Project for CISC-868

David Rappaport

Fall 2011

1 Introduction

This document describes the rules of the game for the written project for 2011 Fall term instance of CISC-868. It can also be used as an exemplar for formatting with LaTeX.

2 The learning outcomes

The intent of this project is to provide you with an opportunity to explore the computational geometry literature and to gain an appreciation for how results are presented. For the project you will have to do the following .

- Read some (3-5) technical papers in the field of computational geometry.
- Understand what you are reading, at a level where you at least know what the paper is about, and appreciate the contribution. You may not understand every nuance of every proof of the paper. That is acceptable. However, you should at least understand the point of every theorem and/or lemma and/or algorithm and be able to explain what it means to a classmate.
- Write a report based on the papers that you read. The papers should be related in way that you make clear in your report. The report should be written so that someone in this class can understand it.

3 Content of Report

The report should be written at a level so that your peers can read and understand it. You may consider one of your classmates in this course. It would even be better to consider someone in the computing program who hasn't necessarily taken this course. A well written report will have the following properties:

- It should be grammatically correct, with a minimal number of typographic errors or spelling mistakes.
- It should be self contained, with appropriate definitions, and clear notation so that the reader is capable of reading through the paper without having to refer to secondary sources.
- It should be technically correct. The terminology and notation should be congruent with standard practice in the field.
- It should describe in a comprehensive way the results presented in 3-5 papers that you have read. You may include figures and/or tables in your report if the would help understanding the results. However, you must do these on your own and not just copy them from the articles that you have read. For example your own original illustrative example, using figures, for the way an algorithm works is acceptable. Copying the figures that illustrate the algorithm directly from the paper is not.

4 Rules of the Game and Due Dates

Step 1 Pick the papers that you will read. All of your choices must be taken from the proceeding of the Canadian Conference on Computational Geometry (CCCG). Please see `http://www.cccg.ca` the web site for CCCG. The advantages of restricting your choice to this single source is that it assures that you will stay within the scope of the course. The papers in these proceedings are short and in any given year of the same length. There are eighteen year of proceedings on line so there should be plenty of papers to choose from. In your references you may cite some additional papers to the ones that you are reporting on. You may consult me on whether the papers that you have chosen to read. You need to submit the list of papers that you will be reading for your report no later than October 17.

Step 2 Write your report. You will be required to use LaTeXand the source file for this document as a template to submit your report. There will be strict page limit. Your report should be six pages including figures and references. The sixth page of your report may be very short or completely full, that will be the only allowable variance.The submission of your report will be due by midnight of November 27. Please note that all submissions will be done electronically. Reports should be submitted as PDF.

5 Other Considerations

This section is to included to provide you with some examples of how to use LATEX. Heres a link to a wealth of information on how to typeset a document using LaTeX. `http://www.latex-project.org/guides/`. I can also cite this reference in the bibliography [**?**].

If you will be using drawn figures this is how to do it.

I think that you will agree that Figure 1 is very striking.

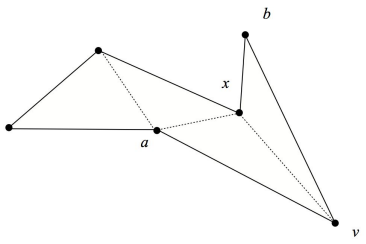

Figure 1: Every triangulations of an n-gon has $n-2$ triangles and $n-3$ diagonals.

If you will include lemmas and proofs here is how to do it.

Lemma 1. *Every polygon with three or more vertices has a diagonal.*

Proof. The proof is in our textbook. □

Or you may want to include a theorem to complement Lemma 1.

Theorem 1. *Every polygon can be triangulated.*

Proof. The proof follows directly from Lemma 1 □

It may be useful to describe the algorithms that you read about using a nice algorithm formatter like algorithm2e. You can find more information on how to use this package on the web `http://texcatalogue.sarovar.org/entries/algorithm2e.html` I use this package to format the algorithms in the notes that I distribute to you. Like the algorithm shown below.

Algorithm 1: SlowConvexHull.

Input: A set of points $P = \{p_1, p_2, \ldots, p_n\}$ in the plane. Note the points are given in no particular order.

Output: A set CH containing the extreme points of P in no particular order.

```
1 CH ← ∅ ;
2 for all ordered pairs (p, q) ∈ P × P with p ≠ q do
3 |   valid ← true;
4 |   for all points r ∈ P, r ≠ p ≠ q do
5 |   |   if r lies to the left of the directed line from p to q or if p, q, r are collinear and r is
  |   |   not between p and q then valid ← false;
6 |   end
7 |   if valid then CH ← CH ∪ {p, q}.;
8 end
9 return CH
```

You will also have to include references in your report. For example here is how to provide a citation [1].

References

[1] LaTeX documentation `http://www.latex-project.org/guides/` accessed on October 3, 2011.

[2] Joseph O'Rourke. *Computational Geometry in C second edition,* Cambridge University Press, 1998.