Seam Carving

CPS592 – Visual Computing and Mixed Reality

Outline

1) Image Resizing
2) Seam Carving Algorithm
3) What is argmin?
4) More Applications

Given image

Display

Different Displays-Different Resolutions-(Different Images??)

Do you remember this?

- Resize the image while maintaining the content and the quality.

Image Resizing

Problem: Difficult to shrink a photo horizontally/vertically without skewing the proportions of the content within.

fail

fail

fail

Image Resizing

Goal: An algorithm that can identify significant aspects of an image.

- Important elements of the photo should be preserved
- Repetitive areas of the image (water, sand) can be reduced

Image Resizing

Solution: Resize a photo by discarding uninteresting areas, maintain proportions of important items.

fail

fail

fail

SUCCESS!

Image Resizing

- In order to narrow an image, delete “unimportant” pixels.
- “Importance” of a pixel can be approximated by looking at how much it varies from its neighbors.
- It is defined as the sum of the differences of its intensity from that of neighboring pixels.

Image Resizing

- Grayscale 3x3 image with the following pixel intensities
- What's the importance of the center pixel?

4	6	5
2	5	7
3	2	6

Image Resizing

- Grayscale 3x3 image with the following pixel intensities
- What's the importance of the center pixel? **1+3+2+3=9**

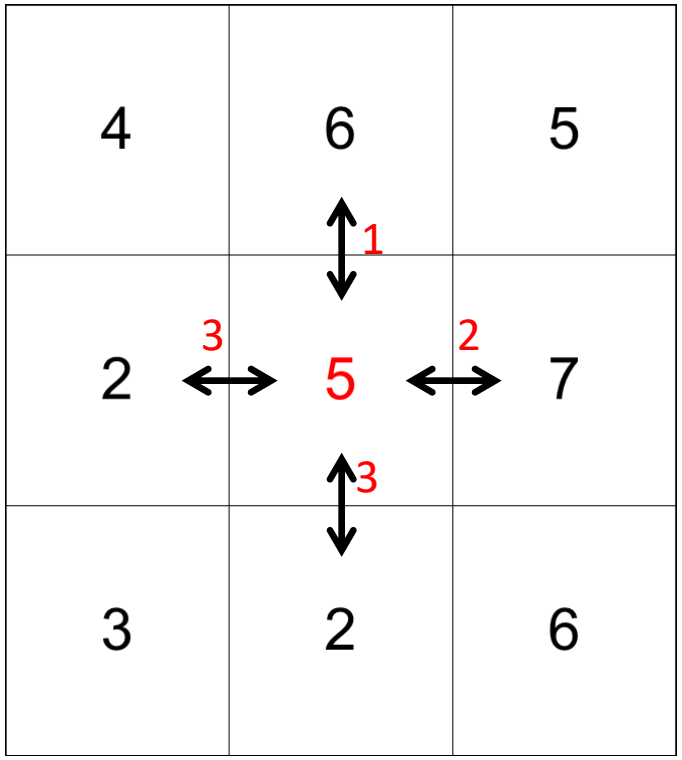
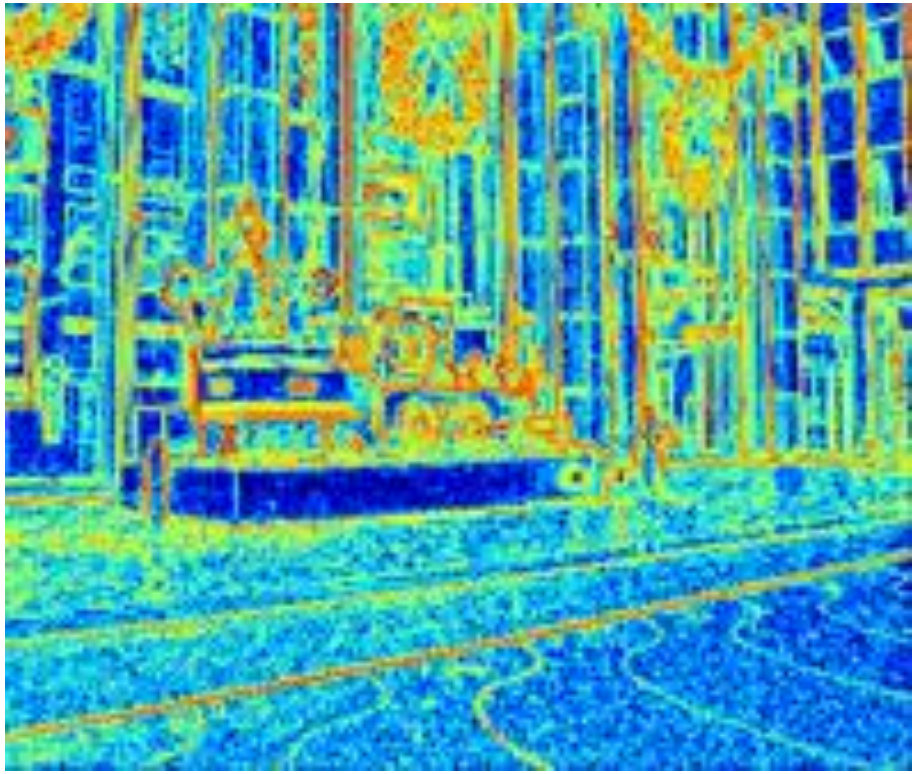


