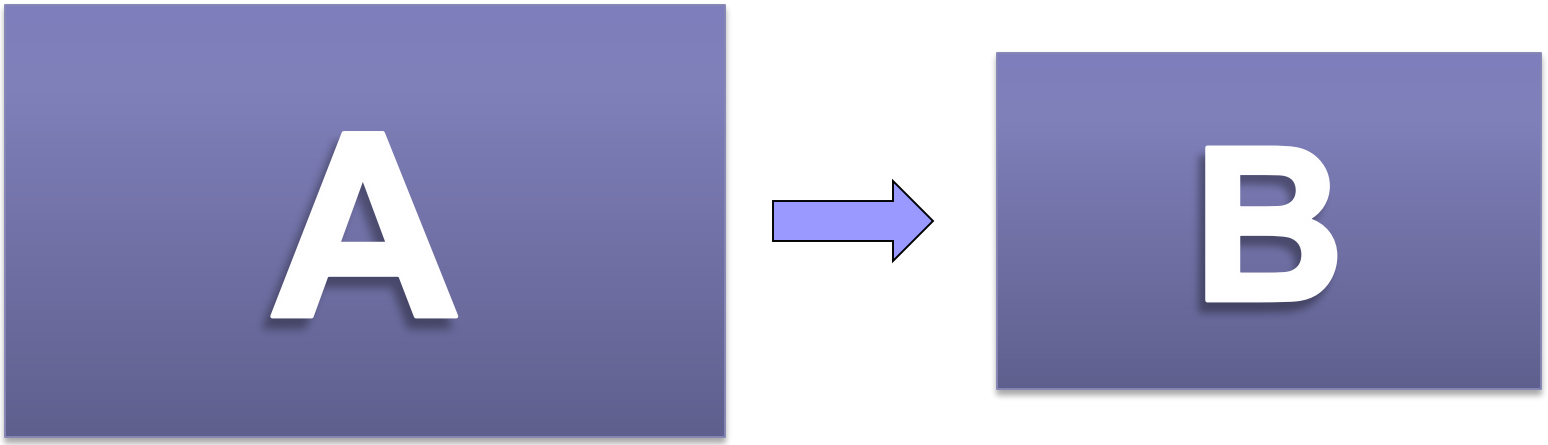
Image Resizing Using Seam Carving

John Entrieri

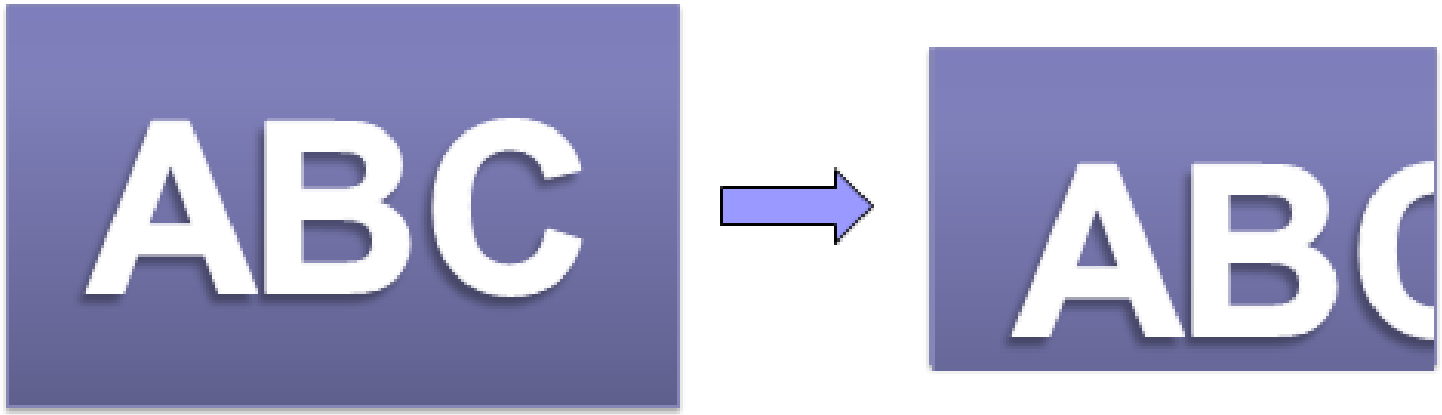
Jiaheng Fu

Background



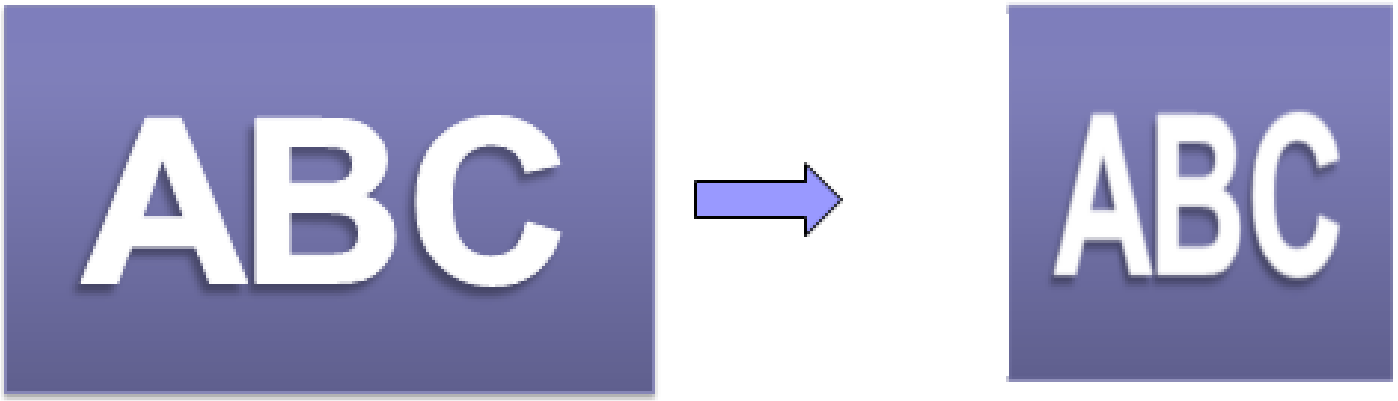
- Image Resizing
  - Shrinking – Data must be removed
  - Enlarging – Data must be added

Conventional Methods



- Cropping
  - Loss of Quality – None
  - Loss of Content – High

Conventional Methods



- Scaling
  - Loss of Quality – High
  - Loss of Content – None

