CS 4800: Algorithms & Data

Lecture 9

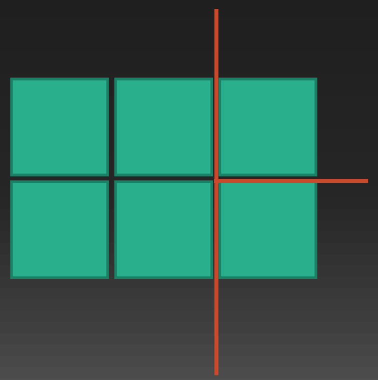
February 6, 2018

Blueprint of DP in knapsack

Subproblems	Best(i, C): max value for sack size C with items {i,i+1, …, n}
Guess	Whether pick item i or not
Recurrence	$Best(i, C) = \max(v_i + Best(i + 1, C - w_i), Best(i + 1, C))$
Order	Compute Best(i,C) for i from n down to 1
Solve orig. problem	Return Best(1, W)

Chompo bar

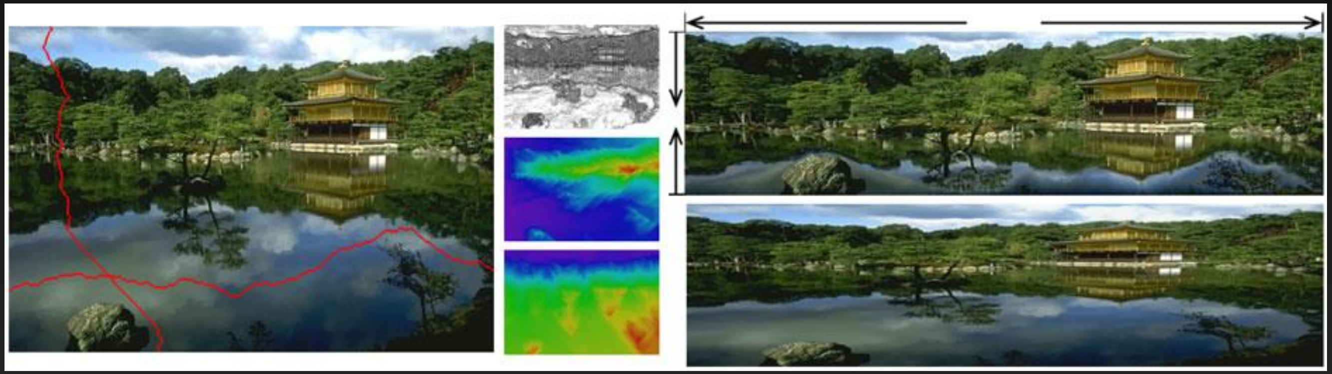
- Given an n x m bar
- Wants to break into perfect squares with minimum number of cuts
- With each cut, can split the bar horizontally or vertically
- 2 x 3
  - -> (2 x 2, 2 x 1)
  - -> (2 x 2, 1 x 1, 1 x 1)



Blueprint of DP in chompo bar

Subproblems	C(j,k) : number of cuts to divide a j x k bar into perfect squares
Guess	First cut
Recurrence	
Order	
Solve orig. problem	

