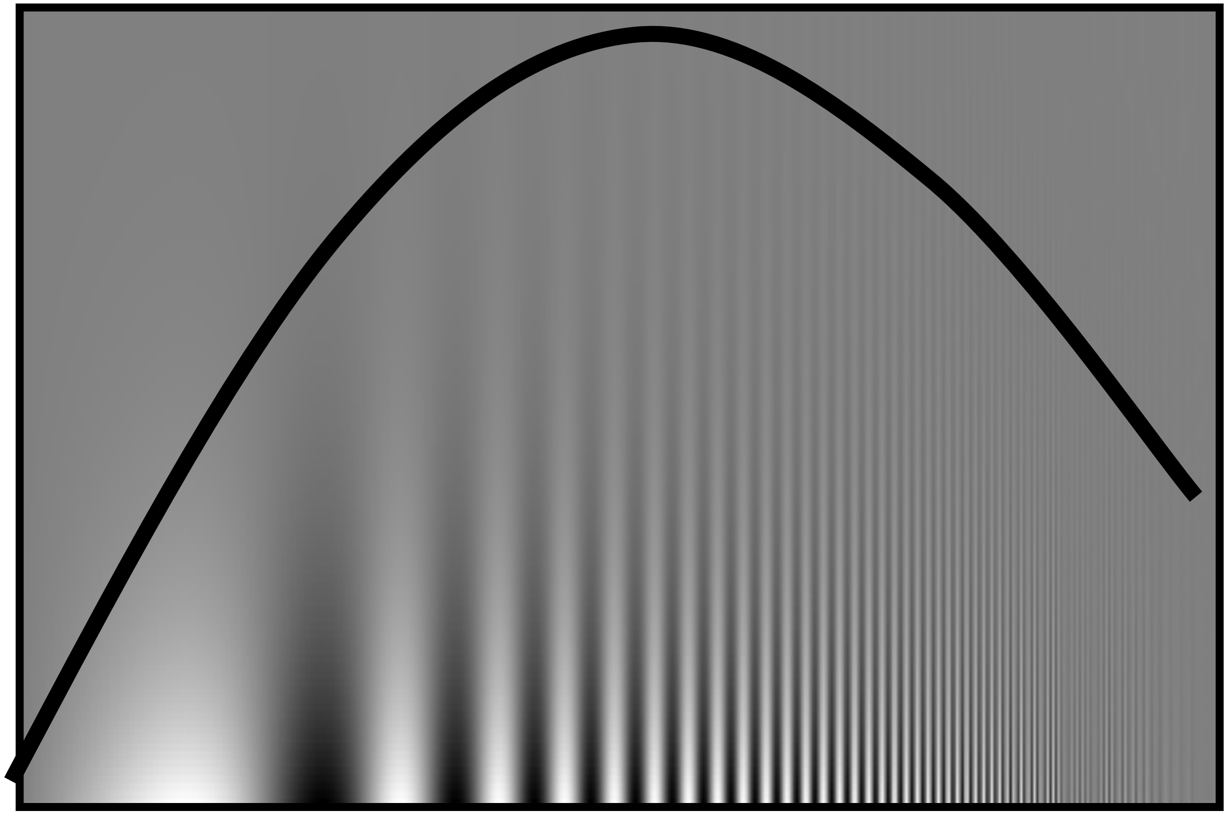
Metameric Failure

Craik-O'Brien Cornsweet Effect



Eye is sensitive to changes

Sine Wave



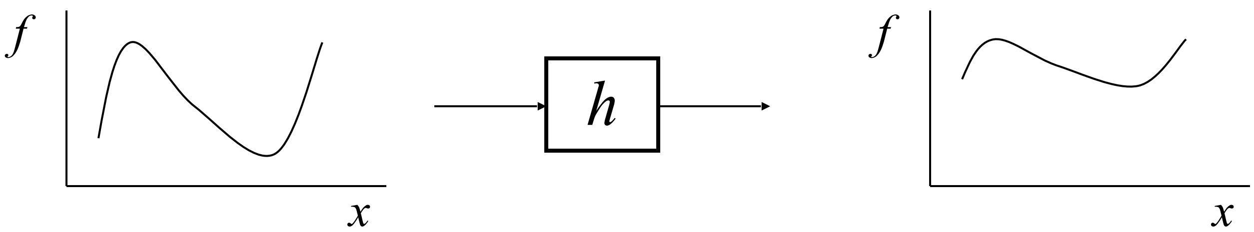
Campbell-Robson contrast sensitivity curve

Point Processing

Recall Image Processing...

- image filtering: change ***range*** of image

$$g(x) = h(f(x))$$



- image warping: change ***domain*** of image

$$g(x) = f(h(x))$$

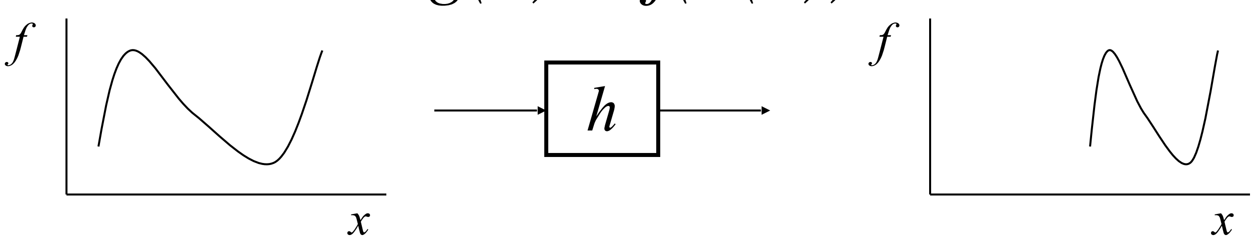
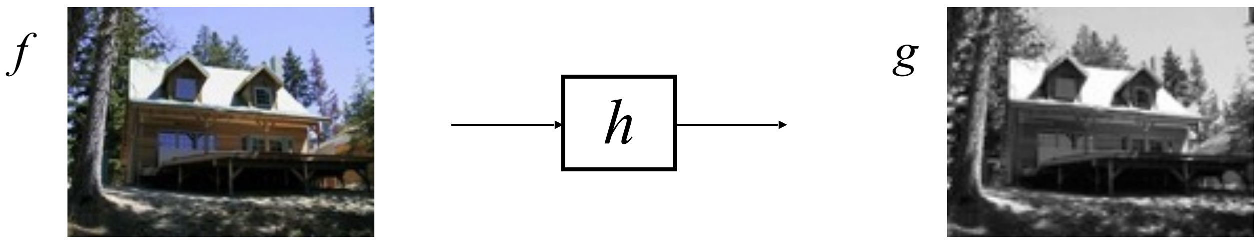


Image Processing

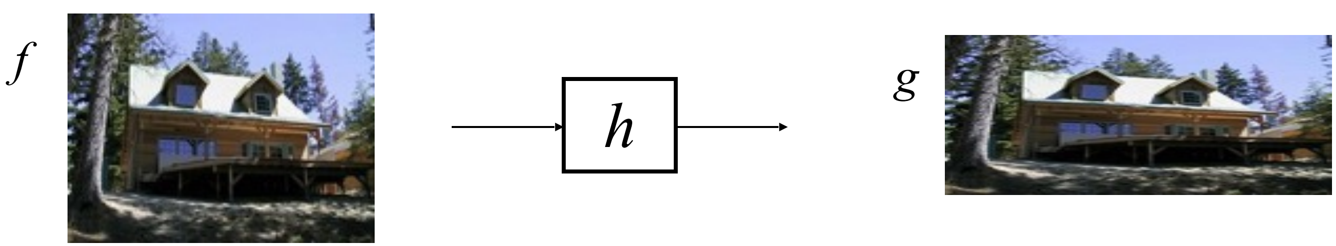
- image filtering: change ***range*** of image

$$g(x) = h(f(x))$$

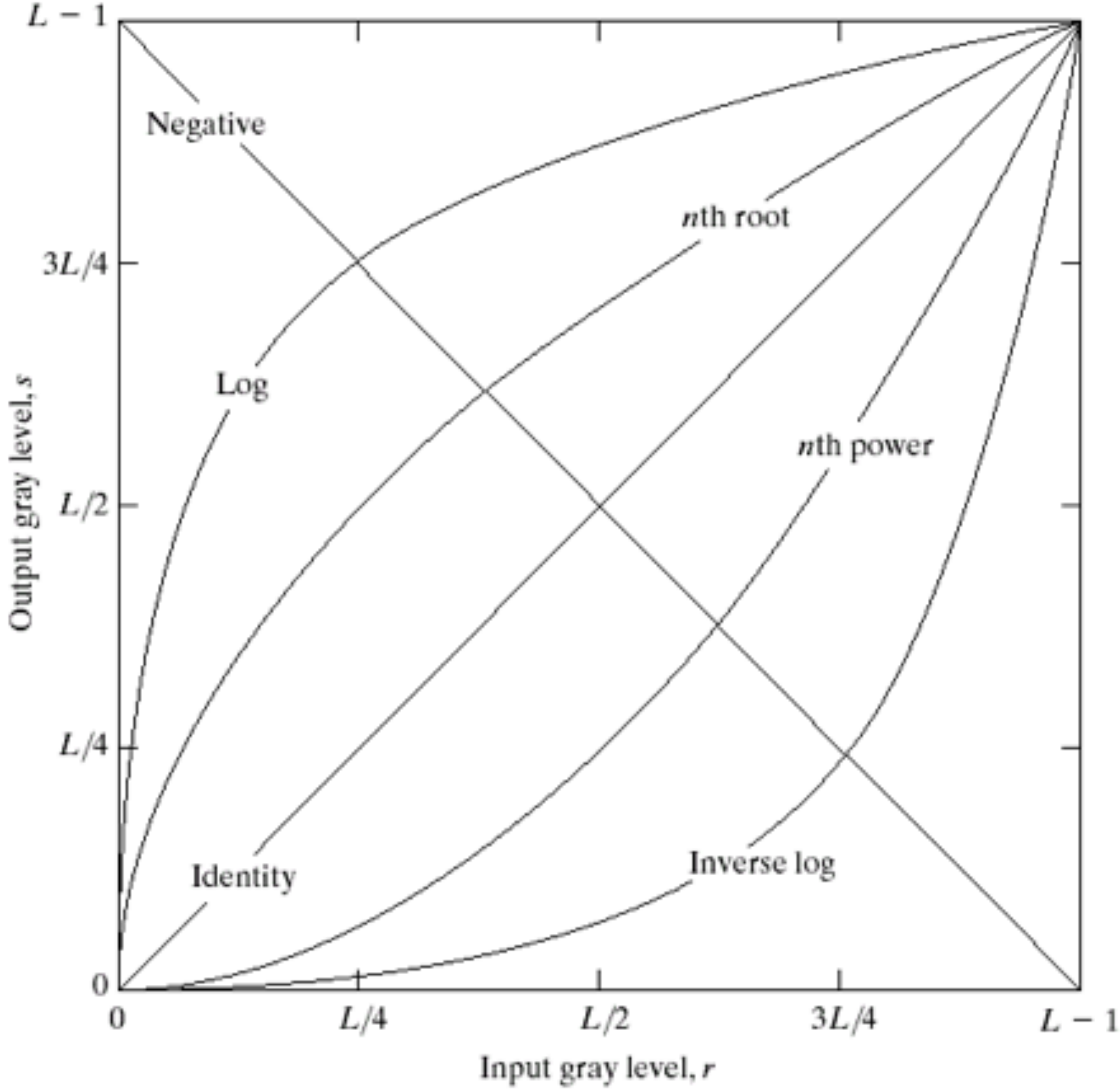


- image warping: change ***domain*** of image

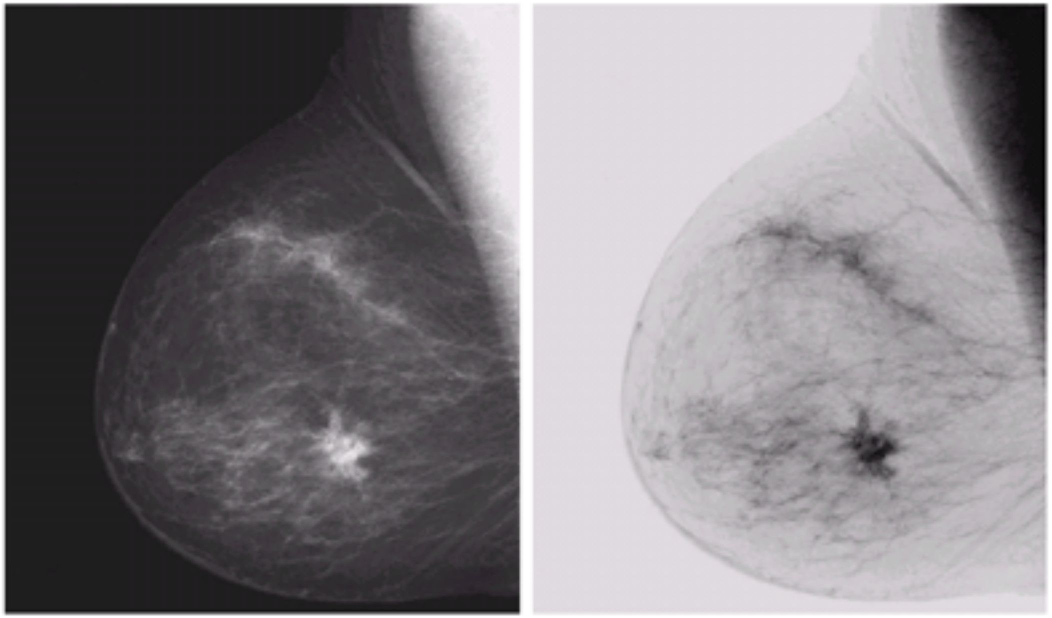
$$g(x) = f(h(x))$$



Basic Point Processing



Negative



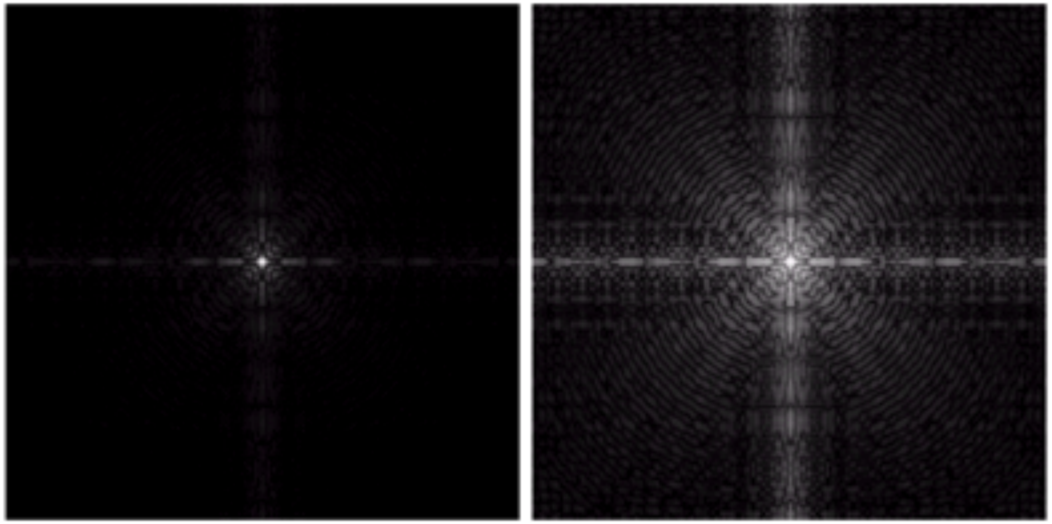
a b

FIGURE 3.4
(a) Original digital mammogram.
(b) Negative image obtained using the negative transformation in Eq. (3.2-1).
(Courtesy of G.E. Medical Systems.)

