

Vorosweep, a Fast Approximate Generalized Voronoi Diagram generator

for version 0.1, September 29 2014

Thibaud Mouton (thibaud.mouton@gmail.com)
Eric Bechet (eric.bechet@ulg.ac.be)

This manual is for Vorosweep (version 0.1, September 29 2014), a Fast Approximate Generalized Voronoi Diagram generator.

Copyright © 2014 Free Software Foundation, Inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Table of Contents

1	Introduction.....	1
2	License.....	2
3	Disclaimer.....	3
4	Preriquisites.....	4
5	Files tree.....	5
6	Compiling Vorosweep.....	6
7	Using Vorosweep.....	7
8	Important notes.....	8
9	History.....	9

1 Introduction

Vorosweep is an algorithm written in C++ which is the reference implementation of a generalized Vorono diagram algorithm in two dimension. This algorithm consists in a sweep of the plane by a wavefront from each generator sites associated to an arbitrary convex metric.

To generate a diagram, a set of data points in 2d is given. These points are preprocessed into a hashing data structure in such a way that collisions between wavefronts are only computed locally. As a consequence, only a limited amount of these events are finally rejected, resulting in an efficient computation of the diagram. In Vorosweep, the distance between two points is mainly not symmetrical. Indeed Vorosweep assumes that the distance between two sites is measured by using the local metric associated to the site from which the distance is measured. The class of distance functions implemented are the so called Minkowski metrics. Moreover, anisotropy and orientation can be also specified for each point site. Nevertheless, this range of metric can be easily extended.

To our knowledge, no other implementation allowing to generate a Vorono diagram is as flexible as Vorosweep. It can generate diagram of thousands of sites in a reasonable amount of time, let say 1000 generators per 2-3 sec with a still good accuracy.

The Vorosweep package can produce the data structures of the Vorono diagram under the a triangulated description of each cell as well as graphical outputs.

The Vorosweep source code and documentation is available from the following web page:

<http://www.cadxfem.org/vorosweep>

2 License

Copyright (c) 2013-2014 Thibaud Mouton All Rights Reserved.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser Public License for more details.

A copy of the terms and conditions of the license can be found in LICENSE.txt or online at

<http://www.gnu.org/copyleft/gpl.html>

To obtain a copy, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

3 Disclaimer

The author make no representations about the suitability or fitness of this software for any purpose. It is provided "as is" without express or implied warranty.

4 Prerequisites

Compiling Vorosweep is straightforward with the regular tools that are present in most of the linux distribution. In addition, few other packages are needed for viewing the results.

- a C++ compiler (tested with g++ 4.9).
- the usual C++ standard library headers.
- the cmake configure tool.
- the ccmake GUI (optional).
- the Paraview viewer (vtk file format viewer).
- any Encapsulated Postscript file viewer (gv, okular, ...).

5 Files tree

First, extract the `vorosweep-0.1.tar.gz` archive. A directory called `vorosweep-0.1` is created. It is the root directory and contains the source files of Vorosweep.

The `extras` directory contains stuff to be included and/or linked with :

- the `libboard` package used to generate vector graphics.
- the `geom` include containing basic primitives.
- the `csv` include for reading formatted input files

The `build` directory is the directory where everything will be built.

The `test` directory contains the test case examples.

6 Compiling Vorosweep

Go to the `build` directory then run `ccmake . . .`

- Press `c` 2 times.
- Press `g` to generate the `Makefile`.
- Exit from the `ccmake` GUI by pressing `q`.

Finally launch `make` in the current directory. This will output the `vorosweep` binary in the `out` directory. If you do not need the Postscript output, you can deactivate the graphical output by setting `USE_LIBBOARD` to `OFF` using `ccmake`.

7 Using Vorosweep

Since `vorosweep` is a binary executable, you only need to provide it a set of input points. In the `test` directory, you can find the input examples. They are formatted csv text files containing several informations for each generator site :

- the index of the point (integer).
- the x and y coordinates (double double).
- the starting time of the site (double).
- the angle of the site (double).
- the 2 main speeds of the site (double double).
- the discretization of the site (integer, the higher the more precise).

The `vorosweep` binary is ran by providing a valid input file :

```
[user@host/vorosweep_root_dir/test]$ ../out/vorosweep input.dat
```

The resulting diagram will be output in 2 files :

- a `vorosweep_0.vtk` file that can be viewed with Paraview.
- a `vorosweep_0.eps` file with is an Encapsulated Postscript file.

8 Important notes

Further developments will provide a stable API to make Vorosweep a library with an usable interface. Please often check for new releases since the project is at its earliest development stage

9 History

- Version 0.1 09/27/14 : Preliminary release