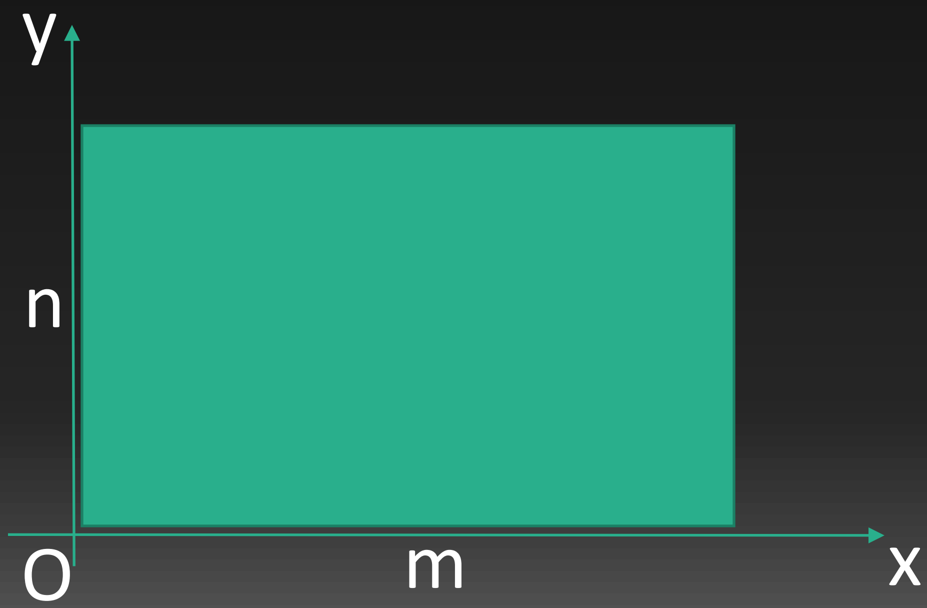
Seam carving



Avidan, Shamir SIGGRAPH’07

Which seam to delete?

- Define energy function ~ amount of information for each pixel
- Find minimally informative seam and delete



Pixel energy

- Energy = How much color variation around (x_0, y_0)
- Energy = Magnitude of gradient

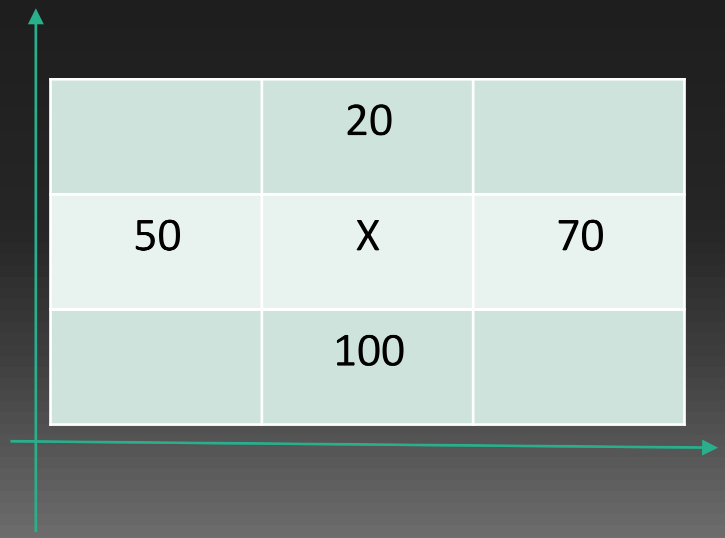
$$e(x_0, y_0) = e_x(x_0, y_0) + e_y(x_0, y_0)$$