Image Resizing

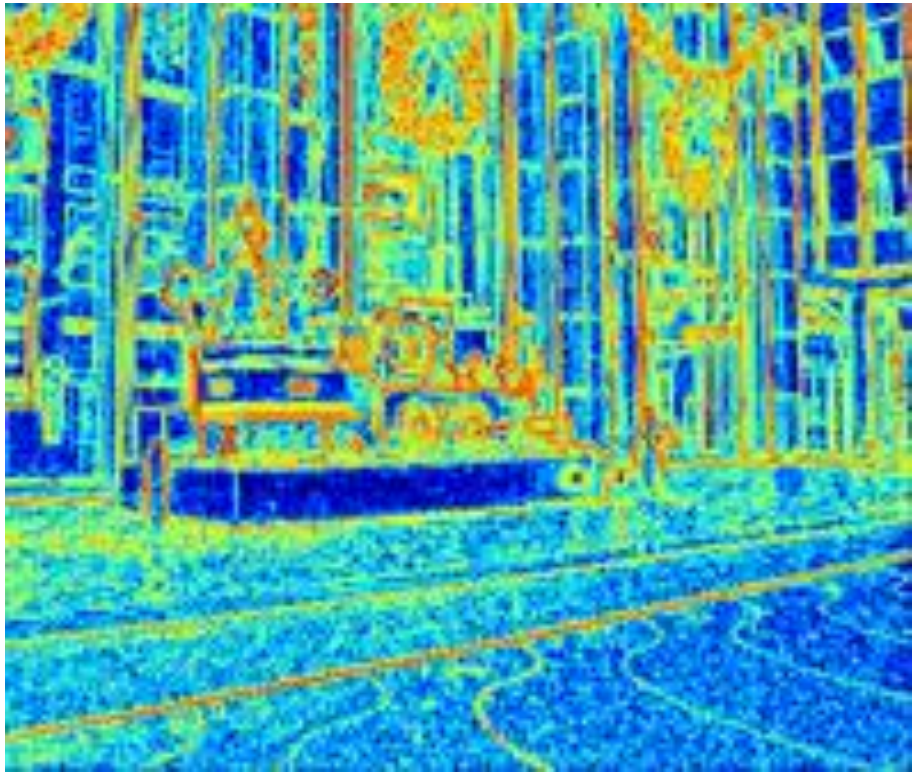
- Apply this method to every pixel of an image to determine the most and least “important” pixels

Low High



What is the importance map?

- It can be considered as a saliency map.