Log

a b

FIGURE 3.5
(a) Fourier spectrum.
(b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.



Power-law Transformations

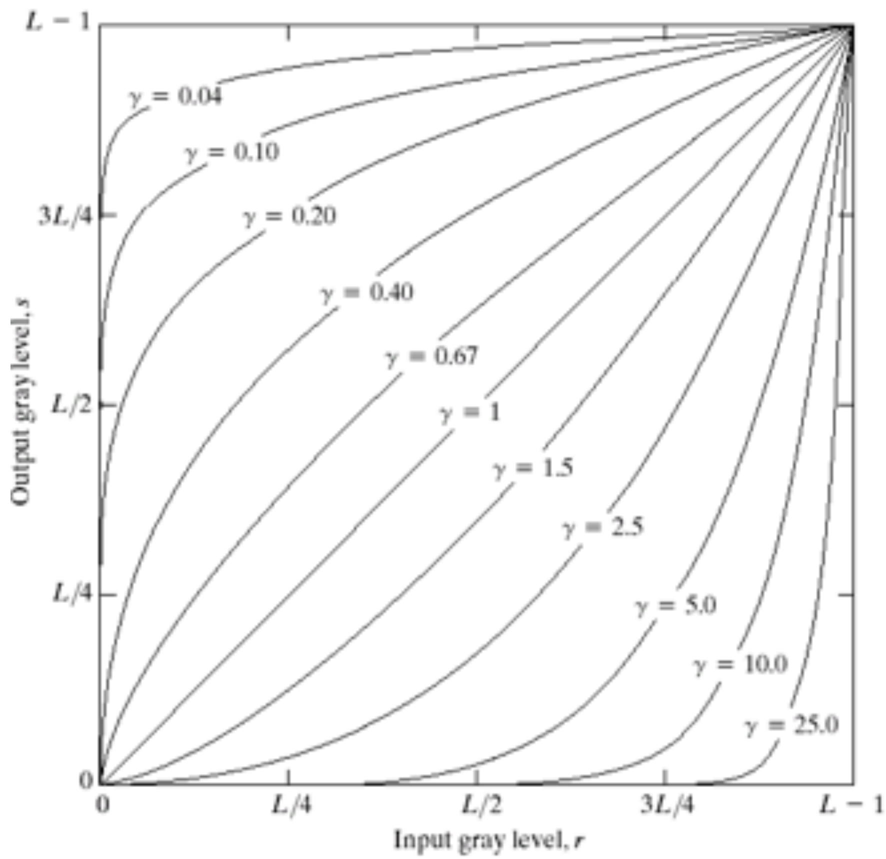
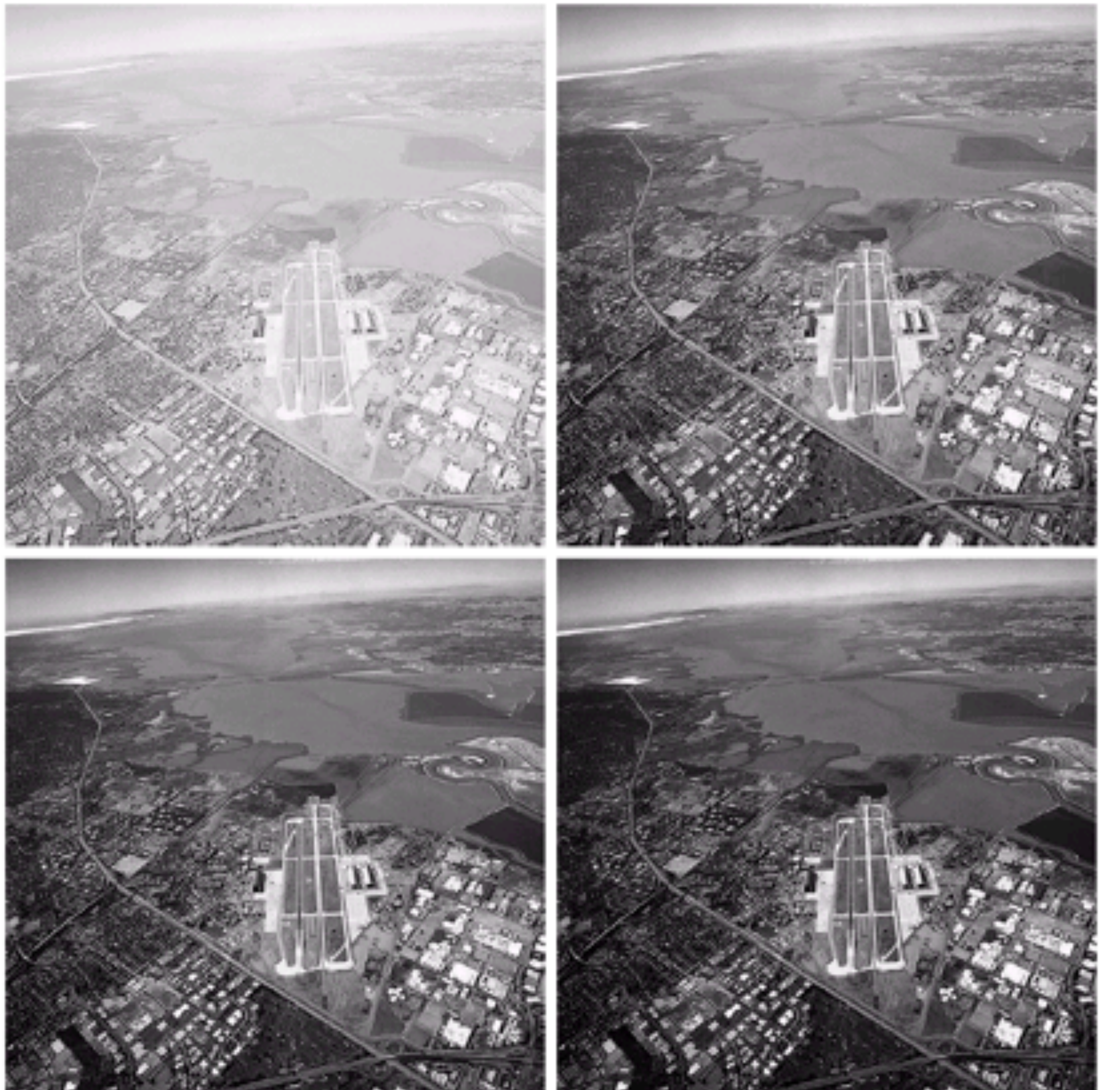


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

$$s = cr^{\gamma}$$

Image Enhancement



$\gamma = 3.0$

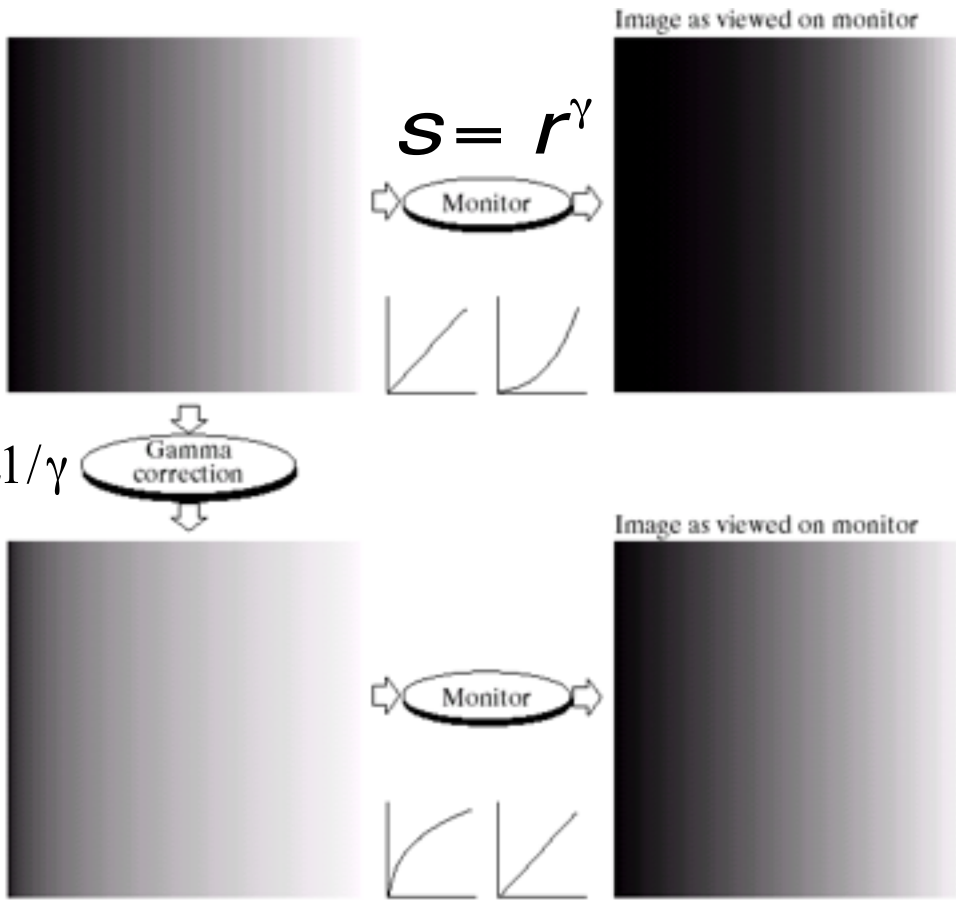
$\gamma = 4.0$

$\gamma = 5.0$

Example: Gamma Correction

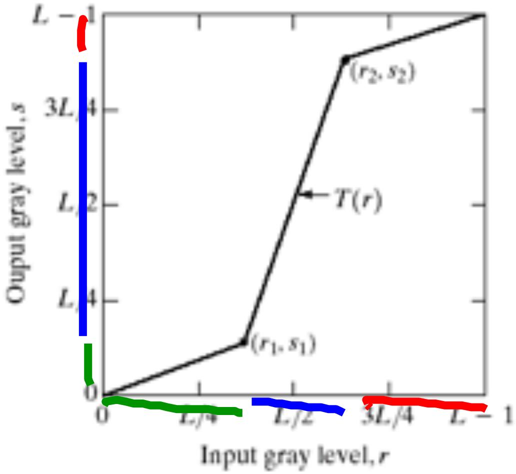
$$s = r^{\gamma}$$

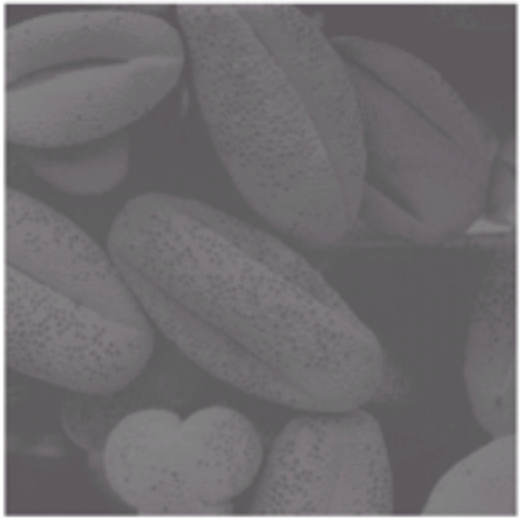
$$s = r^{1/\gamma}$$



$$e.g. \quad 0.25 = 0.5^{2.0}$$

Contrast Stretching





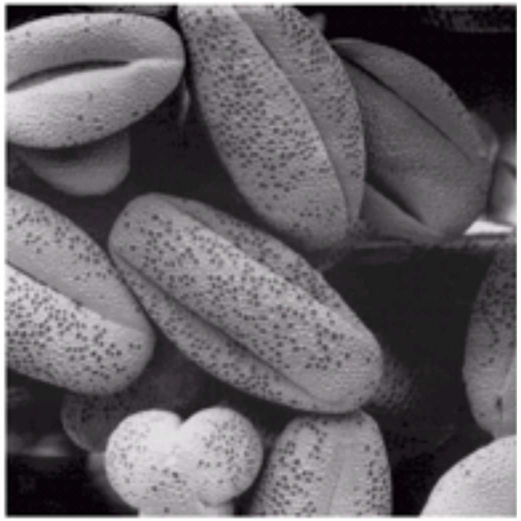
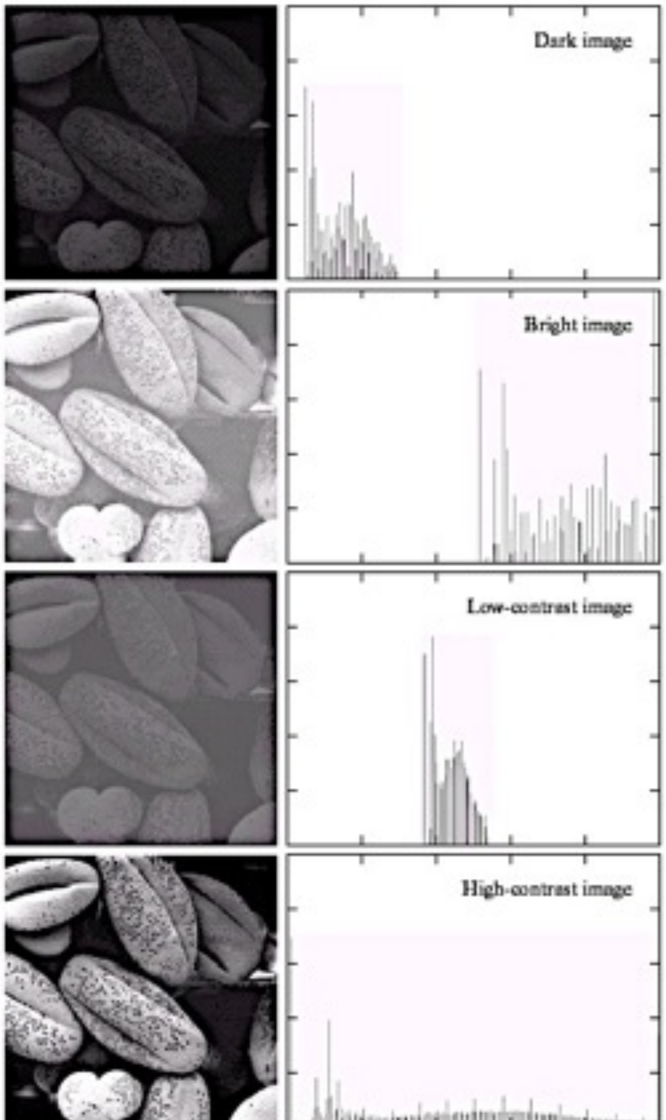
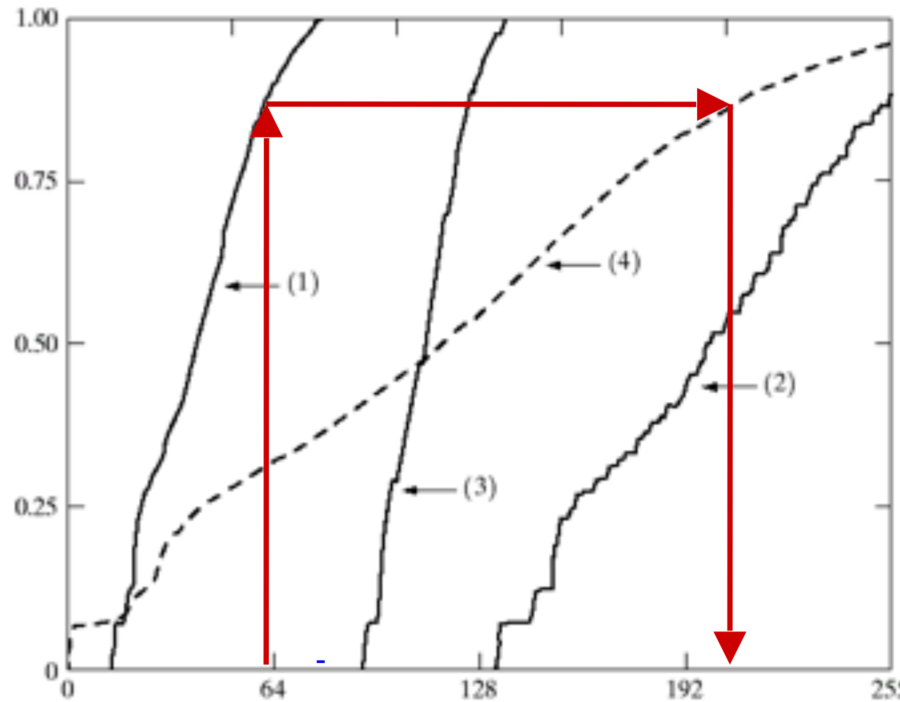