where

$$e_x(x_0, y_0) = |Img(x_0 + 1, y_0) - Img(x_0 - 1, y_0)|$$

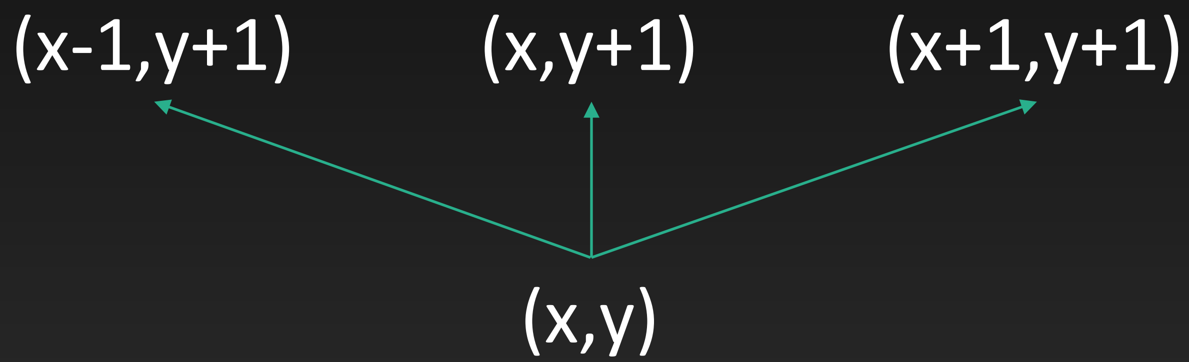
$e_x(x_0, y_0)$=20

$e_y(x_0, y_0)$=80



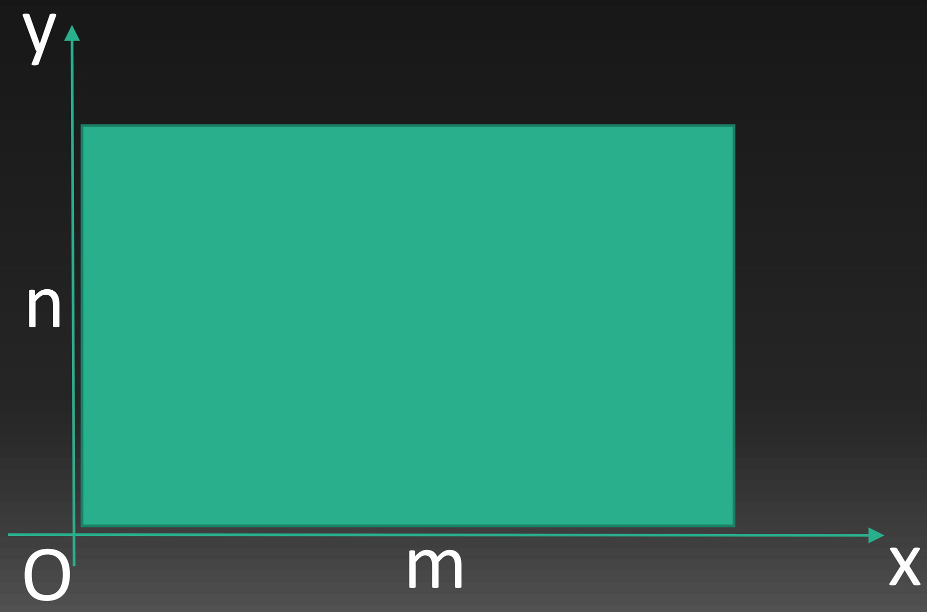
Find least energy vertical seam

- A vertical seam goes from bottom edge to top edge



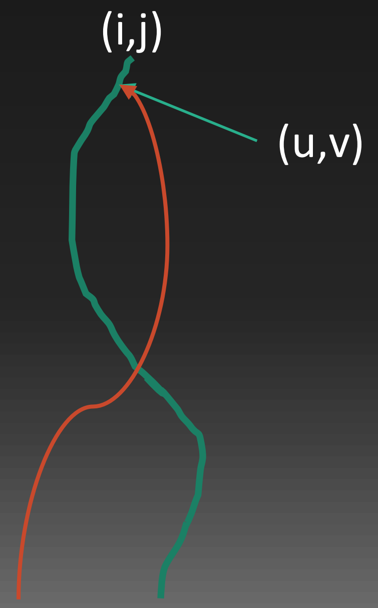
Identify subproblems

- S(x,y): least energy to go from bottom edge to (x,y)
- Best seam to be deleted has to be the one with least energy among S(1,n), S(2,n), …, S(m,n)



