On Generating Polygons: 
Introducing the Salzburg Database

Günther Eder, Martin Held, Steinþór Jasonarson, 
Philipp Mayer, and Peter Palfrader

EuroCG 2020

Würzburg, March 2020
What is the Salzburg Database?

**Keystones**

- A repository of polygonal areas
- Can be used freely
- Database: [https://sbgdb.cs.sbg.ac.at/](https://sbgdb.cs.sbg.ac.at/)
- Generators: [https://github.com/cgalab](https://github.com/cgalab)
- Currently contains 11,507 instances
What is the Salzburg Database?

Keystones

- A repository of polygonal areas
- Can be used freely
  - Database: https://sbgdb.cs.sbg.ac.at/
  - Generators: https://github.com/cgalab
  - Currently contains 11507 instances
What is the Salzburg Database?

Keystones

- A repository of polygonal areas
- Can be used freely
- Database: [https://sbgdb.cs.sbg.ac.at/](https://sbgdb.cs.sbg.ac.at/)
- Generators: [https://github.com/cgalab](https://github.com/cgalab)
- Currently contains 11,507 instances
What is the Salzburg Database?

**Keystones**

- A repository of polygonal areas
- Can be used freely
- Database: [https://sbgdb.cs.sbg.ac.at/](https://sbgdb.cs.sbg.ac.at/)
- Generators: [https://github.com/cgalab](https://github.com/cgalab)
- Currently contains 11507 instances
**What is the Salzburg Database?**

**Keystones**
- A repository of polygonal areas
- Can be used freely
- Database: https://sbgdb.cs.sbg.ac.at/
- Generators: https://github.com/cgalab
- Currently contains 11,507 instances
How to use it?

**Browser**
Per instance via [https://sbgdb.cs.sbg.ac.at/db/](https://sbgdb.cs.sbg.ac.at/db/)
How to use it?

Browser
Per instance via https://sbgdb.cs.sbg.ac.at/db/

Whole Repository

```
git clone https://sbgdb.cs.sbg.ac.at/db/.git

git annex get
```
How to use it?

**Browser**
Per instance via [https://sbgdb.cs.sbg.ac.at/db/](https://sbgdb.cs.sbg.ac.at/db/)

**Whole Repository**
```
$ git clone https://sbgdb.cs.sbg.ac.at/db/.git
$ git annex get
```
What’s the Format?

Requirements
- Can be parsed and stored easily
- Supports the basic geometric types
- Can be extended to support various properties
- A human should be able to read it?
What’s the Format?

Requirements

- Can be parsed and stored easily
- Supports the basic geometric types
  - Can be extended to support various properties
- A human should be able to read it?
What’s the Format?

Requirements

- Can be parsed and stored easily
- Supports the basic geometric types
- Can be extended to support various properties
- A human should be able to read it?
What's the Format?

Requirements

- Can be parsed and stored easily
- Supports the basic geometric types
- Can be extended to support various properties
- A human should be able to read it?
What’s the Format?

Requirements

- Can be parsed and stored easily
- Supports the basic geometric types
- Can be extended to support various properties
- A human should be able to read it?

GraphML to the rescue!
What’s the Format?

Requirements

- Can be parsed and stored easily
- Supports the basic geometric types
- Can be extended to support various properties
- A human should be able to read it?

Properties

- XML – format
  - Supports graphs in general
  - Directed-, undirected-, mixed-, and hyper-graphs
  - Supports edge-weights
What’s the Format?

Requirements

- Can be parsed and stored easily
- Supports the basic geometric types
- Can be extended to support various properties
- A human should be able to read it?

Properties

- XML – format
- Supports graphs in general
  - Directed-, undirected-, mixed-, and hyper-graphs
  - Supports edge-weights
What’s the Format?

Requirements

- Can be parsed and stored easily
- Supports the basic geometric types
- Can be extended to support various properties
- A human should be able to read it?

Properties

- XML – format
- Supports graphs in general
- Directed-, undirected-, mixed-, and hyper-graphs
- Supports edge-weights
What’s the Format?

Requirements
- Can be parsed and stored easily
- Supports the basic geometric types
- Can be extended to support various properties
- A human should be able to read it?