Image Histograms

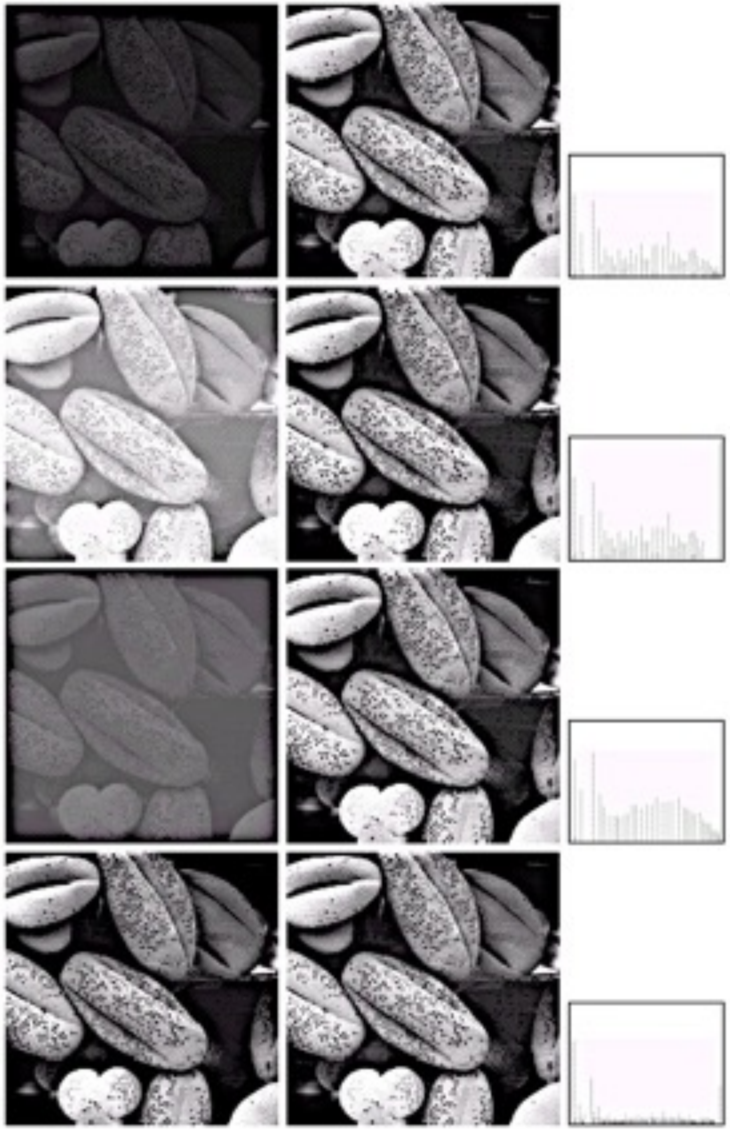




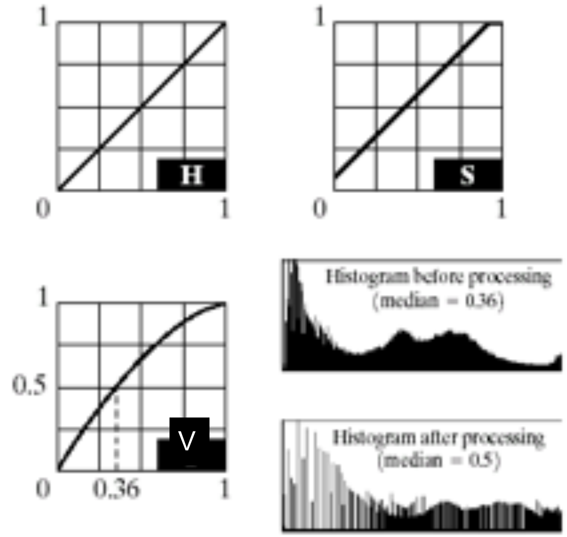
Cumulative Histograms

$s = T(r)$

Histogram Equalization



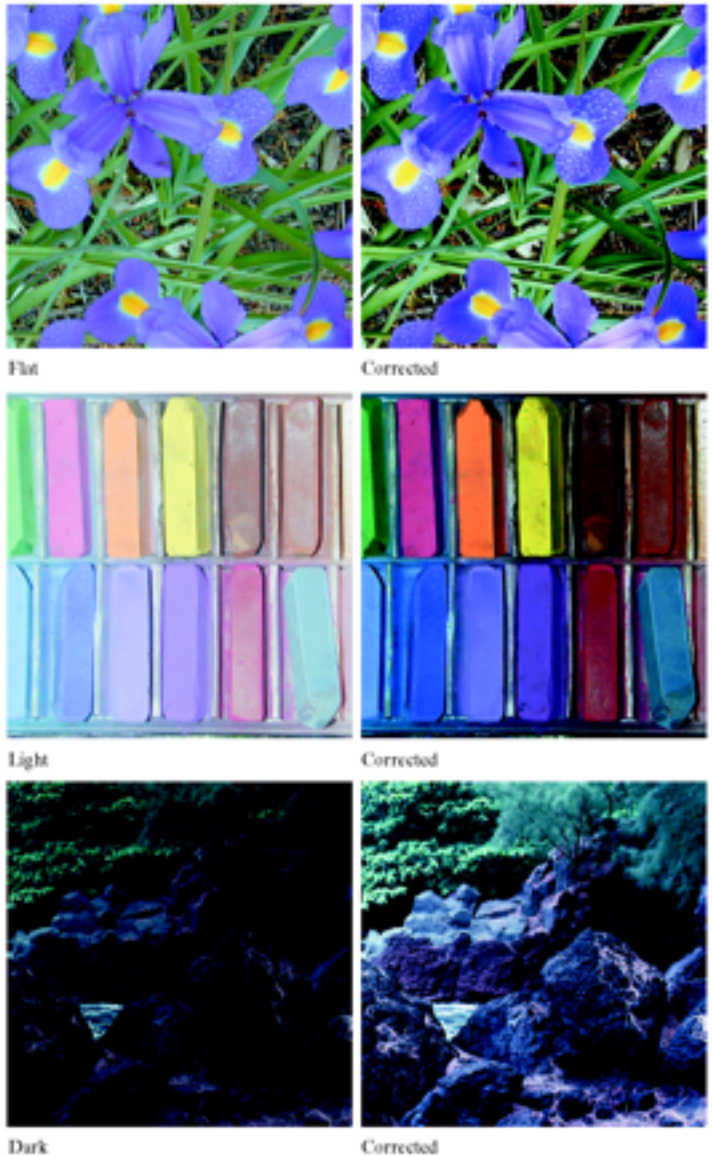
Histogram Equalization



Histogram equalize
the brightness (V)

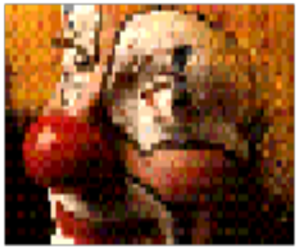
Color Channels

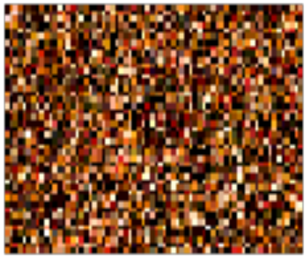
Color Point Processing



Neighborhood Processing (Filtering)

- Q: What happens if I reshuffle all pixels within the image?



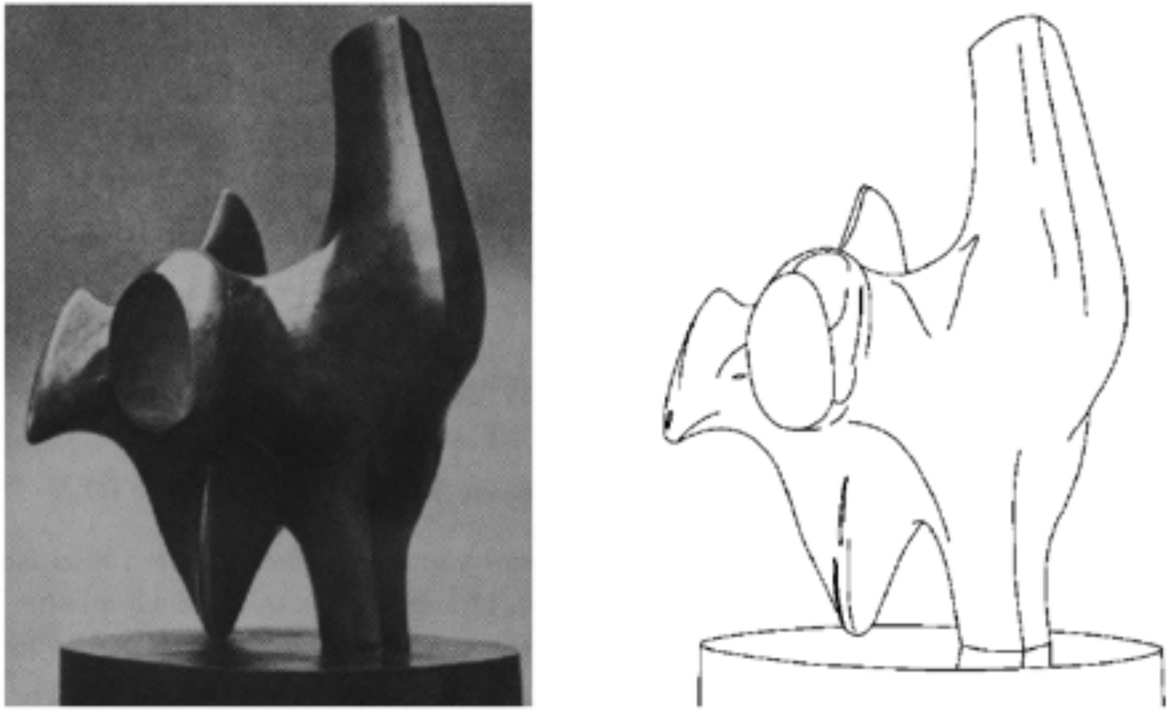


- A: It's histogram won't change. No point processing will be affected…

- Need spatial information to capture this

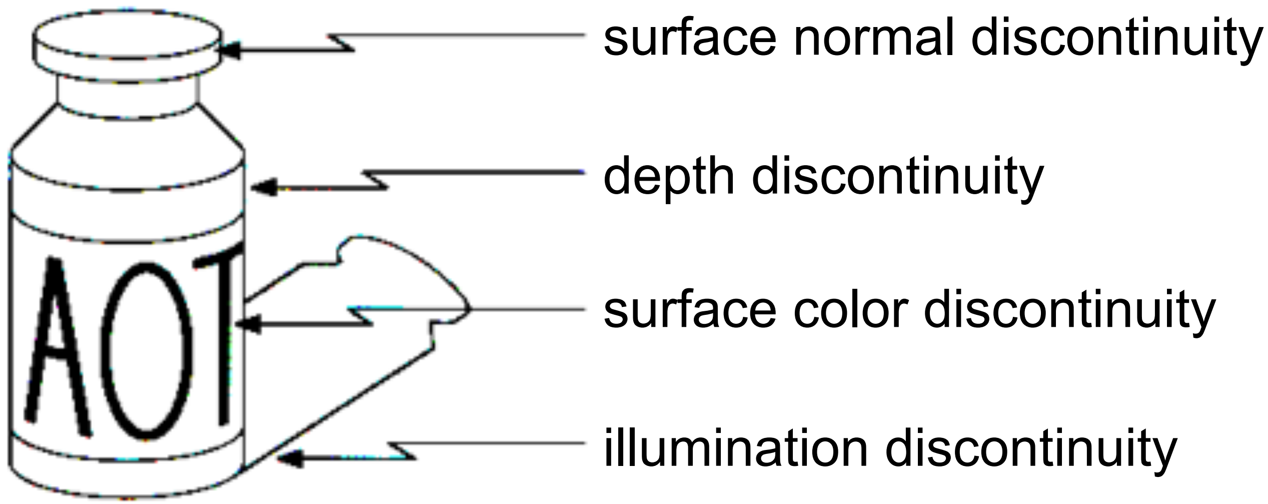
Edge Detection

Edge Detection



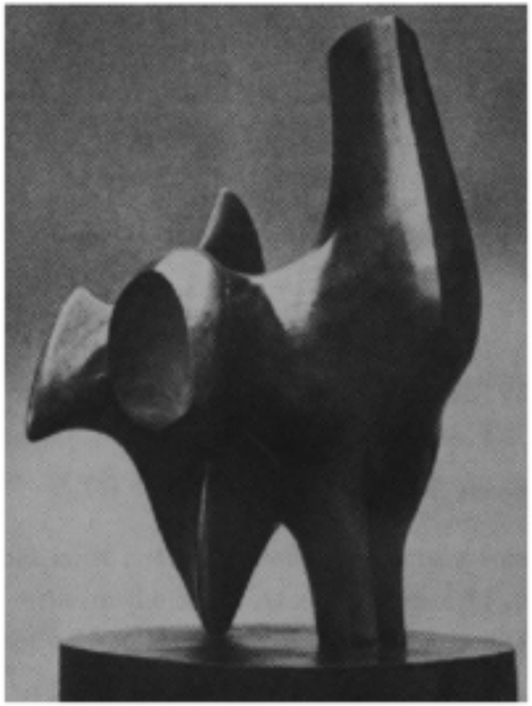
- Convert a 2D image into a set of curves
  - Extracts salient features of the scene
  - More compact than pixels

Origin of Edges



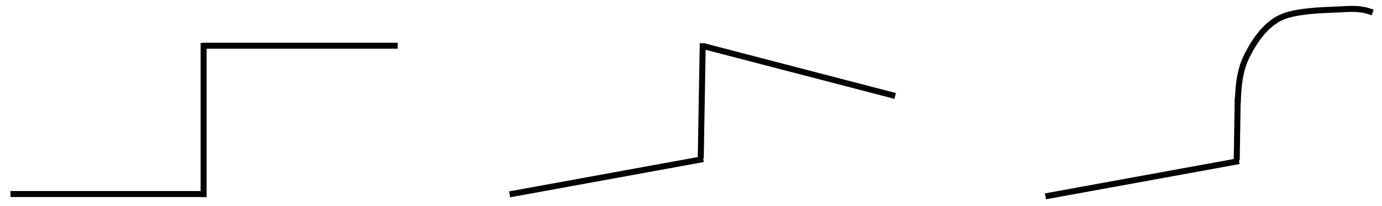
- Edges are caused by a variety of factors

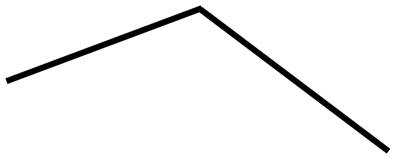
Edge Detection



- How can you tell that a pixel is on an edge?