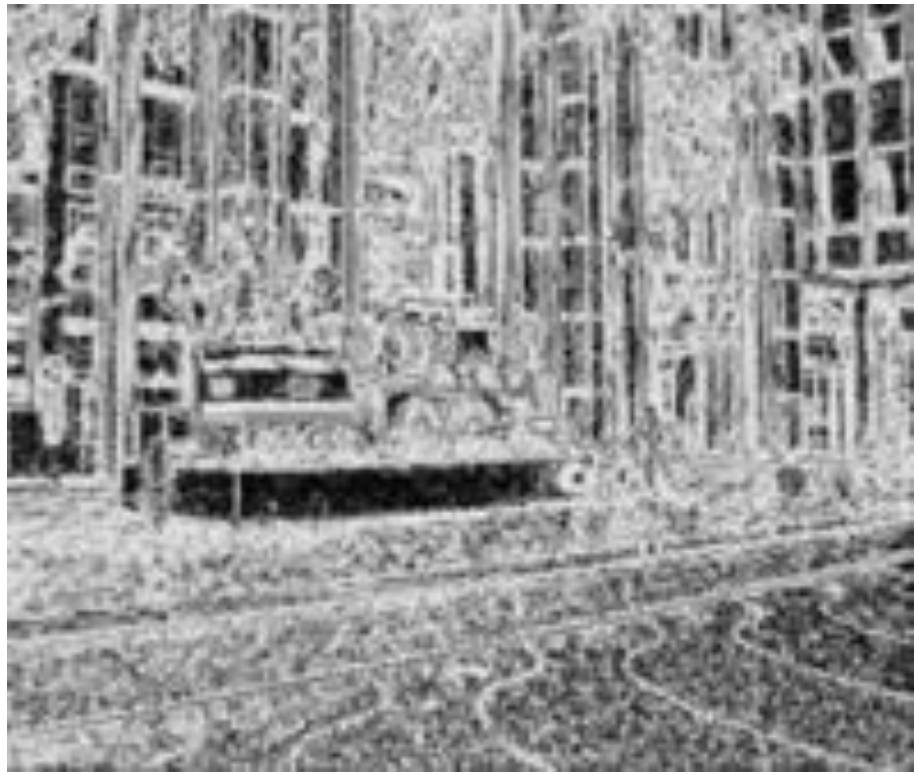


Image Resizing: Approach 1

- Remove the least important pixels in order
- Looks terrible! Not removing the same amount from each row, causing jagged right side

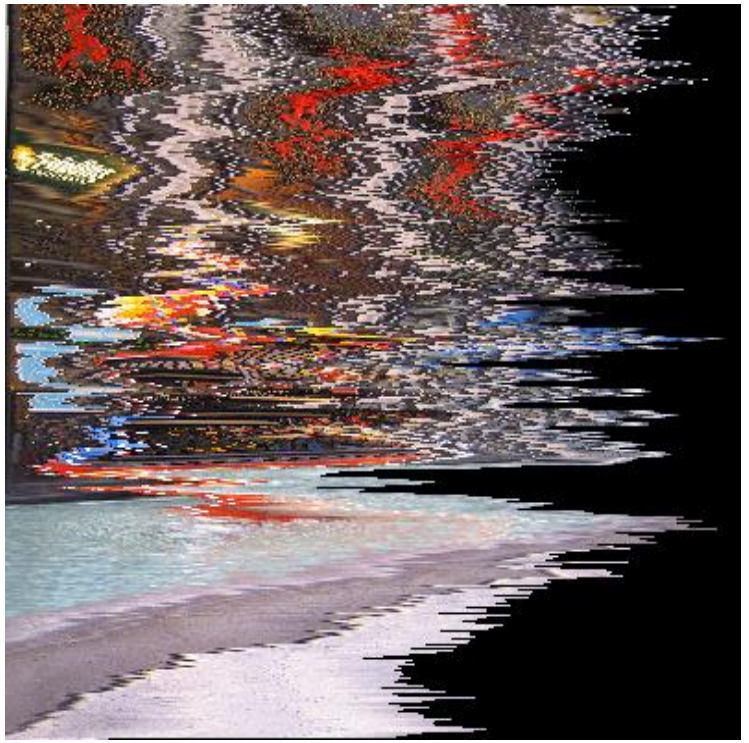


Image Resizing: Approach 2

- Remove *n* least important pixels in each row
- Still not great, too much shifting between adjacent rows