Problem Statement

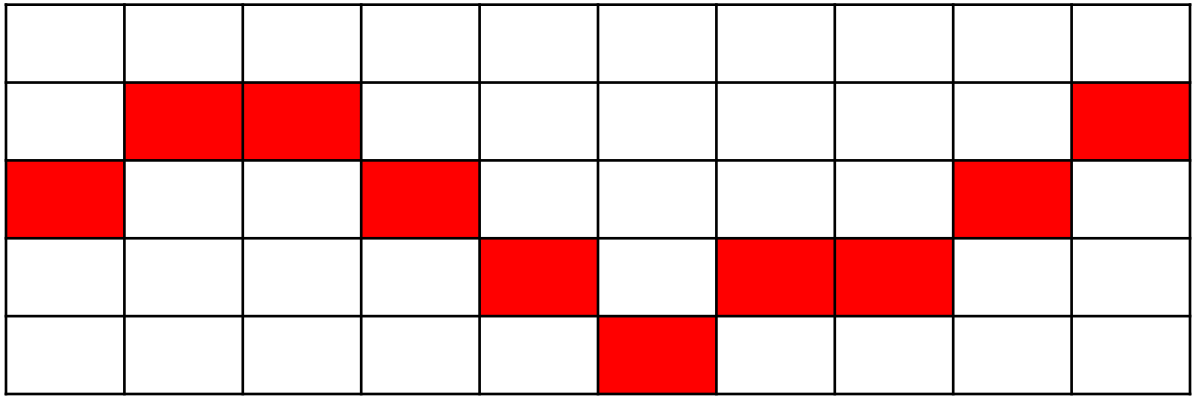
- How can we resize an image in a way that minimizes losses in both quality and content?

SOLUTION

- Seam Carving

Seam Carving

- Seam – Path across an image either horizontally or vertically



- Calculate and remove the least “important” seam, reducing Height or Width by 1 pixel

Energy Function

- How can we define the “importance” of a pixel?
- How can we represent it quantitatively for mathematical use?