Edge Types





Roof Edge

Line Edges

Discrete Edge Operators

How can we differentiate a ***digital*** image?

Finite difference approximations:

Gradients:

$$\frac{\partial I}{\partial x} \approx \frac{1}{2\varepsilon}\left(\left(I_{i+1,j+1} - I_{i,j+1}\right) + \left(I_{i+1,j} - I_{i,j}\right)\right)$$

$$\frac{\partial I}{\partial y} \approx \frac{1}{2\varepsilon}\left(\left(I_{i+1,j+1} - I_{i+1,j}\right) + \left(I_{i,j+1} - I_{i,j}\right)\right)$$

$$\begin{array}{|c|c|}\hline I_{i,j+1} & I_{i+1,j+1} \\ \hline I_{i,j} & I_{i+1,j} \\ \hline \end{array} \;\; \varepsilon$$

Cross-correlation masks :

$$\frac{\partial I}{\partial x} \approx \frac{1}{2\varepsilon} \begin{array}{|c|c|}\hline -1 & 1 \\ \hline -1 & 1 \\ \hline \end{array} \qquad \frac{\partial I}{\partial y} \approx \frac{1}{2\varepsilon} \begin{array}{|c|c|}\hline 1 & 1 \\ \hline -1 & -1 \\ \hline \end{array}$$

The Sobel Operator

- Better approximations of the gradients exist
  - The *Sobel* operators below are very commonly used

$$\frac{\partial}{\partial x} = \frac{1}{8}\begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \qquad \frac{\partial}{\partial y} = \frac{1}{8}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}$$

  - The standard defn. of the Sobel operator omits the 1/8 term
    - doesn’t make a difference for edge detection
    - the 1/8 term **is** needed to get the right gradient value, however

Discrete Edge Operators

2nd order partial derivatives:

$$\frac{\partial^2 I}{\partial x^2} \approx \frac{1}{\varepsilon^2}\left(I_{i-1,j} - 2I_{i,j} + I_{i+1,j}\right)$$

$$\frac{\partial^2 I}{\partial y^2} \approx \frac{1}{\varepsilon^2}\left(I_{i,j-1} - 2I_{i,j} + I_{i,j+1}\right)$$

$I_{i-1,j+1}$	$I_{i,j+1}$	$I_{i+1,j+1}$
$I_{i-1,j}$	$I_{i,j}$	$I_{i+1,j}$
$I_{i-1,j-1}$	$I_{i,j-1}$	$I_{i+1,j-1}$

Laplacian :

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

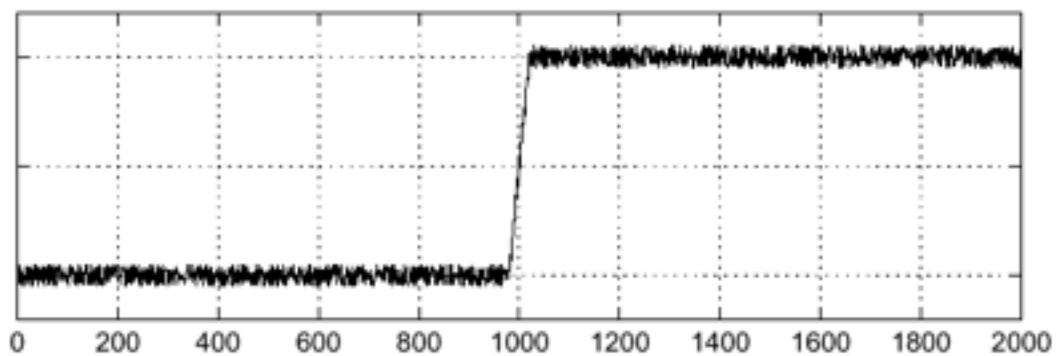
Cross-correlation masks :

$$\nabla^2 I \approx \frac{1}{\varepsilon^2}\begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline\end{array} \quad \text{or} \quad \frac{1}{6\varepsilon^2}\begin{array}{|c|c|c|}\hline 1 & 4 & 1 \\ \hline 4 & -20 & 4 \\ \hline 1 & 4 & 1 \\ \hline\end{array}$$

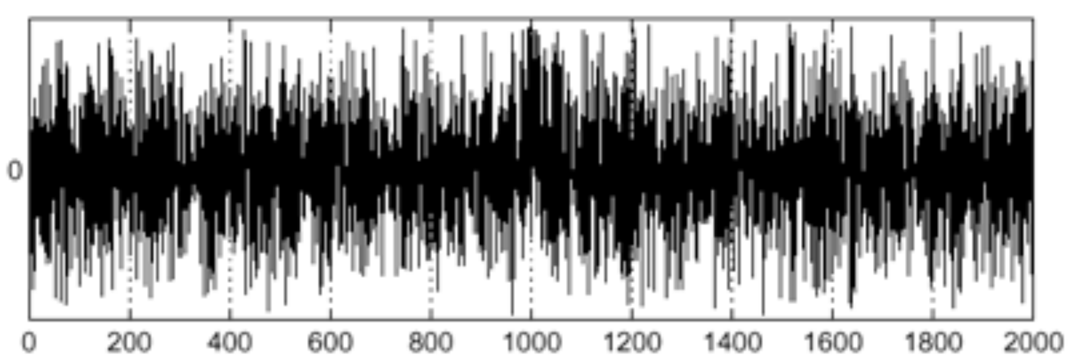
Effects of Noise

- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal

$f(x)$

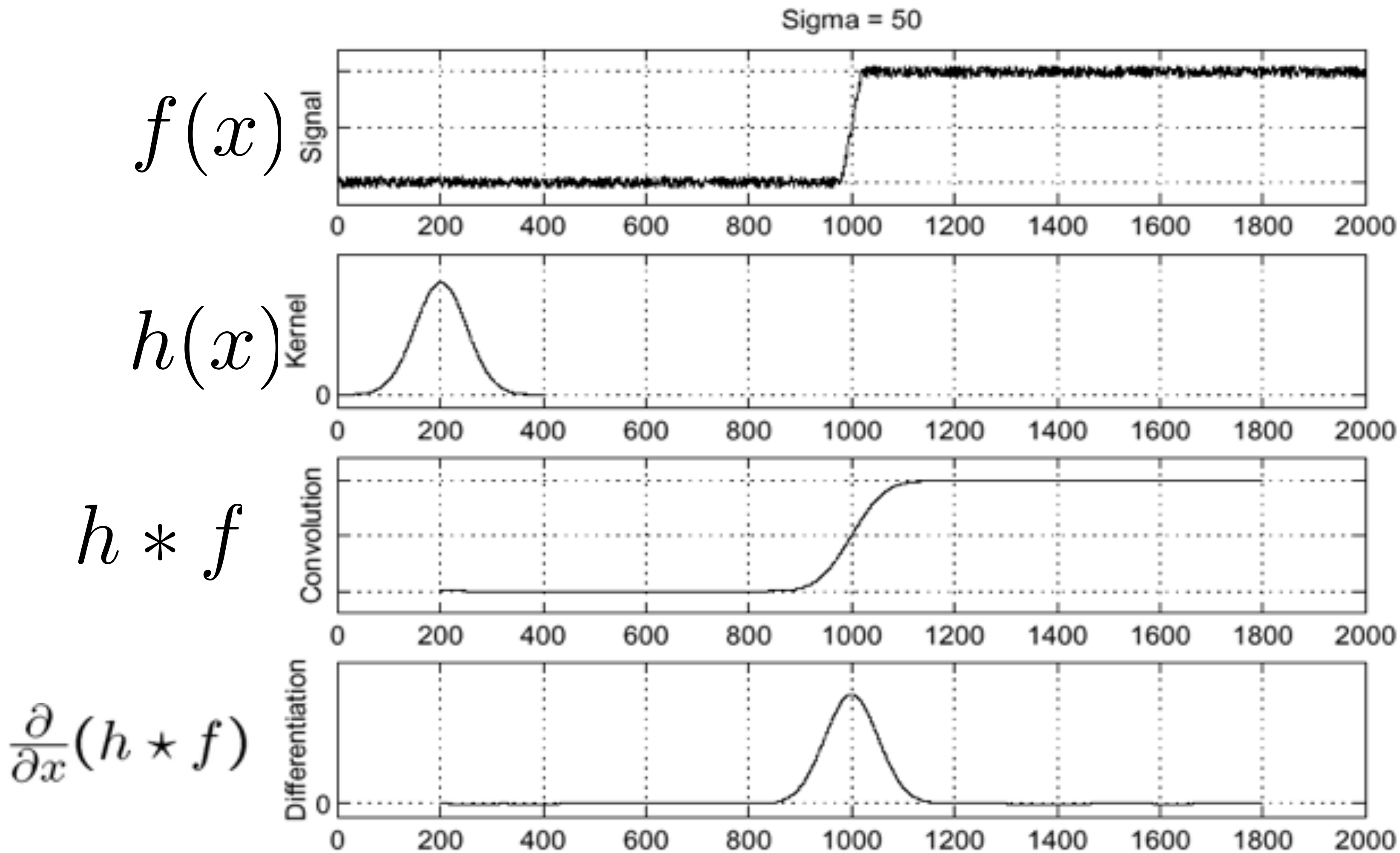


$\frac{d}{dx}f(x)$



Where is the edge?

Solution: Smooth First



Where is the edge?   Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Derivative Theorem of Convolution

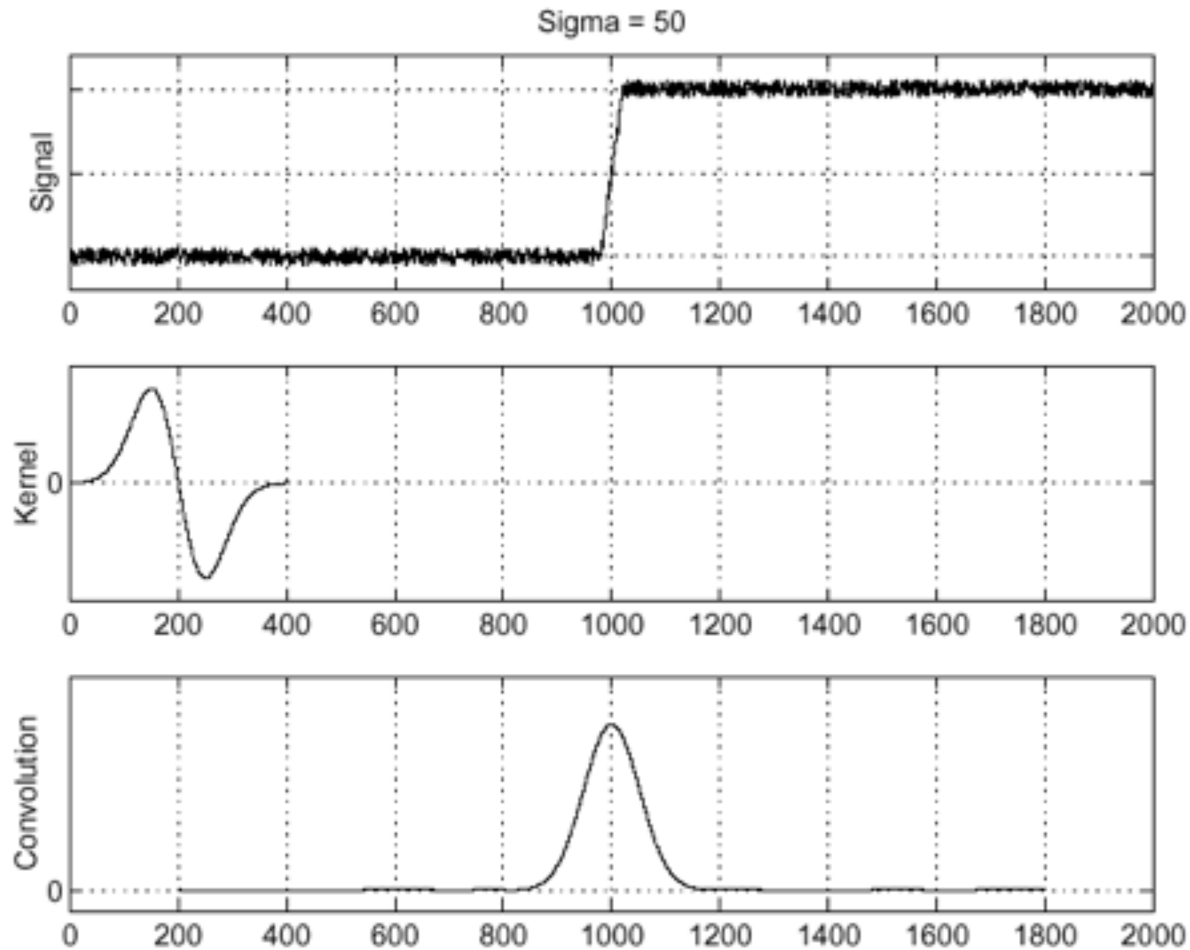
$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