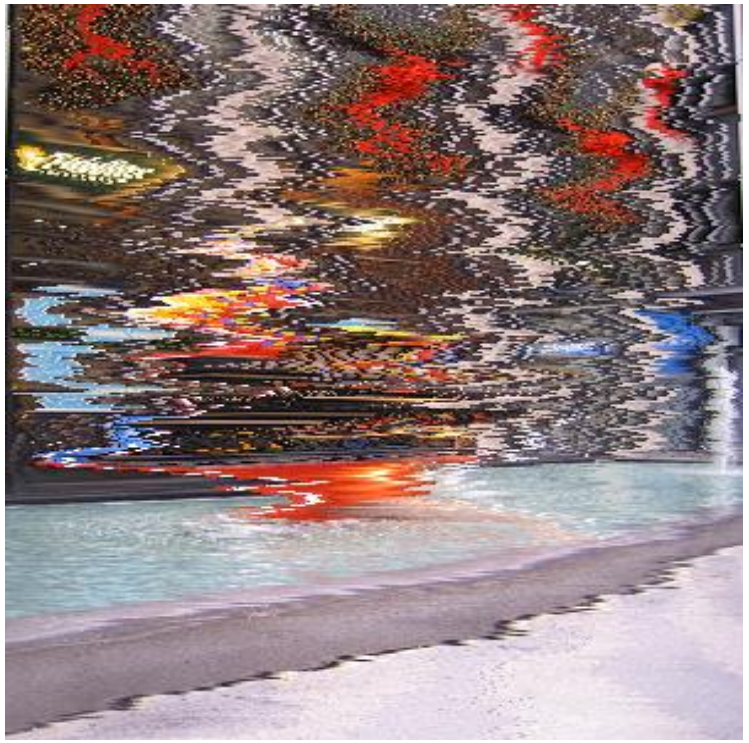


Image Resizing: Approach 3

- Remove the column whose total importance is smallest, and repeat
- Much better! But not perfect…

Image Resizing

- Problem: removing an entire column (or an entire row) distorts the image

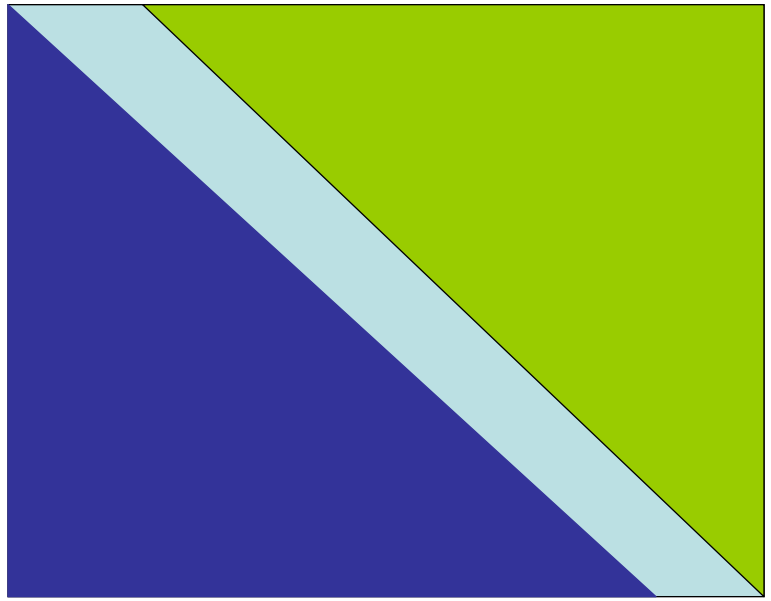


Image Resizing with Seam Carving

- A: What is a seam?
- Q: It is a line along which two pieces of fabric are sewn together in a garment or other article.

Image Resizing with Seam Carving

- Solution: A *seam*, a path from top to bottom that moves left or right by at most one pixel per row (vertical seam).
- In this example, a vertical and horizontal seam is shown.

Image Resizing with Seam Carving

- Using the Seam Carving approach.
- Near perfection!