0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0

1	0	1	1
1	0	1	0
1	1	0	0
1	1	0	0

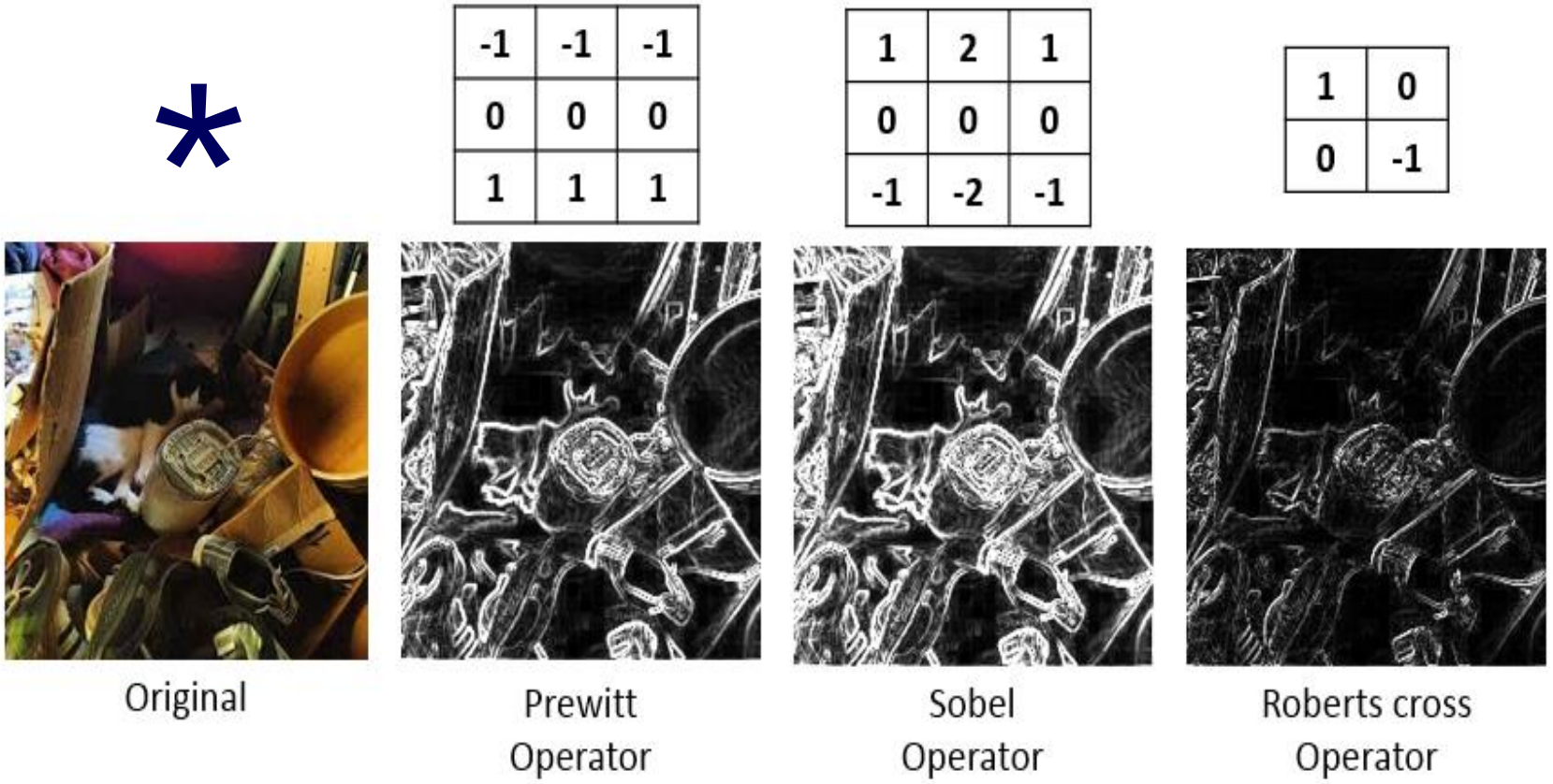
Which pixels are “important”?

Energy of a pixel:

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

Image Gradient

- Convolve each image channel (RGB) with one of several differentiation operators and average them.



Path Detection

- Using this new “energy matrix” how can we determine which seam to remove?
- Straight Forward Approach – Compare the sums along every potential path
  - Far too “expensive”
- Dynamic programming – Create “energy map” or matrix of minimum possible path sum at a given location

Creating an Energy Map

x →

y ↓

1	3	2	4
2	1	1	3
4	3	3	1
2	2	5	1

Energy of Image: E(x,y)

1	-	-	-
2	-	-	-
4	-	-	-
2	-	-	-

Copy first row or column

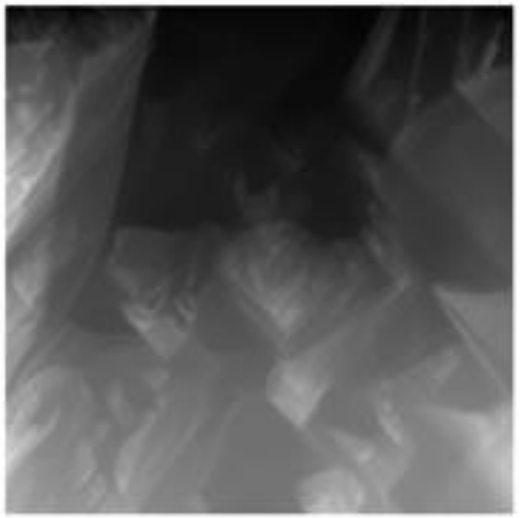
1	3+1	-	-
2	-	-	-
4	-	-	-
2	-	-	-

EM(x,y) = E(x,y)+minimum possible touching EM value (options highlighted)