This saves us one operation:

$$f(x)$$

$$\frac{\partial}{\partial x}h$$

$$(\frac{\partial}{\partial x}h) \star f$$



Laplacian of Gaussian

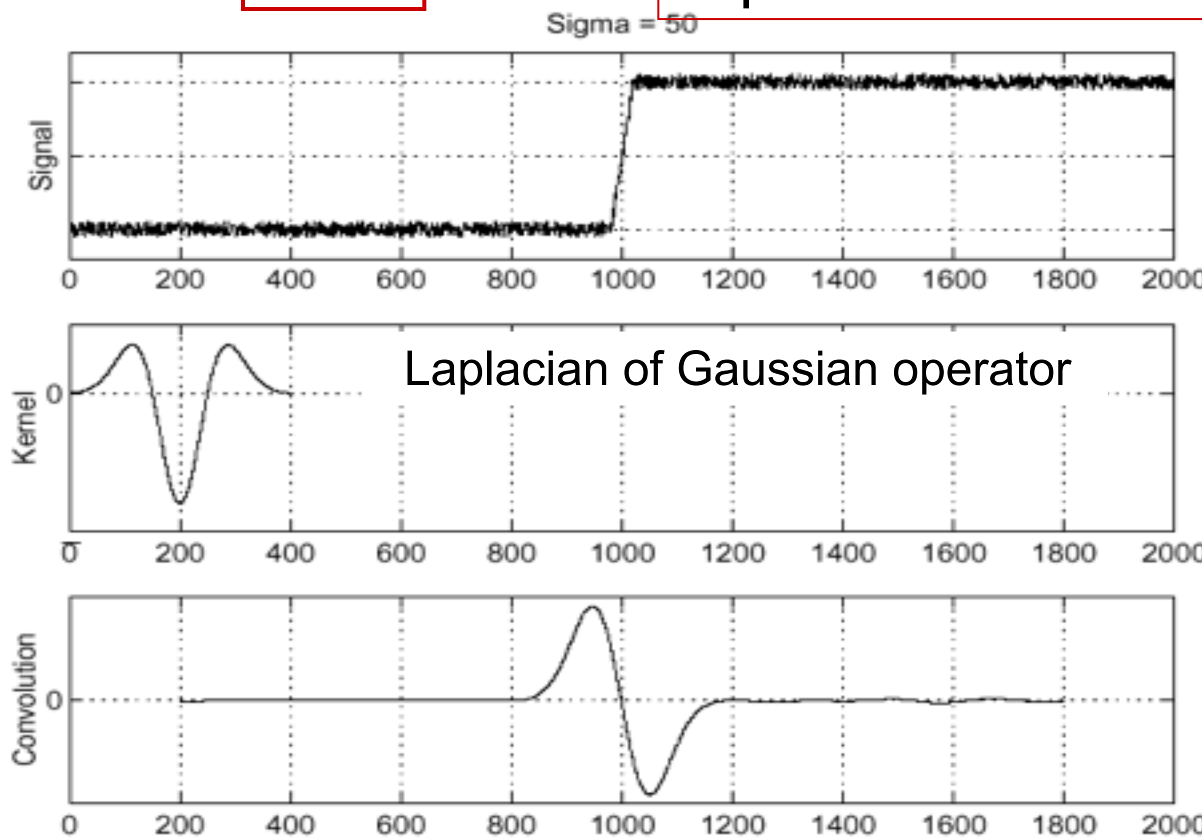
$$\frac{\partial^2}{\partial x^2}(h*f)=\boxed{\left(\frac{\partial^2}{\partial x^2}h\right)}*f$$

Laplacian of Gaussian

$$f(x)$$

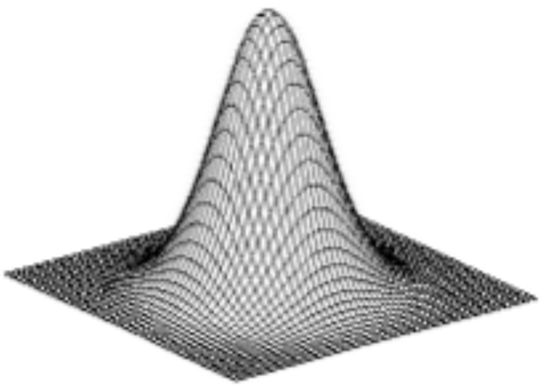
$$\frac{\partial^2}{\partial x^2}h$$

$$\left(\frac{\partial^2}{\partial x^2}h\right)\star f$$



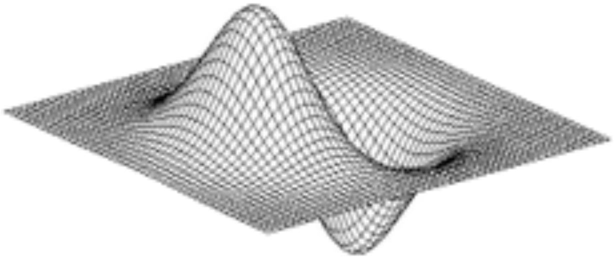
Where is the edge? Zero-crossings of bottom graph !

2D Gaussian Edge Operators



Gaussian

$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$

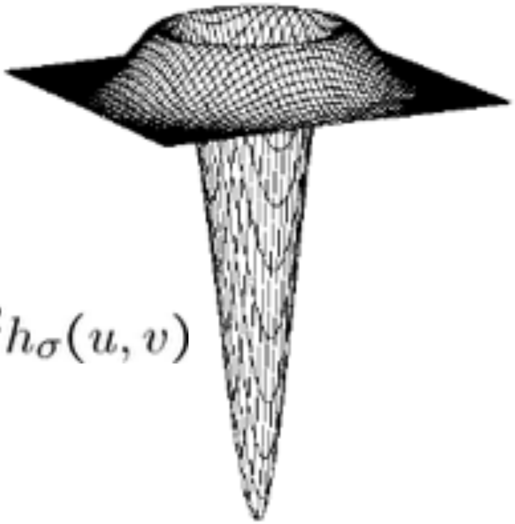


derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

Laplacian of Gaussian

Mexican Hat (Sombrero)



$$\nabla^2 h_\sigma(u,v)$$

Canny Edge Operator

original

Canny with $\sigma = 1$

Canny with $\sigma = 2$

- The choice of σ depends on desired behavior
  - large σ detects large scale edges
  - small σ detects fine features

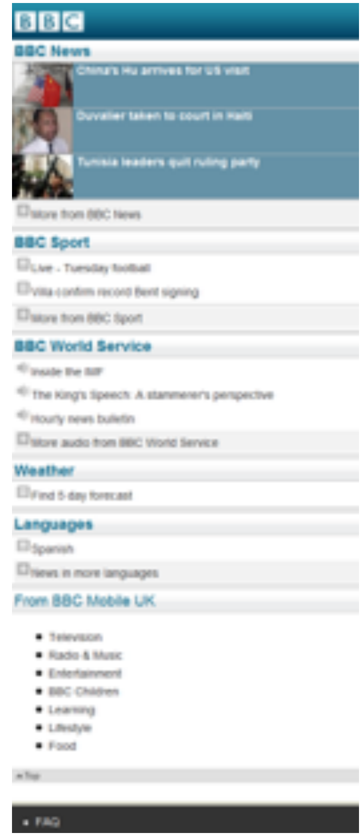
Image Retargeting

Slides by Michael Rubenstein, Ariel Shamir, and Shai Avidan

Display Devices

Content Retargeting

PC



iPhone

Page Layout

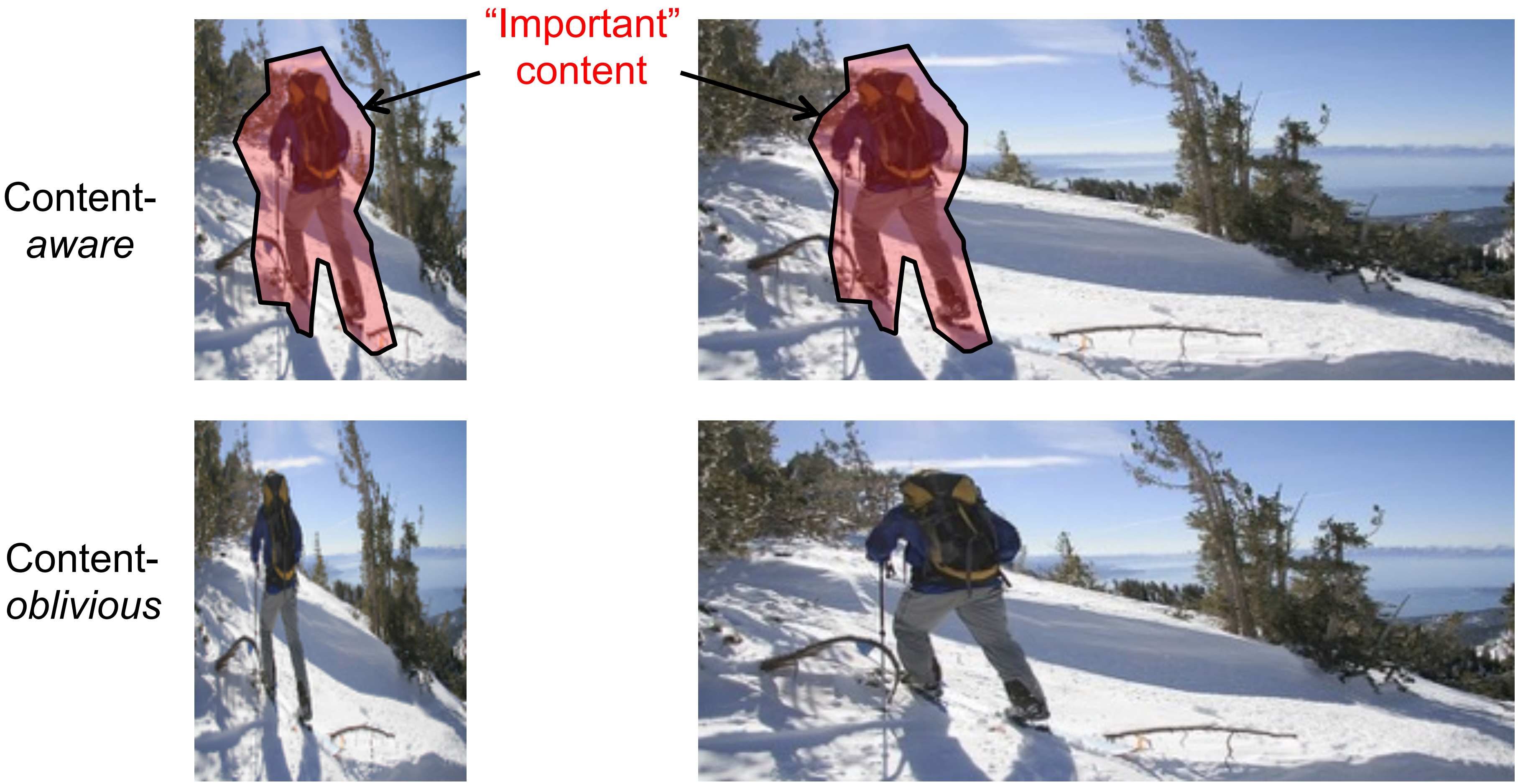
Page layout - Wikipedia, the free encyclopedia - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://en.wikipedia.org/wiki/Page_layout
Page Layout
Page layout - Wikipedia, the ...
Log in / create account
Article Discussion
Read Edit View history
Search
Wikipedia
The Free Encyclopedia
Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Toolbox
Print/export
Languages
Česky
Dansk
Deutsch
Esperanto
Français
Italiano
Nederlands
Polski
Português
Suomi
Page layout
From Wikipedia, the free encyclopedia
For the Wikipedia guideline on an articles' layout, see Wikipedia:Manual of Style (layout).
This article does not cite any references or sources.
Page layout is the part of graphic design that deals in the arrangement and style treatment of elements (content) on a page.
Contents [hide]
1 History and development
2 Grids versus templates
3 Front-end versus back-end
4 See also
History and development
[edit]
Beginning from early illuminated pages in hand-copied books of the Middle Ages and proceeding down to intricate modern magazine and catalog layouts, proper page design has long been a consideration in printed material. With print media, elements usually consist of type (text), images (pictures), and occasionally place-holder graphics for elements that are not printed with ink such as die/laser cutting, foil stamping or blind embossing.
Consumer magazine sponsored advertisements and covers rely heavily on professional page layout skills to compete for visual attention.
Since the advent of personal computing, page layout skills have expanded to electronic media as well as print media. The electronic page is better known as a graphical user interface (GUI) when interactive elements are included. Page layout for interactive media overlaps with (and is often called) interface design. This usually includes interactive elements and multimedia in addition to text and still images. Interactivity takes page layout skills from planning attraction and eye flow to the next level of planning user experience in collaboration with software engineers and creative directors.[citation needed]
A page layout may be designed in a rough paper and pencil sketch before producing, or produced during the design process to the final form. Both design and production may be achieved using hand tools or page layout software. Producing a web page may require knowledge of markup languages along with WYSIWYG editors to compensate for incompatibility between platforms. Special considerations must be made for how the layout of an HTML page will change (reflow) when resized by the end-user. Cascading style sheets are often required to keep the page layout consistent between web browsers.
Grids versus templates
[edit]
Grids and templates are page layout design patterns used in advertising campaigns and multiple-page publications, including websites.
Done

Simple Media Retargeting Operators

Scaling

Letterboxing

Content-aware Retargeting Operators



Content-aware Retargeting

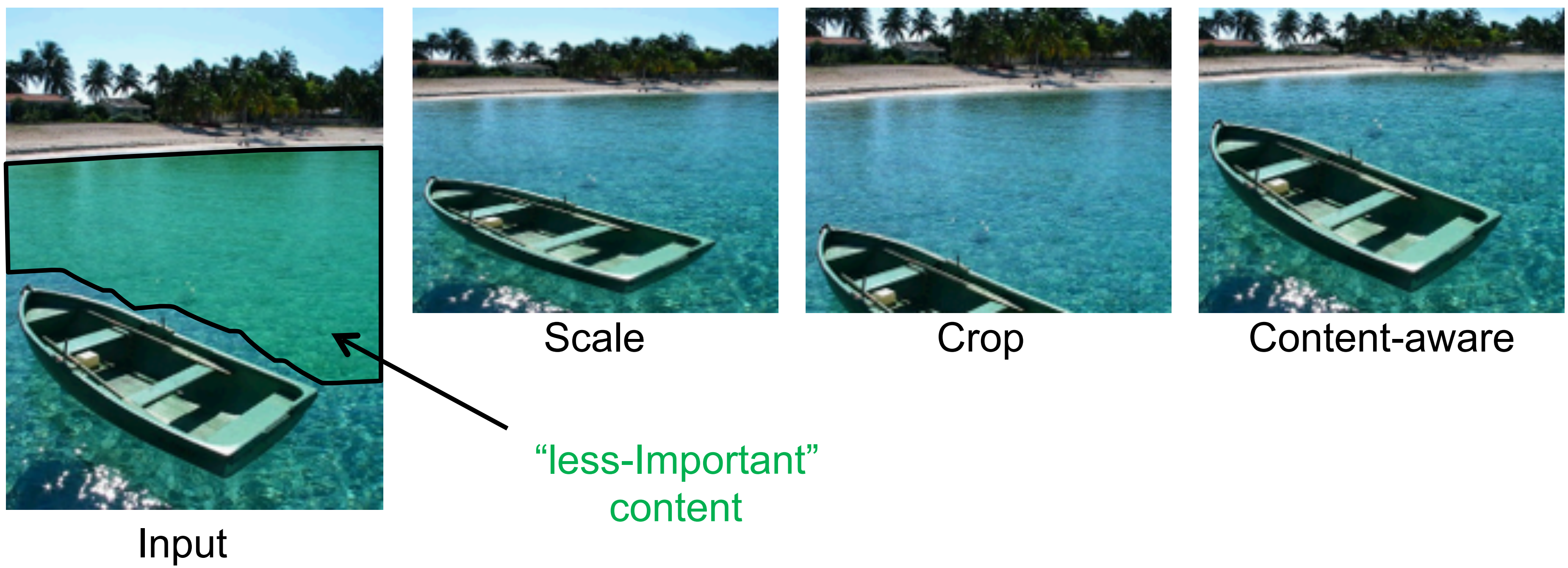


Image Retargeting

This is indexing the image as a matrix!!

