CS 7353: Analysis of Algorithms Homework 4: Visible Lines Problem

Noah Schrick

March 8, 2022

Contents

1 Problem Introduction

The Hidden Surface Problem is widely used in computer graphics. When displaying images, not all images should be visible (for example, if one object is in front of another). To avoid overlapping images from being displayed and causing collisions, computations are performed to remove all or parts of an object from the field of vision. This problem is a simplified version focusing on lines in a 2D plane. Rather than handling shifting perspectives, this problem will be further simplified to focus on the perspective as looking down from y=∞.

The problem statement can be defined as follows:

You are given n nonvertical lines in the plane, labeled L_1,...,L_n, with the ith line specified by the equation y=a_ix+b_i. We will make the assumption that no three of the lines all meet at a single point. We say line L_i is uppermost at a given x-coordinate x_0 if its y-coordinate at x_0 is greater than the y-coordinates of all the other lines at x_0: $a_i x_0 + b_i$¿$a_j x_0 b_j$ for all j≠i.

We say line L_i is visible if there is some x-coordinate at which it is uppermost-intuitively, some portion of it can be seen if you look down from ”y=∞”.

Determine which lines are visible and at which restricted domains.

2 Program Preface

This assignment was given with instruction to implement this in C++, or justify why Python should be used instead. This problem was solved using C++ on a Linux system. Attached with submission is a zip folder that contains:

- A CMakeLists.txt file for compiling
- An ”images” folder that contains:
  1. Various images included in this report
- A ”src” folder that contains:
  1. A Line.cpp and Line.h file for the Line class and associated functions
  2. A HiddenLines.cpp and HiddenLines.h file for the problem instance class and associated functions
  3. A GNUPlot .hpp file for plotting
  4. The main file
- A ”data” folder that contains:
  1. A CSV file for defining the line functions to use
- A ”build” folder that contains:
  1. A build.sh script to simplify the build process
  2. A run.sh script to simplify running the program
  3. Various CMake files
  4. The compiled binaries for the program and associated libraries

This program requires the installation of GNUPlot for plotting purposes, but modification can be performed to remove the plotting. Removing the plotting does not alter the solution computation. In addition, this program promises no guarantee of working on other Operating Systems - no testing was conducted on any other platform besides on the local Linux machine.

3 Coding Approach

3

4 Results

4