The Seam Carving Algorithm

Function: `find_least_important_seam(vals)`
Input: *`vals`*, a 2D array of importance values

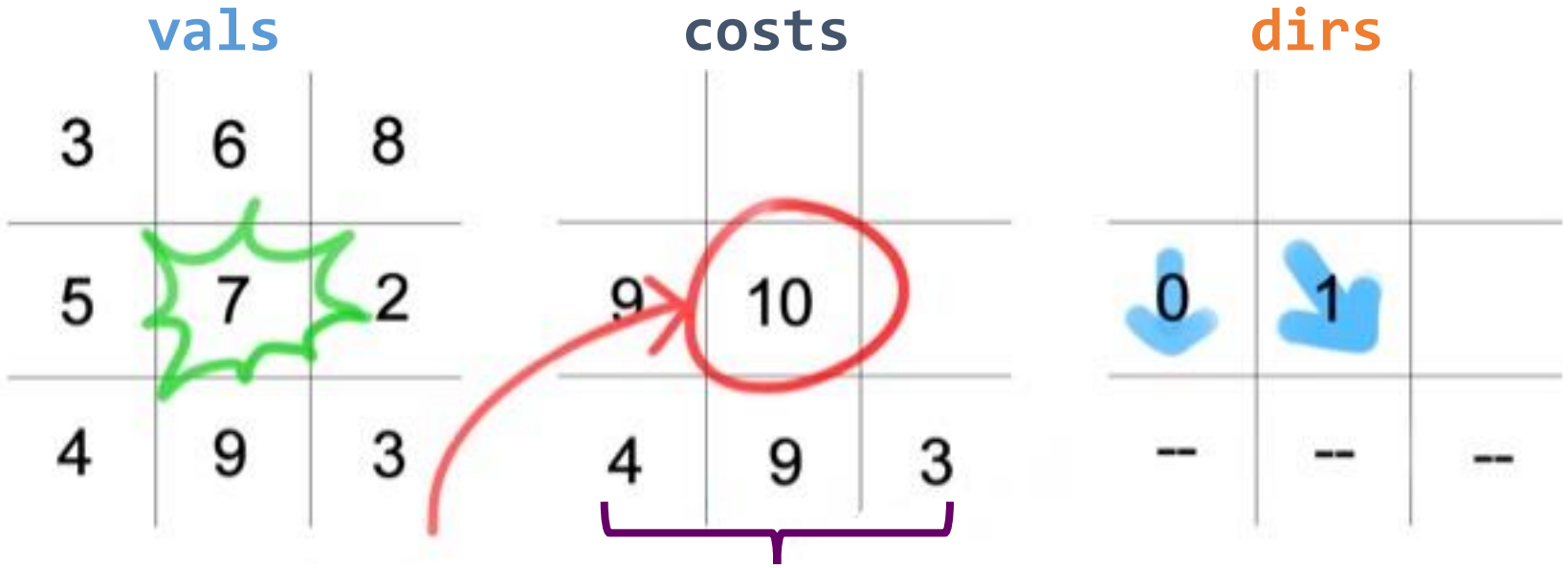
Output: sequence of column-indices, each differing from the last by at most 1, with corresponding pixels being the least-total-importance vertical seam

```
[[ - S - - ],
 [ S - - - ],
 [ - S - - ],
 [ - - S - ]]
```

```
[1, 0, 1, 2]
```

Data Structures needed

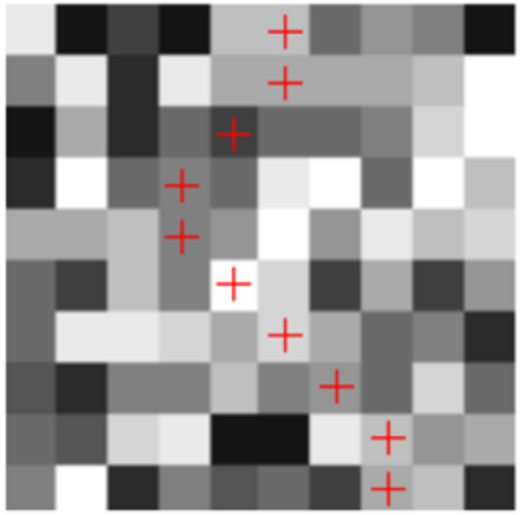
- `costs` - 2D array, filled in from bottom to top
  - `costs[row][col]` holds the total importance of the lowest cost seam starting from the bottom row and ending at `costs[row][col]`
- `dirs` - 2D array, filled in at the same time as `costs`
  - `dirs[row][col]` holds the direction (-1, 0, or 1) of the previous pixel in the lowest cost seam ending at `costs[row][col]`