m columns, n rows
width m, height n

- **Problem statement:**
  - Input Image nxm, and new size n'xm'
  - Output Image of size n'xm' which will be "good representative" of the original image

- **To date, no agreed definition, or measure, as to what a good representative is in this context!**

Image/Video Retargeting

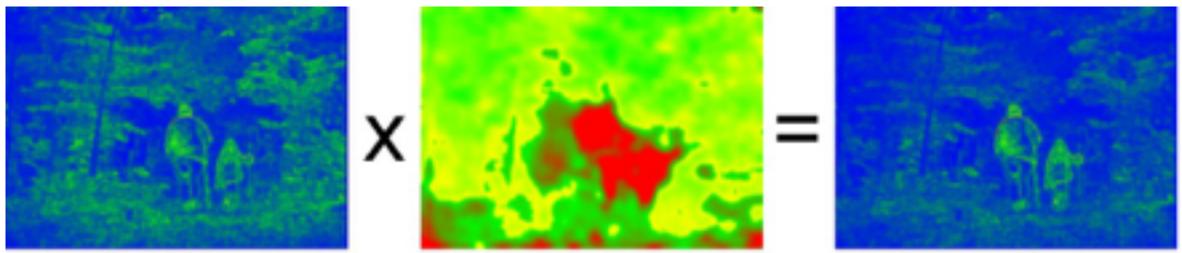
- **In large, we would expect:**

1. Adhere to the **geometric constraints** (display/aspect ratio)
2. Preserve the **important *content*** and *structures*
3. **Limit *artifacts***
4. **Perhaps a new representation that will support different sizes?**

- **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

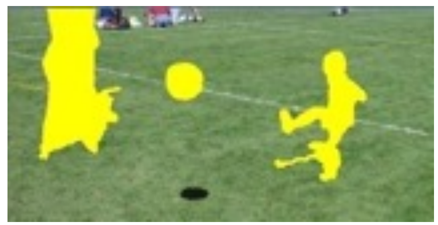
- **More sophisticated: attention models, eye tracking (gazing studies), face detectors, …**

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

General Retargeting Framework

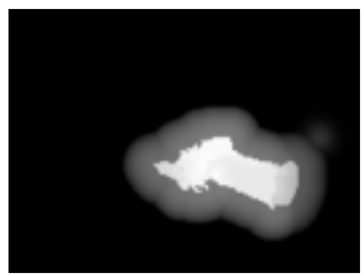
1. Define an energy function $\mathbf{E(I)}$ (interest, importance, saliency)

2. Use some operator(s) to change the image $\mathbf{I}$

Recompose

Setlur et al. [2005]

Crop

Santella et al. [2005]


Warp

Gal et al. [2006]

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"

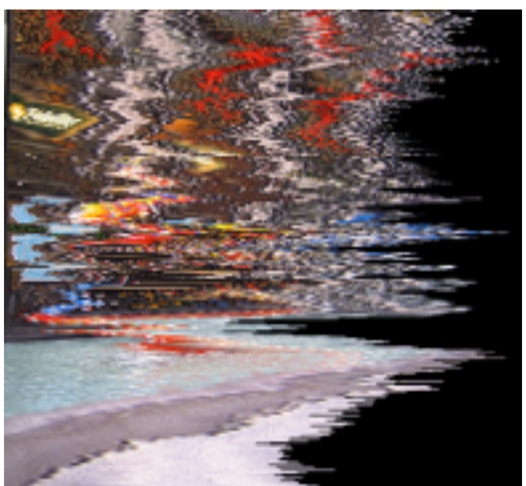
$$E_1(\mathbf{I}) = |\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}|.$$

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

Energy

Pixel Removal

Seam carving!!



Globally optimal

Least-energy pixels (per row)

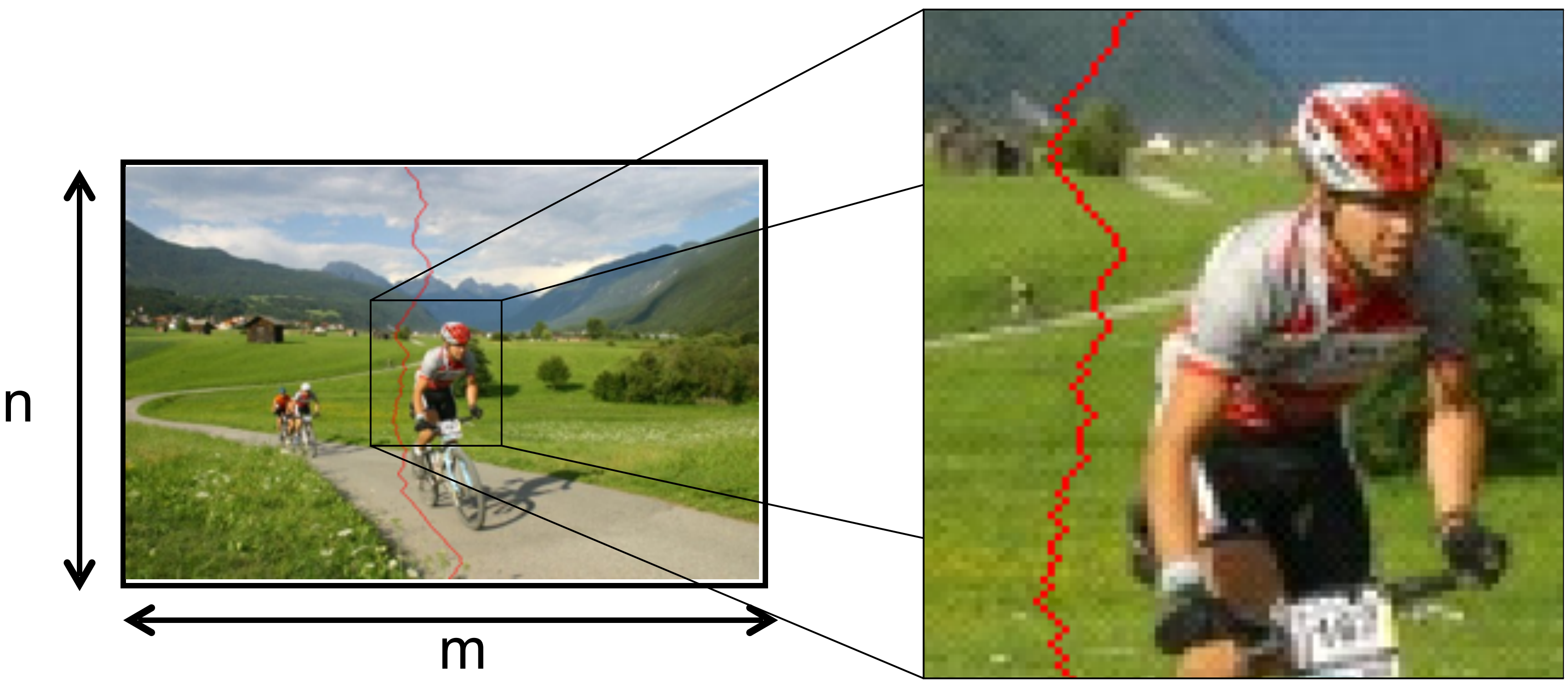
Least-energy columns

A Seam

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

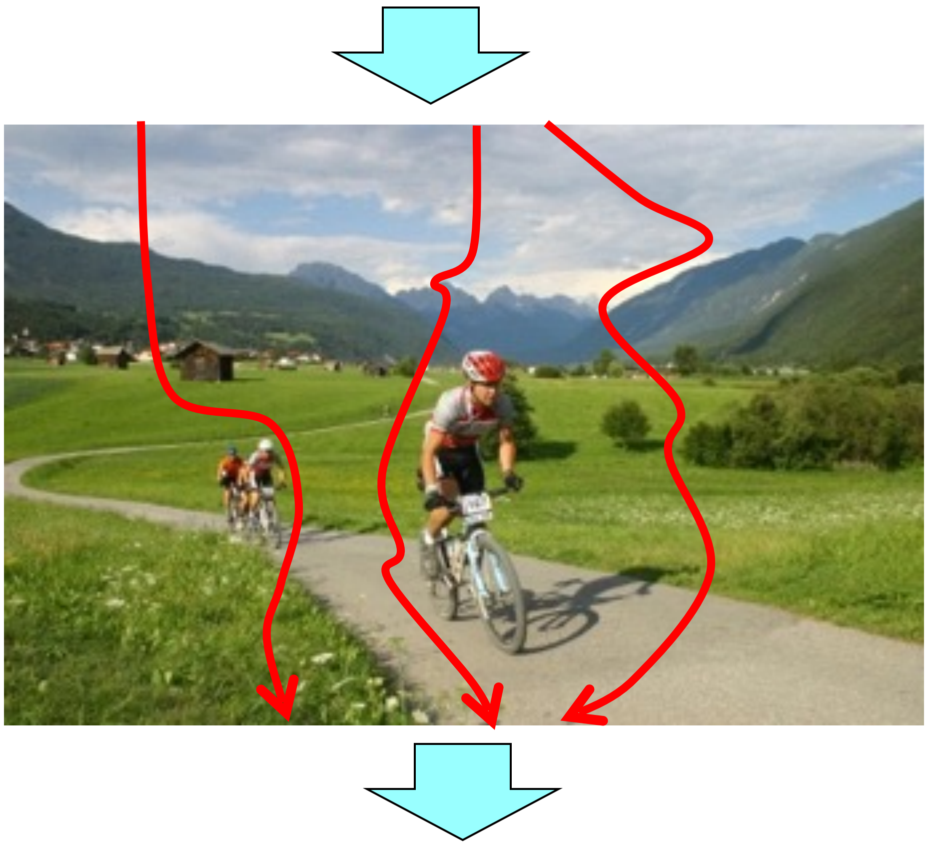
$$\mathbf{s}^{\mathbf{x}} = \{s_i^x\}_{i=1}^{n} = \{(x(i), i)\}_{i=1}^{n}, \text{ s.t. } \forall i, |x(i) - x(i-1)| \leq 1$$

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

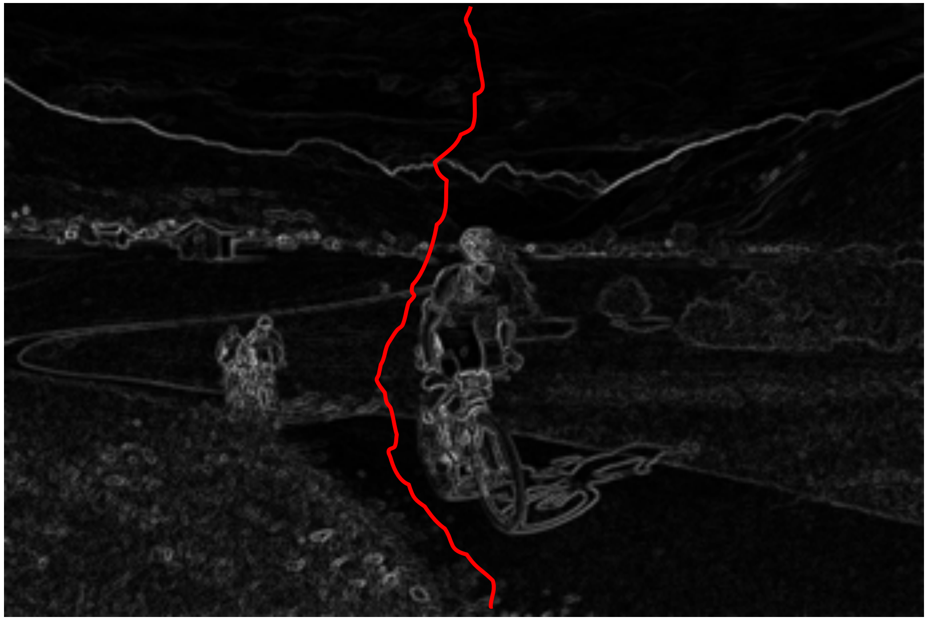


Michael Rubinstein — MIT CSAIL – mrub@mit.edu

Finding the Seam?



The Optimal Seam



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

Find a set of seam pixels whose sum of energy is the smallest.

The Optimal Seam

- **The recursion relation**

Cumulative minimum energy

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

(row, column)

- **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 aggregated cost of a seam going through (i,j) (satisfying the seam properties)

