CGAL and CGAL arrangements

Efi Fogel

Tel Aviv University

Computational Geometry Seminar
Oct. 28th, 2014
Kaplun
Outline

1 **Cgal News**
   - Administration
   - Content

2 **News About Cgal Arrangement and Derivatives**
   - 2D Minkowski Sums
   - 2D Arrangement on Surfaces (in 3D)
   - 2D Regularized Boolean Set Operations
CGAL Website Features

- Has a responsive layout
- Has pleasant layout
  - “The Magical Number Seven, Plus or Minus Two” is one of the most highly cited papers in psychology.
- Does not have much dynamic content
- Supports editing pages directly on the server
Cgal Website Specification

- use Jekyll
  - A static website creator based on ruby
  - Instantiates a website template with various page contents (e.g., people and FAQ).
  - produces a folder, which can be rsynced with a server
  - does not care about the layout. (It’s a simple text replace processor.)
  - There exist many deployment strategies

- Use bootstrap framework for the layout
  - Provides building blocks to create responsive (mobile first) webpages.
  - Various templates exist.
Cgal Documentation

- Cgal uses Doxygen
- All doxygen code and comments are CC BY-NC-ND
  - Free to copy and redistribute the material in any medium or format

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes.

NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material.
Cgal Development Environment

- **Cgal** uses git for revision source control
- **Cgal** may use GitHub
  - Uses Markdown—a lightweight and easy-to-use syntax for styling all forms of writing on the GitHub platform
- **Cgal** uses CMake, a cross platform build system
  - About to move to version 3.0
- **Cgal** will use CTest, a testing tool distributed as a part of CMake
- The master branch will soon become public
Polyhedron Shortest Path

- A new **CGAL** package
- Provides methods to compute the surface-constrained shortest path on 3D a triangulated surface mesh
- Was developed as a GSoC 2014 project
- Authored by Stephen Kiazyk, supervised by Sebastien Loriot and Eric Colin de Verdiere

The shortest paths computation on a terrain


**Triangulation in Any Dimension**

- A package of CGAL 4.6
- Authored by Olivier Devillers, Clement Jamin, and Samuel Hornus

```cpp
#include <CGAL/Triangulation_data_structure.h>
#include <iostream>
#include <vector>

int main()
{
    typedef CGAL::Triangulation_data_structure<CGAL::Dimension_tag<7>> TDS;
    TDS S;
    assert(7 == S.maximal_dimension());
    assert(-2 == S.current_dimension());
    assert(S.is_valid());
    std::vector<TDS::Vertex_handle> V(10);
    V[0] = S.insert_increase_dimension();  // insert first vertex
    assert(-1 == S.current_dimension());

    for(int i = 1; i <= 5; ++i)
    {
        V[i] = S.insert_increase_dimension(V[0]);
        // the first 6 vertices have created a triangulation of the 4-dimensional
        // topological sphere (the boundary of a five dimensional simplex).
        assert(4 == S.current_dimension());
        assert(6 == S.number_of_vertices());
        assert(6 == S.number_of_full_cells());
    }
}
```

*Cgal and Cgal arrangements*
Segment Delaunay Graph Under $L_\infty$

- Voronoi Diagrams of Segments and Points Under the $L_\infty$ Metric
- A new package of Cgal (soon)
- Authored by Panagiotis Cheilaris, Sandeep Kumar Dey, and Evanthia Papadopoulou
- Has applications in VLSI design, in particular in critical area extraction
- Based on the Segment Delaunay graph package of Cgal

The $L_\infty$ segment Voronoi diagram for a set of weakly (left) and strongly (right) intersecting sites.
2D Visibility

- Computes the visibility area within polygonal regions in the plane
- A package of CGAL 4.6
- Authored by Michael Hemmer, Francisc Bungiu, and Kan Huang
- Based on the 2D Arrangements package
- Depends on 2D Segment Delaunay Graphs package
Combinatorial Maps

- A Versatile Framework for Handling Subdivided Geometric Objects
- A generalization of the halfedge data structure to higher dimension
- An existing package of CGAL
  - Authored by Guillaume Damiand and Pascal Lienhardt
  - Authored by Guillaume Damiand and Pascal Lienhardt

Examples

Representations
Outline

1. **CGAL News**
   - Administration
   - Content

2. **News About CGAL Arrangement and Derivatives**
   - 2D Minkowski Sums
   - 2D Arrangement on Surfaces (in 3D)
   - 2D Regularized Boolean Set Operations
Cgal Arrangements in Numbers