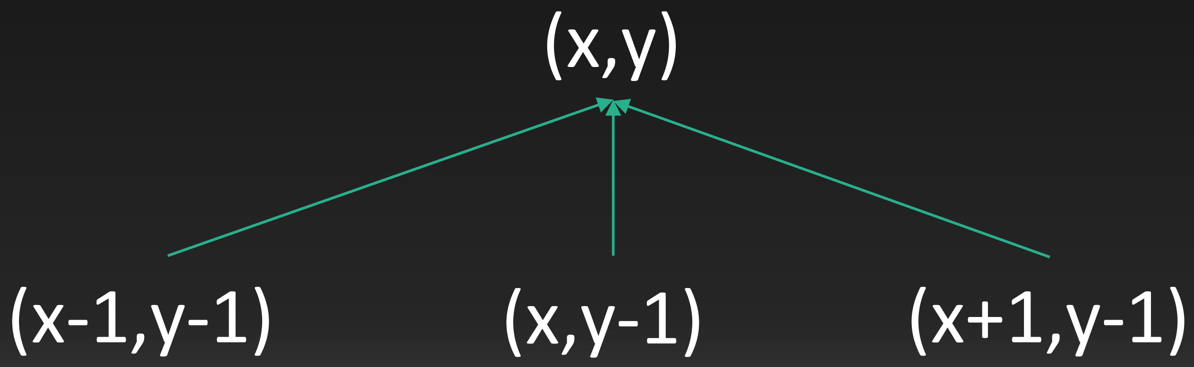
Optimal substructure

- Consider the optimal seam to reach (i,j)
- Consider the last point (u,v) on the seam before reaching (i,j)
- Path from bottom to (u,v) must also be optimal



Recursive relation

- $S(x,y) = e(x,y) + min \begin{cases} S(x-1, y-1) \\ S(x, y-1) \\ S(x+1, y-1) \end{cases}$



Evaluation order

- Initialize $S[x, 0] \leftarrow 0\ \forall x$ // base cases
- Initialize $S[0, y] \leftarrow \infty, S[m+1, y] \leftarrow \infty$ // sentinels
- For $y \leftarrow 1\ to\ n,$ // go from bottom to top
  - For each x in {1,…,m}

$$S(x, y) \leftarrow e(x, y) + min \begin{cases} S(x-1, y-1) \\ S(x, y-1) \\ S(x+1, y-1) \end{cases}$$

Compute final answer

- Pick best among all end points on the top row
- $best\_x \leftarrow 1$
- For x from 1 to m
  - If $S(x, n) \ < \ S(best\_x, n)$
    - $best\_x \leftarrow x$
- Return $S(best\_x, n)$

Trace back the seam

- Initialize $S[x, 0] \leftarrow 0 \; \forall x$ // base cases
- Initialize $S[0, y] \leftarrow \infty, S[m+1, y] \leftarrow \infty$ // sentinels
- For $y \leftarrow 1 \; to \; n,$
  - For each x in {1,…,m}
    - $S(x, y) \leftarrow e(x, y) + min \begin{cases} S(x-1, y-1) \\ S(x, y-1) \\ S(x+1, y-1) \end{cases}$
    - $choice(x, y) \leftarrow either \; x-1, x, or \; x+1 \; depending \; on \; which \; one \; is \; the \; minimum \; above$

Trace back the seam

- $x \leftarrow best\_x$
- $for\ y \leftarrow n\ downto\ 1$
  - Output point (x, y)
  - $x \leftarrow choice[x, y]$

Text justification

Packing words into lines

- Sequence of words $w_1, \ldots, w_n$
- w_i is the width of i-th word
- Want to pack words into lines in the most aesthetically pleasing way

Rule 1: no text in margin

The quick brown fox jumps over the lazy dog

Rule 2: avoid slack if possible

O, they have lived long on the alms-basket
of words. I marvel thy master hath not
eaten thee for a word; for thou art not so
long by the head as
honorificabilitudinitatibus: thou art easier
swallowed than a flap-dragon.

Slack

Objective

- Partition words into lines so that
  - Total width of each line is at most page width W

$$(\#words\ on\ line\ i\ ) - 1 + \sum_{j\ on\ line\ i} w_j \leq W$$

Spaces

  - Slack on line i
    $W - (\#words\ on\ line\ i\ ) + 1 - \sum_{j\ on\ line\ i} w_j$
  - Minimize sum of slacks cubed

Identify subproblems

- Best(n): minimum badness for typesetting first n words