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

Dynamic Programming

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

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

Michael Rubinstein — MIT CSAIL – mrub@mit.edu

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	9
7	3	4	2
4	5	7	8

Michael Rubinstein — MIT CSAIL – mrub@mit.edu

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

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

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

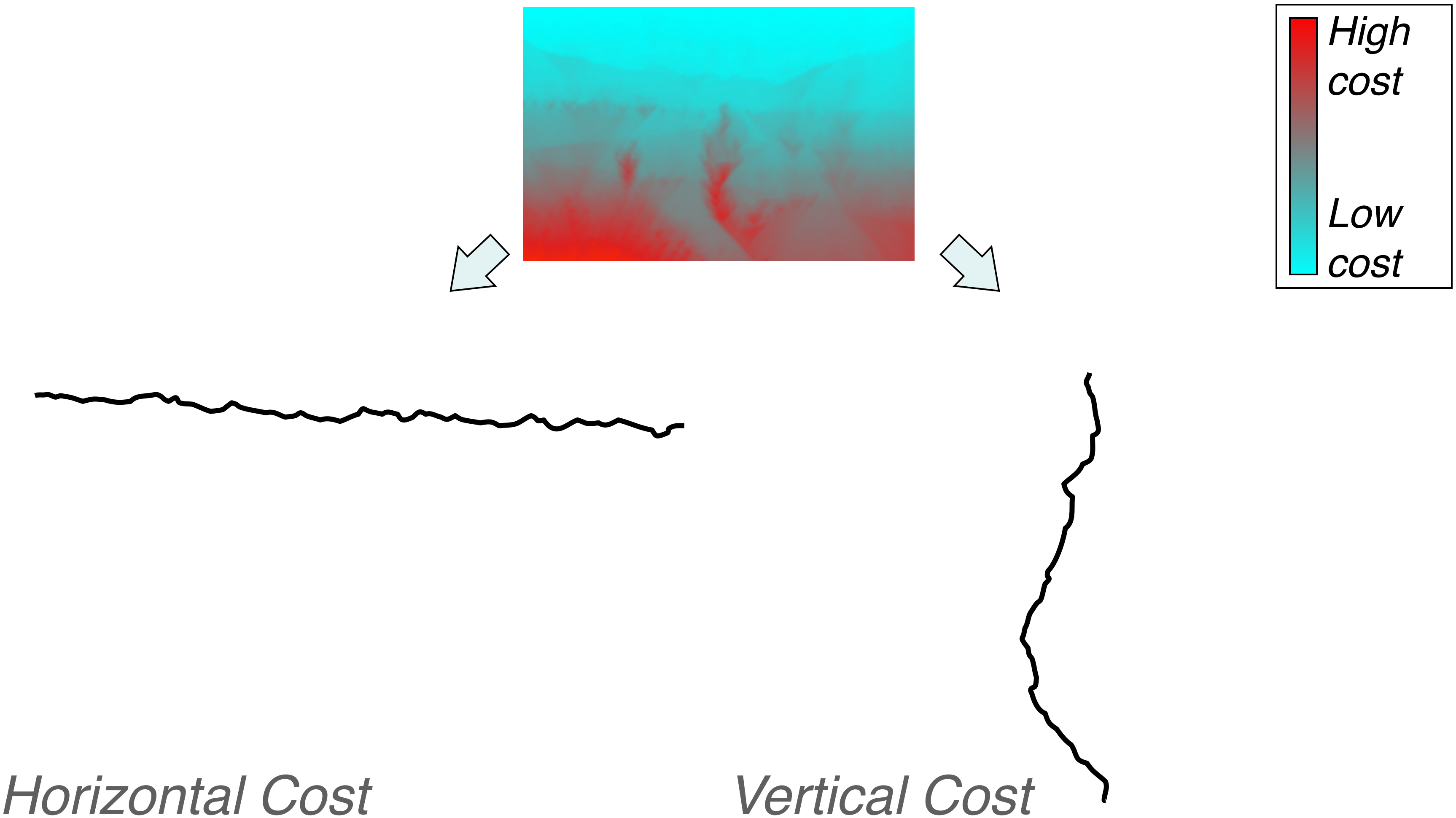
Backtracking the Seam

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

Backtracking the Seam

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

H & V Cost Maps



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

Michael Rubinstein — MIT CSAIL – mrub@mit.edu

Changing Aspect Ratio

Original

Seam Carving

Scaling

Changing Aspect ratio

Cropping

Seams

Scaling

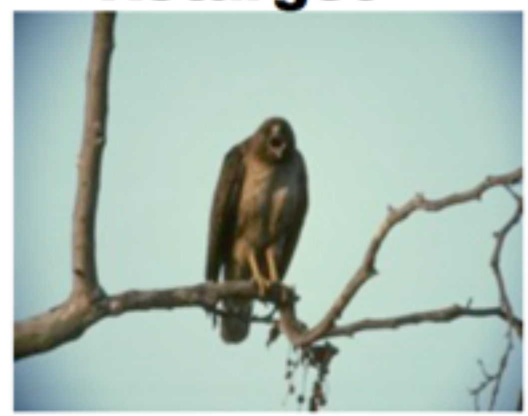
Changing Aspect Ratio

Original **Retarget** *Scaling*

Changing Aspect Ratio

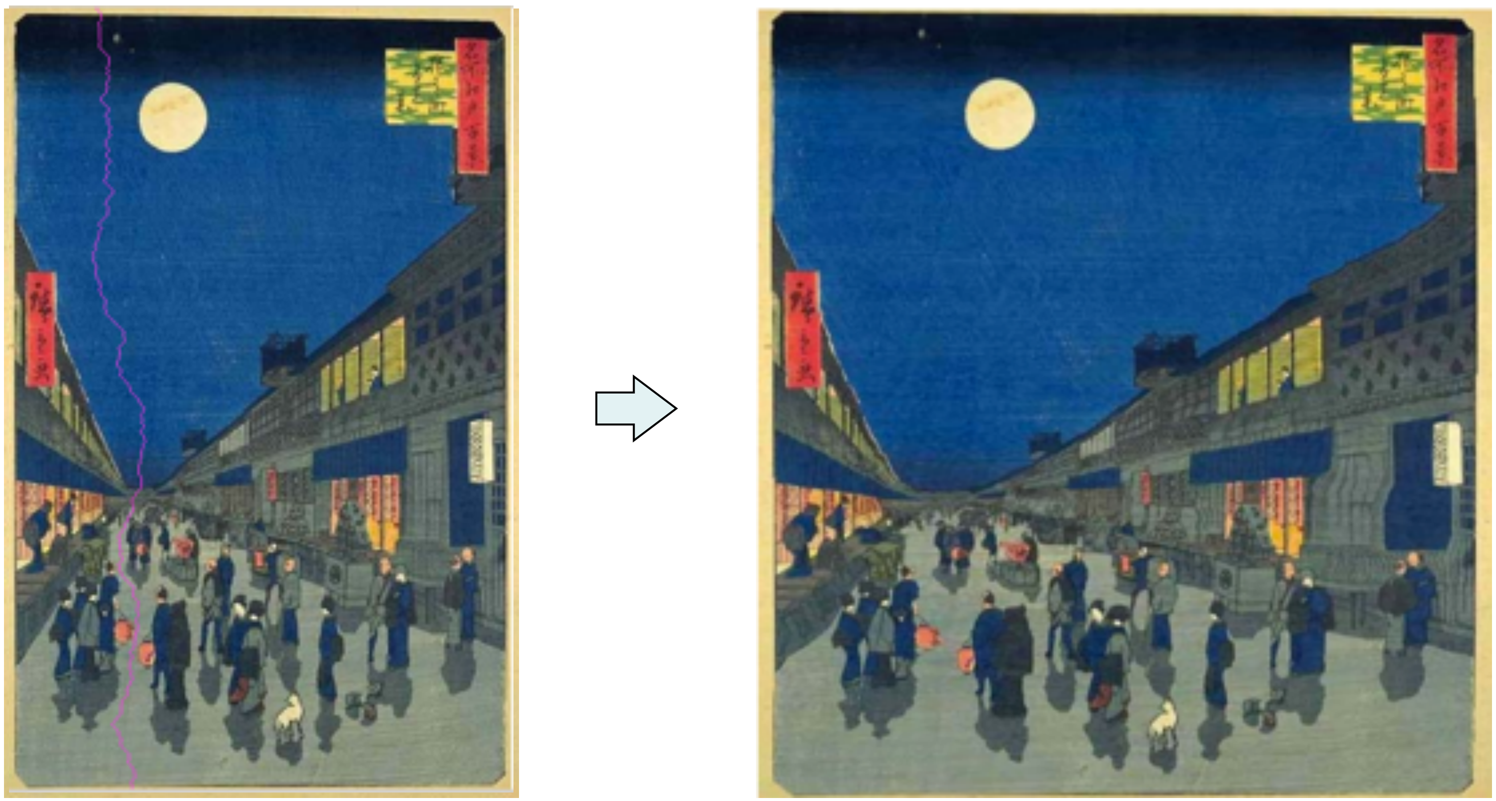
Original

Retarget



Scaling

Image Expansion (Synthesis)



Combined Insert and Remove

Insert & remove seams

Scaling

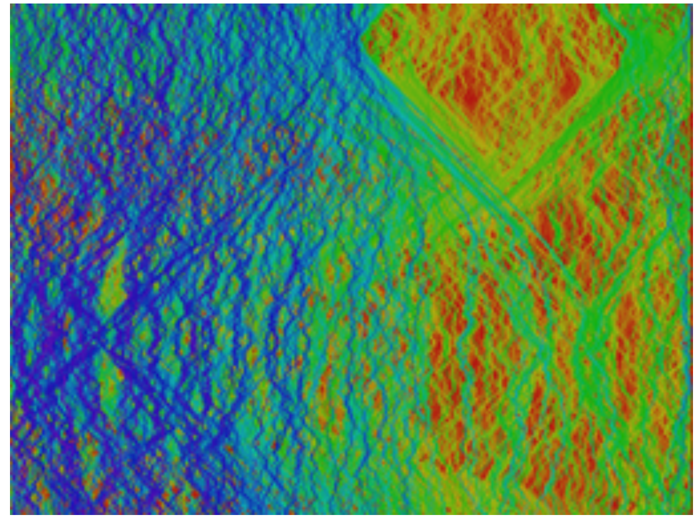
Multi-Size Images

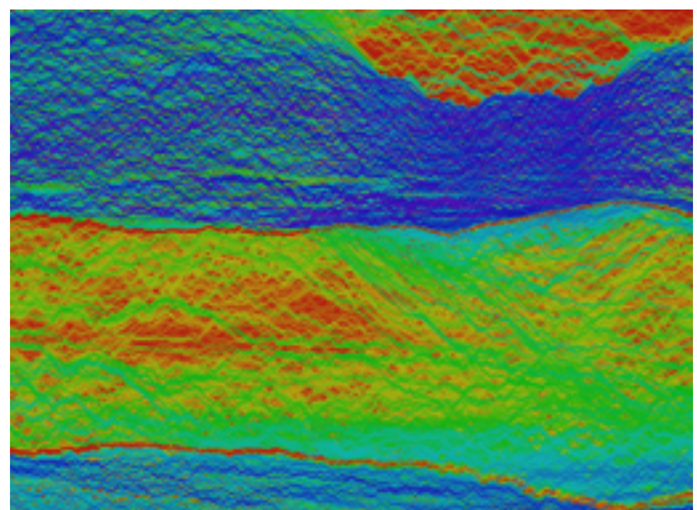
- **We can create a new *representation* of an image that will allow adapting it to different sizes!**

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

Multi-Size Images

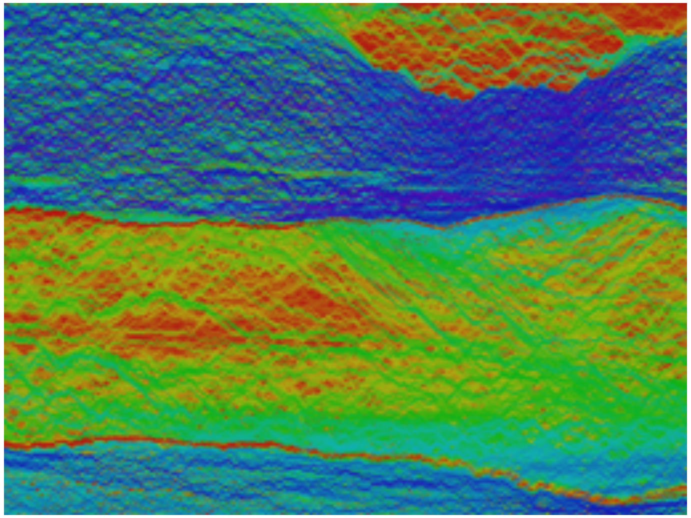
First to be removed
•
•
•
Last to be removed





Multi-Size Image Representation

+



Multi-Size Image Representation

Extension: Adjusted Energy

Object Removal

Object Removal

Input | Retargeted | Pigeon Removed | Girl Removed

Find the Missing Shoe!




