

CSC 301: Analysis of Algorithms

Week 12: Final Project

William Rebelsky
[rebelskyw]

Grinnell College

Monday 29th April, 2024

Table of Contents

- 1 Monday: Final Project
 - Final Project Example
 - Final Project
- 2 Wednesday: No Class
- 3 Friday: Flow Redux
 - Flow

CSC 301: Analysis of Algorithms

Class 37: Final Project

William Rebelsky
[rebelskyw]

Grinnell College

Monday 29th April, 2024

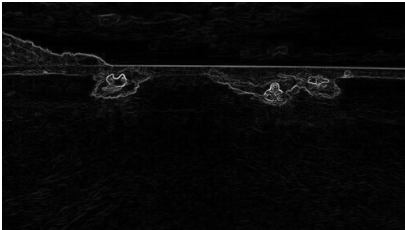
Reminders

- Homework 9 is due Wednesday
- Quiz 9 is due tonight
- I have sent preliminary grades to everyone. Note that the Quiz 3 grade is incorrect and used the wrong threshold
- Schedule:
 - This week:
 - Monday: Final Project description, time to ask questions and work on it in class. Please send me an email with your group by the end of the day. If you did not send me an email and get permission for another group size already, you should work in groups of size 2-3
 - Wednesday: No class, please work on the final project. I will also not be hosting office hours
 - Friday: Graphs and Flow

Final Project Description

- In this project, you will be working on designing and implementing the "seam carving" algorithm
- This is a content-aware image resizing technique unlike traditional scaling techniques
- Discovered in 2007 and popularized in Adobe Photoshop
- References: Code for generating the Gifs from Cooper Sigrist, Nifty assignments

Images



Demonstrations

For a variety of reasons, this doesn't work well in the pdf, so I will be showing them in class.

Final Project Goals

- Given an energy function, an image, and a set of final dimensions, design an algorithm to compute the minimum value vertical seam of the image, remove the seam, and repeat until the current image has the correct final dimensions.
- Prove that it is correct
- Give the runtime
- Implement it in your language of choice (python?)
- Outputs: Energy of the original image, first vertical seam, final image

Energy Computation

$$E(x, y) = \begin{cases} 1000 & (x, y) \text{ is on the border of the image} \\ \sqrt{\sum_{d \in \Delta} d^2} & \text{otherwise} \end{cases}$$

- If the pixel is on the border: set it to 1000
- Otherwise calculate the vertical cost by RGB channel by comparing the pixels above and below it
 - That is, if the pixel of interest is (1,1) compute

$$\Delta_y^2 = (R_{0,1} - R_{2,1})^2 + (G_{0,1} - G_{2,1})^2 + (B_{0,1} - B_{2,1})^2$$
- Calculate the horizontal cost by RGB channel in a similar method
 - That is, if the pixel of interest is (1,1) compute

$$\Delta_x^2 = (R_{1,0} - R_{1,2})^2 + (G_{1,0} - G_{1,2})^2 + (B_{1,0} - B_{1,2})^2$$
- Combine the results and take the square root:

$$E(1, 1) = \sqrt{\Delta_x^2 + \Delta_y^2}$$

- Note that the RGB values of the pixel itself are not used

Table of Contents

- 1 Monday: Final Project
 - Final Project Example
 - Final Project
- 2 Wednesday: No Class
- 3 Friday: Flow Redux
 - Flow

CSC 301: Analysis of Algorithms

Class 38: No class: Work on the Final Project

William Rebelsky
[rebelskyw]

Grinnell College

Wednesday 1st May, 2024

Reminders

- Homework 9 is due tonight
- There is no class today, please work on the Final Project with your partners
- There will be no office hours today

Table of Contents

- 1 Monday: Final Project
 - Final Project Example
 - Final Project
- 2 Wednesday: No Class
- 3 Friday: Flow Redux
 - Flow

CSC 301: Analysis of Algorithms

Class 39: More work on Flow and graphs

William Rebelsky
[rebelskyw]

Grinnell College

Friday 3rd May, 2024

Reminders

- Please Review your grades and work on turning in any outstanding redos as soon as possible
- Please work on your final projects. They are due next Friday. If you need an extension let me know.

Flow Requirements

- Directed Graph
- Each edge has a nonnegative capacity
- Every vertex lies on a path from the source, s , to the sink, t
- There are no reverse or antiparallel edges (i.e. if (u,v) exists, (v,u) does not)
- Flow is a function that satisfies two properties
 - Capacity constraint: The flow along every edge is non negative and not larger than the capacity of the edge
 - Flow conservation: Flow into a vertex equals the flow out of it

Max-Flow, Min-Cut

Theorem 24.6: The following are equivalent

- f is a maximum flow in G
- the residual network G_f has no augmenting paths
- $|f| = c(S, T)$ for some cut (S, T) of G

Ford-Fulkerson

- Set all flows to 0, while there is a path, add flow
- Finding the path is the major contributor to runtime
- If irrational edge capacities: may never terminate
- If rational edge capacities: convert to integer capacities
- If Integer capacities (which is most common): Find an augmenting path in linear time

Example

- Class chooses the following: (likely the people who get to class early)
 - 1 number of vertices
 - 2 number of edges
 - 3 possible edge weights
- I will go through the FF algorithm

Example 2

- Exam Scheduling
 - n classes with individual capacities
 - r rooms with individual capacities
 - t time slots
 - p proctors: can supervise any class, are only available at certain times, can only supervise 5 exams max
- I will go through my thought process

Problems to work through

- In the circulation flow problem, the goal is to determine if there is a way to satisfy everyone's demand
- Given:
 - F factories that can produce f_i goods
 - V villages that each want v_j goods
 - Roads between factories and villages (and each other) that can handle r_k goods going along them
- In the project selection problem, the goal is to maximize revenue
- Given:
 - P projects that have an associated revenue of p_i . Note that p_i may be negative (e.g. buying new equipment)
 - E prerequisites: project v may require project w . For example, you can't sell and support new software without hiring new employees.