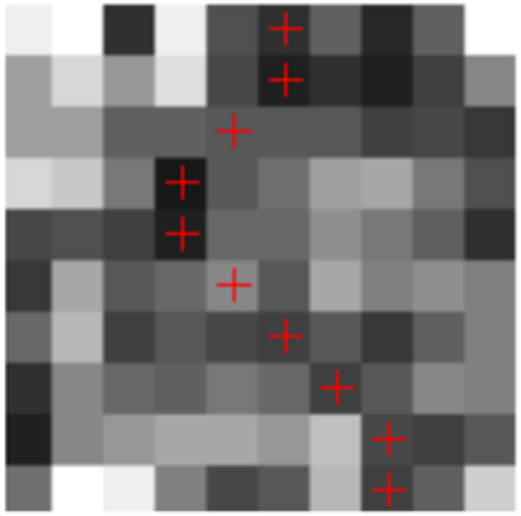

```
costs[row][col] = min(costs[row+1][col-1 to col+1]) + vals[row][col]
```

Data Structures illustrated

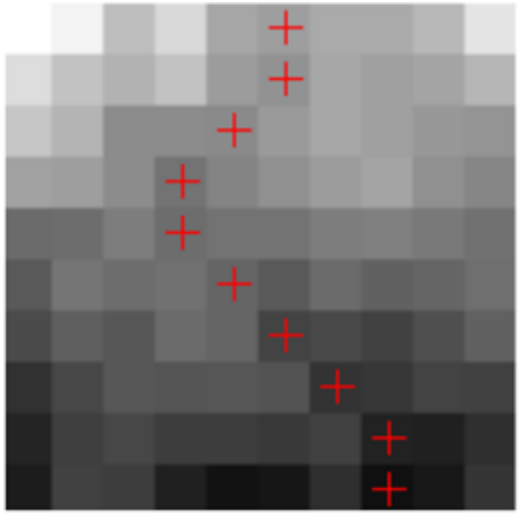
original image



`vals`



`costs`



high

low

Finding the least important seam

- Once **`costs`** has been completely filled in, the cell with the minimum value in the top row of **`costs`** will be the first pixel in the least important seam of the entire image.

- Starting from that pixel, we can use the **`dirs`** array to backtrack our way through the rest of the seam and build the final list of column indices.

Seam Carving Pseudocode

```
function find_least_important_seam(vals):
    dirs = 2D array with same dimensions as vals
    costs = 2D array with same dimensions as vals
    costs[height-1] = vals[height-1] // initialize bottom row of costs

    for row from height-2 to 0:
        for col from 0 to width-1:
            costs[row][col] = vals[row][col] +
                              min(costs[row+1][col-1],
                                  costs[row+1][col],
                                  costs[row+1][col+1])
            dirs[row][col] = -1, 0, or 1 // depending on min

    // Find least important start pixel
    min_col = argmin(costs[0]) // Returns index of min in top row

    // Create vertical seam of size 'height' by tracing from top
    seam = []
    seam[0] = min_col
    for row from 0 to height-2:
        seam[row+1] = seam[row] + dirs[row][seam[row]]

    return seam
```

What is $\mathbf{argmin()}$?

- What does the $\min()$ function do?
  - Returns the minimum output value of a function
- So what about the $\mathrm{argmin}()$ function?
  - Given a function f(x), returns the value of x for which f(x) is minimal.
- Examples:
  - Consider the array $L = [5, 4, 1, 3, 9]$
    - $\min(\mathrm{L}) = 1$
    - $\mathrm{argmin}(\mathrm{L}) = 2$ ← The index of the minimum value

How to do **argmin** in MATLAB

- L = [5 4 1 3 9];
- b = min(L);
- [b c] = min(L);

Which one is argmin?

More Applications: Image Enlarging

- Seam Carving can be used to enlarge an image.
- To this end, least important seams are added.

More Applications: Object Removal

- User marks the target object to be removed.
- Then seams are removed from the image until all marked pixels are gone.
- To regain the original size, seam insertion is employed.

More Applications: Object Removal

- Find the missing shoe…

Q&A