Michael Rubinstein — MIT CSAIL – mrub@mit.edu

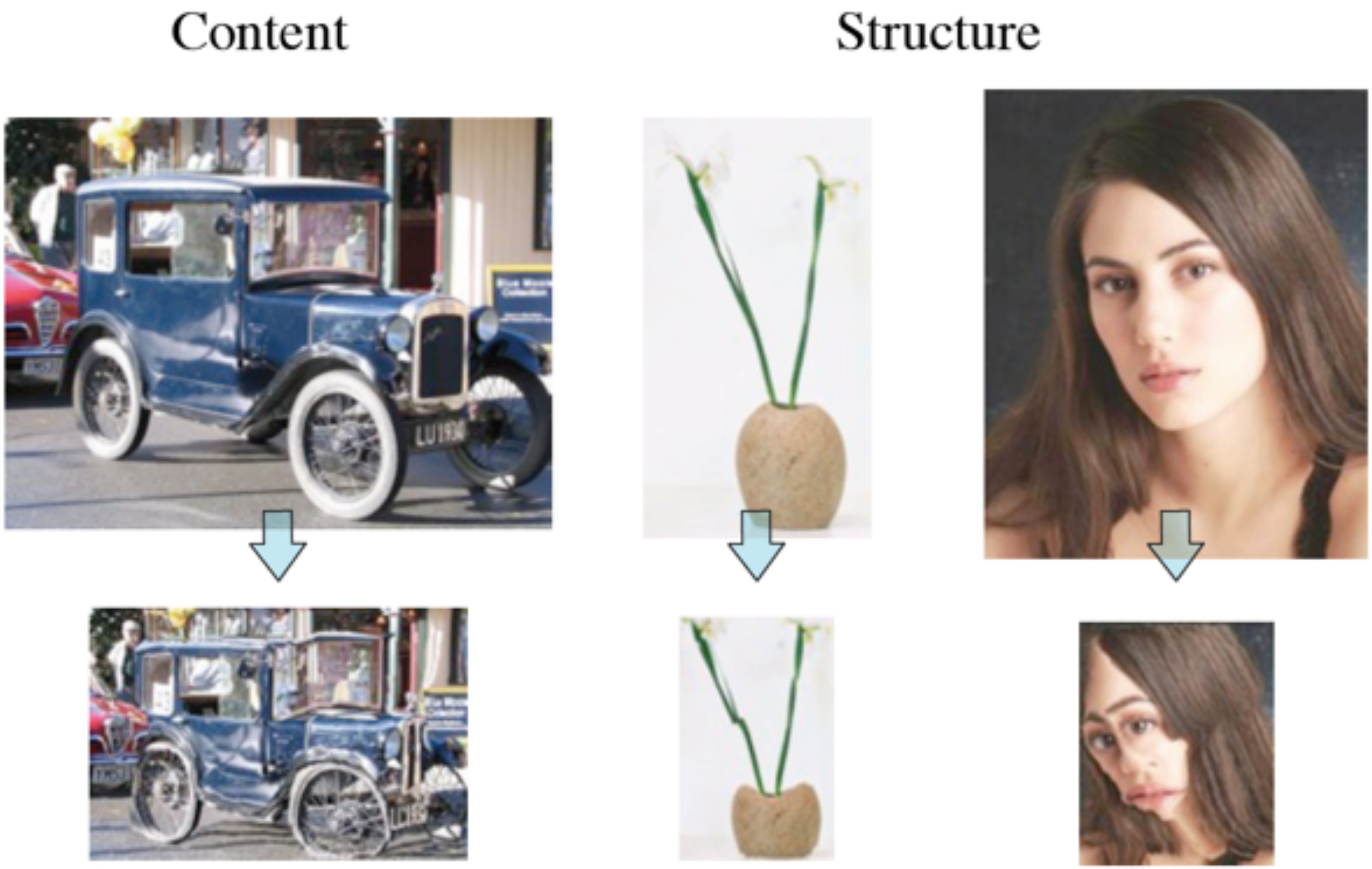
Solution

With face detector

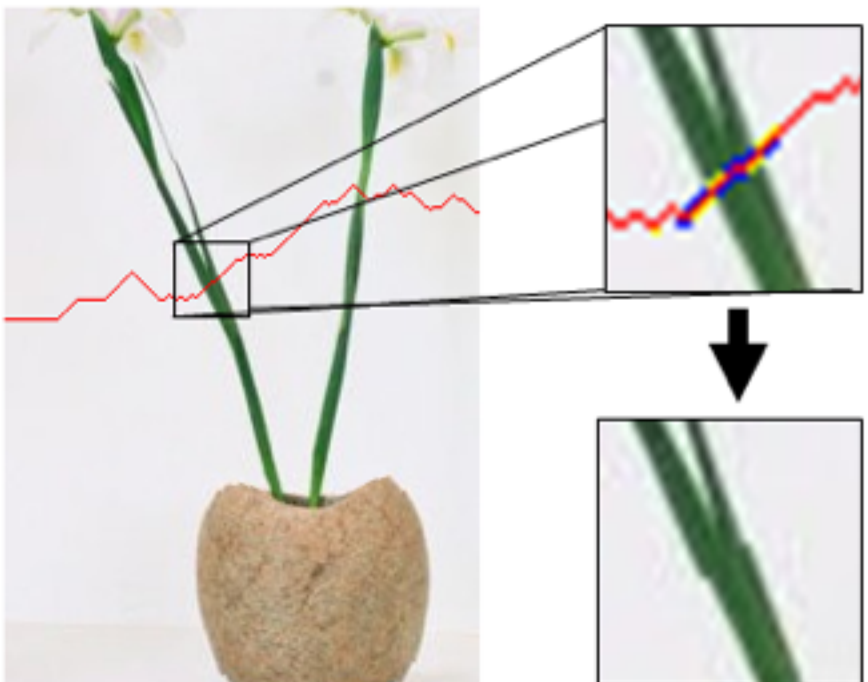
With User Constraints

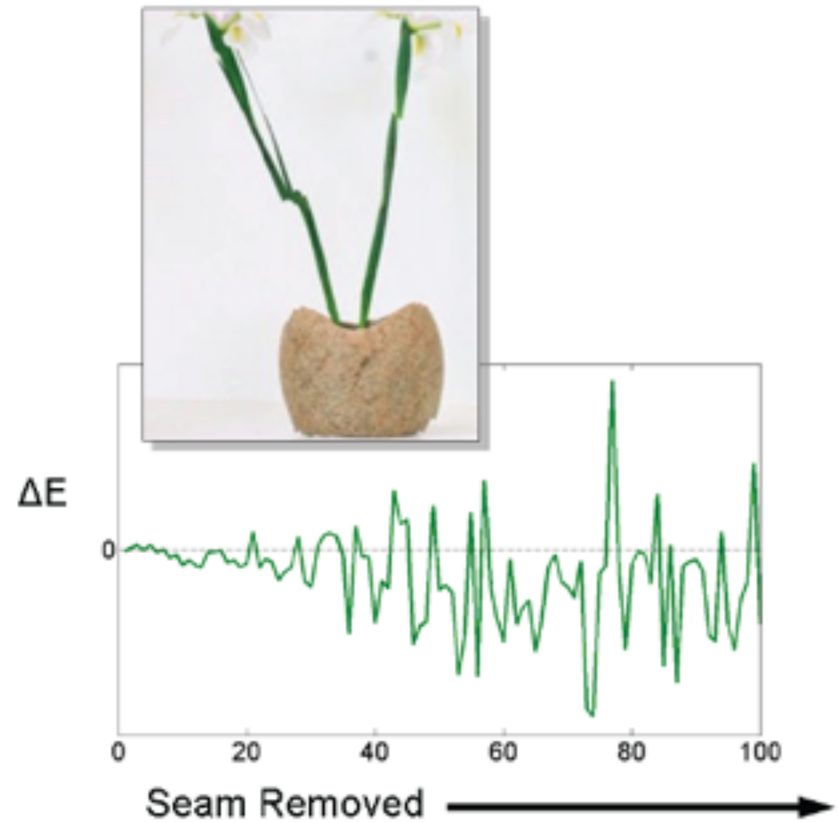
Extension: Insert Energy

Limitations



Problem: Removing a seam INCREASES energy!





Minimize Inserted Energy

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

Old "Backward" Energy

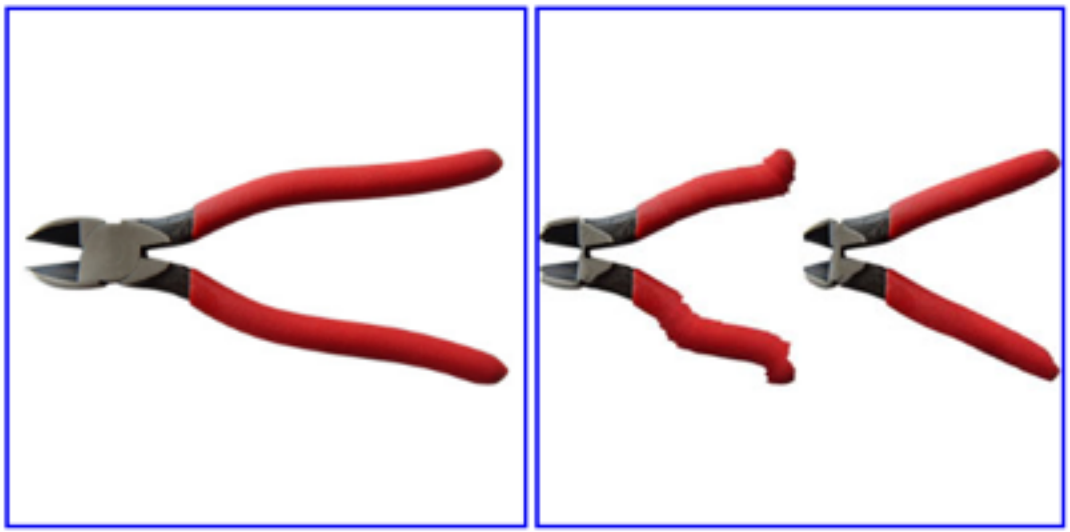
$$M(i,j) = E\,(i,j) + \min \begin{cases} M(i-1,j-1) \\ M(i-1,j) \\ M(i-1,j+1) \end{cases}$$

New Forward Looking Energy

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

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

Results



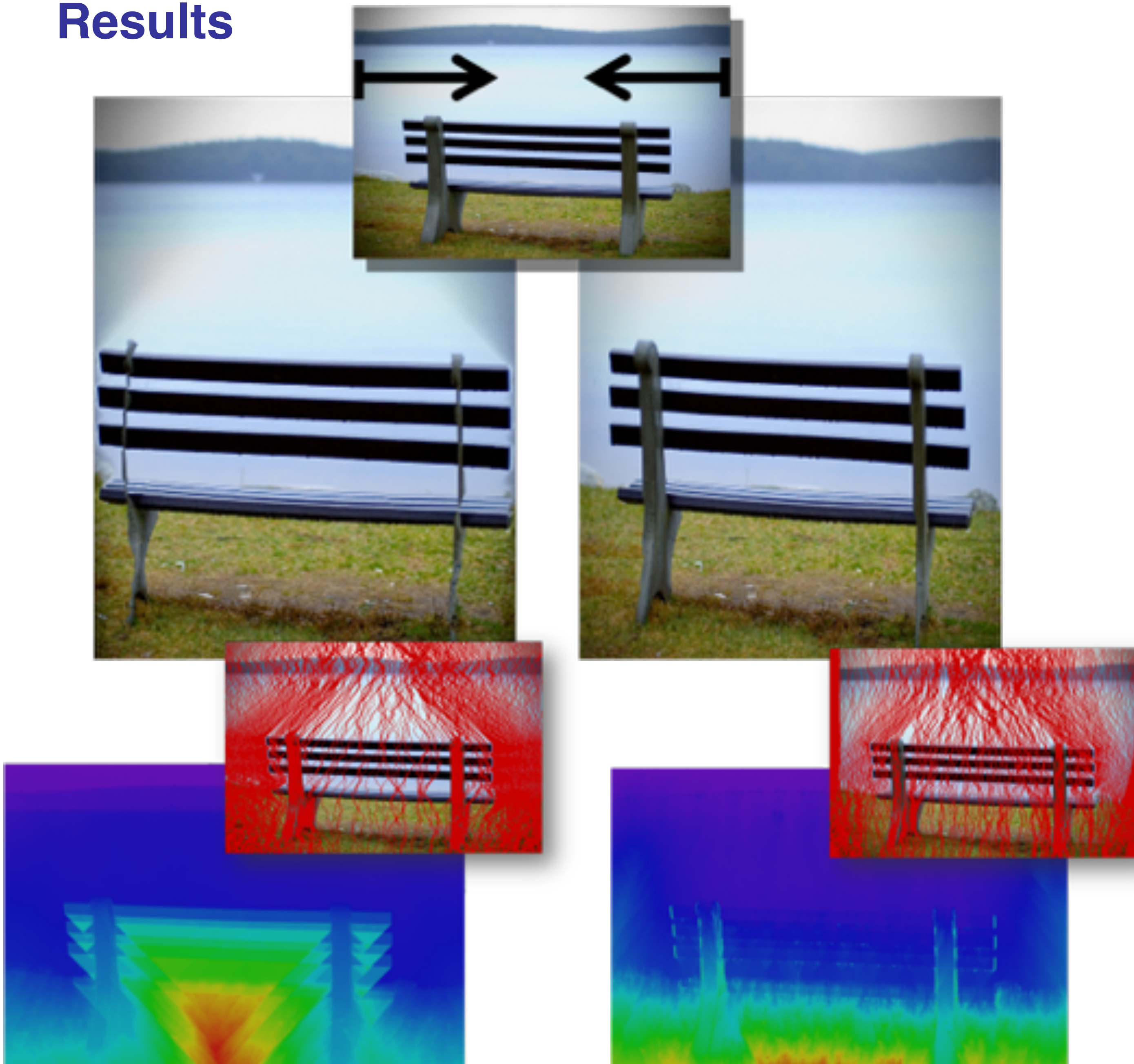
Input Backward Forward

Input

Backward

Forward

Results



Backward vs. Forward

Backward

Forward

Extension: Video

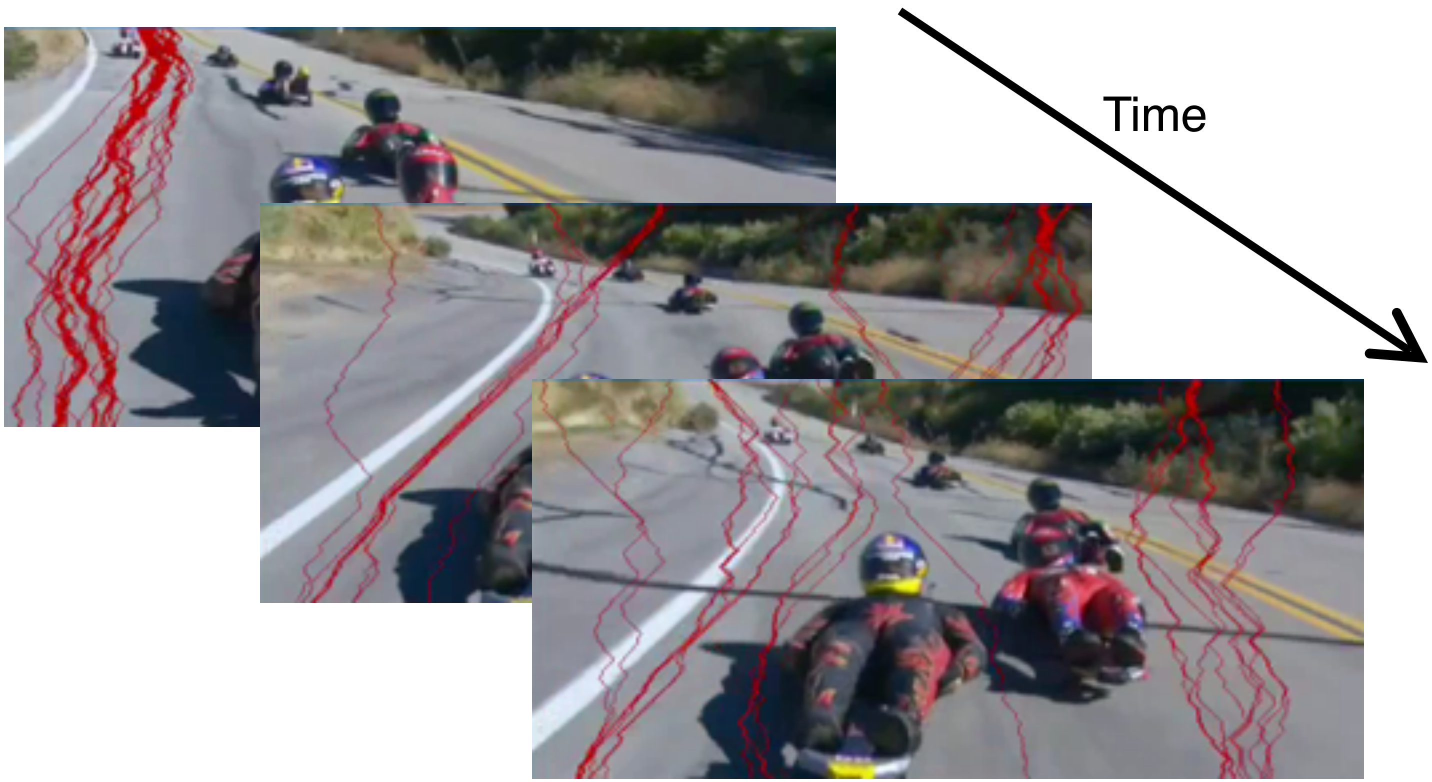
From Images to Videos

- **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**



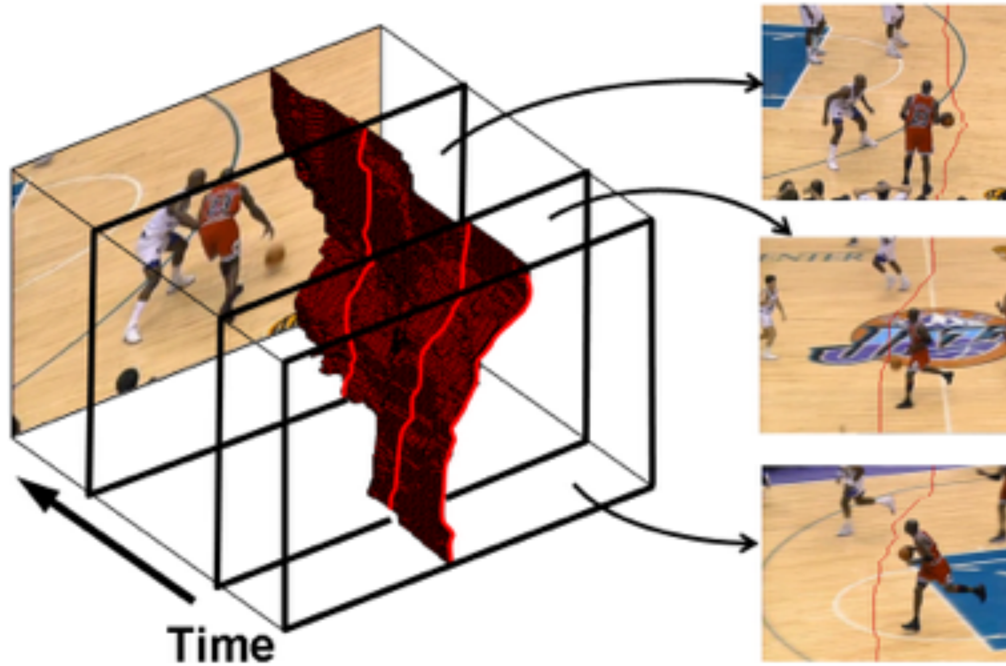
Frame-by-frame Seam-Carving

*Representative seams

Michael Rubinstein — MIT CSAIL – mrub@mit.edu

From 2D to 3D

1D paths in images



2D manifolds in video cubes

Michael Rubinstein — MIT CSAIL – mrub@mit.edu

Challenges

- **Dynamic Programming no longer applicable**
  - Reduction to min-cut graph problem

- **Cut must fulfill seam constraints**
  1. *Monotonic* (cut each row exactly once)
  2. Connected

Cut should be a function!

Improved Seam Carving for Video Resizing

Michael Rubinstein
Mitsubishi Electric Research Lab
Ariel Shamir
The InterDisciplinary Center
Shai Avidan
Adobe Systems Inc.