1	4	4	7
2	2	3	6
4	5	5	4
2	4	9	6

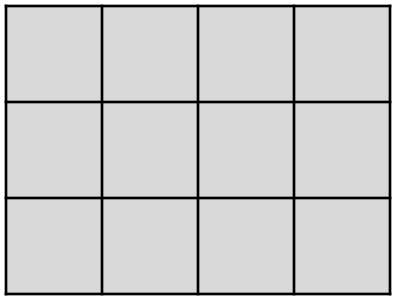
Energy Map of Image: EM(x,y)

Energy Map viewed as an image



Using an Energy Map

1	4	4	7
2	**2**	**3**	6
4	5	5	**4**
2	4	9	6

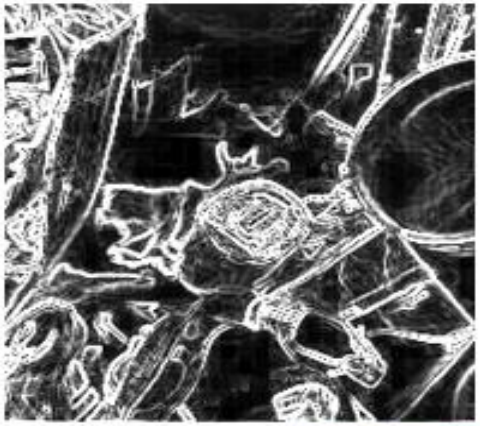


- Work Backwards
  - Find smallest value in final column
  - Find minimum touching, previous value
  - Continue until edge is reached
- Remove chosen locations from original image matrix and shift
- Result is a successfully resized (by 1px) image

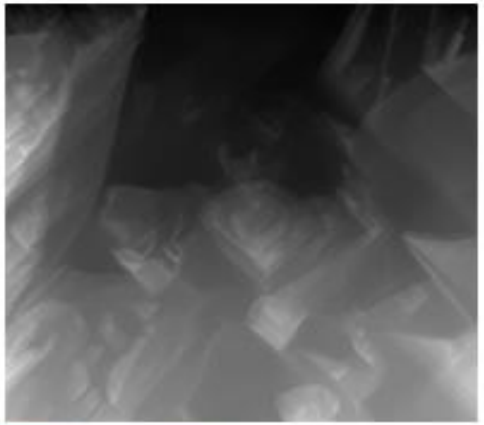
The Seam Carving Algorithm

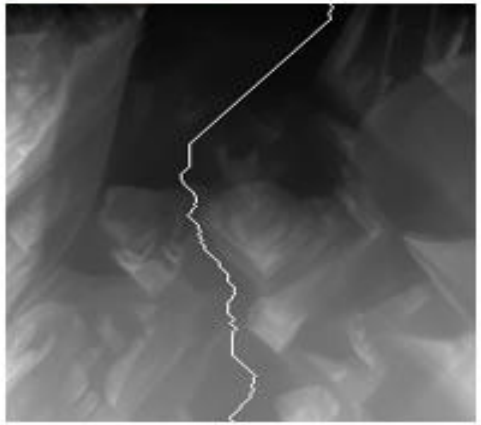
Original

Energy



Energy Map





Seam Discovery

Results

Original Picture

25 Seams Removed

50 Seams Removed

- Quality retained
- Nothing meaningful lost

More Results

Original Picture

25 Seams Removed

50 Seams Removed

- Algorithm runs out of unimportant seams and distorts meaningful parts of the image

- Scaling is the better method for these “high energy” images

Even More Results

Original Picture

25 Seams Removed

50 Seams Removed

- Computers are “dumb”, cannot infer what we want
- High energy trees, grass, and rocks cause chosen seams to cut through the houses

Possible Solutions

- In some cases, the energy function disagrees with what the user wants.
- The user can highlight areas they wish to not be tampered with
- High energy values are injected into the section's pixels

Conclusions

- For certain images, seam-carving blows alternatives away
- In some cases, scaling may be superior
- Sometimes, the user must provide extra input to obtain the desired output.

Future Applications

- Real-time Processing
  - Web browsers
  - Image editing tools
  - Office suites
- Image Completion
  - Similar to solution on previous slide
  - Remove unwanted sections of an image
  - Highlight areas to be removed, and low energy values are injected

References

[1] S. Avidan and A. Shamir, "Seam Carving for Content-Aware Image Resizing," ACM Transactions on Graphic, 2007.

[2] D. Luong, "Seam Carving for content aware image resizing: GUI implementation demo," 2009.

Questions?

Thank you for your time