265 288 319 files of C++ code (headers)
137,080 155,853 172,922 lines of C++ code (headers)
527 538 646 pages of manual
9 9 3 developers: Michael, Eric, and Efi
7 7 7 packages (AOS/SL, BSO, MS, 2D & 3D Env., SR, LER)

AOS 2D Arrangements on Surfaces
SL 2D Intersection of Curves (Sweep Line)
BSO 2D Regularized Boolean Set Operations
MS 2D Minkowski Sums
SR 2D Snap Rounding
LER 2D Largest Empty Iso Rectangles
**Cgal Arrangements Code in Numbers**

<table>
<thead>
<tr>
<th>Count Package</th>
<th>C++ Files</th>
<th>C++ Lines</th>
<th>Doc. Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>older</td>
<td>old</td>
<td>new</td>
</tr>
<tr>
<td>AOS/SL</td>
<td>180</td>
<td>203</td>
<td>224</td>
</tr>
<tr>
<td>BSO</td>
<td>44</td>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>MS</td>
<td>18</td>
<td>18</td>
<td>27</td>
</tr>
<tr>
<td>2D Env.</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>3D Env.</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>SR</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>LER</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>265</td>
<td>288</td>
<td>319</td>
</tr>
</tbody>
</table>
Minkowski Sums

- Added new methods that handle polygons with holes
- Convolution Approach: Reduced Convolution
  - Based on Alon’s thesis
  - Authored by Sebastian Morr Suppervised by Michael Hemmer
- Decomposition Approach
  - Convex decomposition based Vertical on decomposition
  - Convex decomposition utilizes constrained triangulation
Polycurves

- A traits class of the *2D Arrangements* package, which supports continuous piecewise-any curves
- A generalization of polylines (continuous piecewise-linear curves)
  - Enhanced by Dror Atariah
  - Prepared for further extensions
  - Polyline can be directed left to right or right to left
- Being enhanced by Waqar Khan (a student of Eric)
  1. Supports unbounded polylines
  2. Supports poly-geodesic-arcs on the sphere
  3. Supports continuous piecewise-any curves
- Will be used to approximate certain algebraic curves on the torus, which arise in the configuration space of a two-link revolute-revolute planar robotic arm.
Point Location and Observers

typedef CGAL::Exact_predicates_exact_constructions_kernel Kernel;
typedef CGAL::Env_plane_traits_3<Kernel> Traits_3;
typedef CGAL::Envelope_diagram_2<Traits_3> Envelope_diagram_2;
typedef CGAL::Arr_landmarks_point_location<Envelope_diagram_2::Base::Base> Landmarks_pl;

template <typename GeomTraits, typename Dcel = Envelope_3::Envelope_pm_dcel<GeomTraits, typename GeomTraits::Xy_monotone_surface_3> >
class Envelope_diagram_2 :
    public Envelope_diagram_on_surface_2<GeomTraits, typename Default_planar_topology<GeomTraits, Dcel>::Traits>
{ ... };
Point Location and Observers

Old

template <typename Arrangement_,
        typename Nearest_neighbor_ =
        Arr_landmarks_nearest_neighbor<Arrangement_> >
class Arr_landmarks_generator_base : public Arr_observer<Arrangement_> {
}

New

template <typename Arrangement_,
        typename Nearest_neighbor_ =
        Arr_landmarks_nearest_neighbor<Arrangement_> >
class Arr_landmarks_generator_base : public Arrangement::Observer {
}
Arrangements on Surfaces
Arrangement Demo

- Enhanced by Alex Tsui as part of the Google Summer of Code 2014 program.
- Supports Polylines, Circular arcs, Bezier curves, and algebraic curves.
- The algebraic support includes a polynomial parser
  - Implemented with Boost.Spirit
  - Handles multivariate
  - Handles non-integer
Boolean Set Multiway Operations

- Supports the computation of the union or the intersection of a range of polygons.

```cpp
template <typename InputIterator, typename OutputIterator>
OutputIterator intersection (InputIterator begin, InputIterator end,
                               OutputIterator oi)
{ ... }
```

```cpp
template <typename InputIterator1, typename InputIterator2, typename OutputIterator>
OutputIterator intersection (InputIterator1 pgn_begin1, InputIterator1 pgn_end1,
                               InputIterator2 pgn_begin2, InputIterator2 pgn_end2,
                               OutputIterator oi)
{ ... }
```

- It is optimally tuned to use either the sweep-line framework or a divide and conquer approach.