Properties
- XML – format
- Supports graphs in general
- Directed-, undirected-, mixed-, and hyper-graphs
- Supports edge-weights
Format-Converter

- https://github.com/cgalab/format-converter
- MIT license
- Written in Python

- Reading and writing .graphml, .ipe, .obj-files
- Reading .line, .poly .site-files
- Additional options for edge-weights
- Adding additional formats is simple.
Converter

Format-Converter

- [https://github.com/cgalab/format-converter](https://github.com/cgalab/format-converter)
- MIT license
- Written in Python

- Reading and writing `.graphml`, `.ipe`, `.obj`-files
- Reading `.line`, `.poly`, `.site`-files
- Additional options for edge-weights
- Adding additional formats is simple.
Converter

Format-Converter

- https://github.com/cgalab/format-converter
- MIT license
- Written in Python

- Reading and writing .graphml, .ipe, .obj-files
- Reading .line, .poly .site-files
- Additional options for edge-weights
- Adding additional formats is simple.
Format-Converter

- https://github.com/cgalab/format-converter
- MIT license
- Written in Python

- Reading and writing .graphml, .ipe, .obj-files
- Reading .line, .poly .site-files
- Additional options for edge-weights
- Adding additional formats is simple.
Format-Converter

- [https://github.com/cgalab/format-converter](https://github.com/cgalab/format-converter)
- MIT license
- Written in Python

- Reading and writing `.graphml`, `.ipe`, `.obj`-files
- Reading `.line`, `.poly` `.site`-files
- Additional options for edge-weights
- Adding additional formats is simple.
Format-Converter
- https://github.com/cgalab/format-converter
- MIT license
- Written in Python

- Reading and writing .graphml, .ipe, .obj-files
- Reading .line, .poly .site-files
- Additional options for edge-weights
- Adding additional formats is simple.
**Format-Converter**

- [https://github.com/cgalab/format-converter](https://github.com/cgalab/format-converter)
- MIT license
- Written in Python

- Reading and writing `.graphml`, `.ipe`, `.obj`-files
- Reading `.line`, `.poly`, `.site`-files
- Additional options for edge-weights
- **Adding additional formats is simple.**
Generators

- **Rpg** — Various heuristics
- **Srpg** — On the integer grid
- **Koch, Sierpinski, Hilbert, and Lebesgue**
- **Fpg** — Triangulation Perturbation
- **Spg** — Sweep-line & 2-Opt
Generators

- **Rpg** — Various heuristics
- **Srpg** — On the integer grid
  - Koch, Sierpinski, Hilbert, and Lebesgue
- **Fpg** — Triangulation Perturbation
- **Spg** — Sweep-line & 2-Opt
Generators

- Rpg — Various heuristics
- Srpg — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- Fpg — Triangulation Perturbation
- Spg — Sweep-line & 2-Opt
Generators

- Rpg — Various heuristics
- Srpg — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- Fpg — Triangulation Perturbation
  - Spg — Sweep-line & 2-Opt
Generators

- Rpg — Various heuristics
- Srgp — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- Fpg — Triangulation Perturbation
- Spg — Sweep-line & 2-Opt
Generators

- $R_{pg}$ — Various heuristics
- $S_{rpg}$ — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- $F_{pg}$ — Triangulation Perturbation
- $S_{pg}$ — Sweep-line & 2-Opt
Generators

- **Rpg** — Various heuristics
- **Srpg** — On the integer grid
- **Koch, Sierpinski, Hilbert, and Lebesgue**
- **Fpg** — Triangulation Perturbation
- **Spg** — Sweep-line & 2-Opt
Generators

- Rpg — Various heuristics
- Srpg — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- Fpg — Triangulation Perturbation
- Spg — Sweep-line & 2-Opt
Generators

- **Rpg** — Various heuristics
- **Srpg** — On the integer grid
- **Koch, Sierpinski, Hilbert, and Lebesgue**
- **Fpg** — Triangulation Perturbation
- **Spg** — Sweep-line & 2-Opt
Generators

- \textit{Rpg} — Various heuristics
- \textit{Srpg} — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- \textit{Fpg} — Triangulation Perturbation
- \textit{Spg} — Sweep-line & 2-Opt
Generators

- **Rpg** — Various heuristics
- **Srpg** — On the integer grid
- **Koch, Sierpinski, Hilbert, and Lebesgue**
- **Fpg** — Triangulation Perturbation
- **Spg** — Sweep-line & 2-Opt
Generators

- Rpg — Various heuristics
- Srpg — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- Fpg — Triangulation Perturbation
- Spg — Sweep-line & 2-Opt
Generators

- $R_{pg}$ — Various heuristics
- $S_{rpg}$ — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- $F_{pg}$ — Triangulation Perturbation
- $S_{pg}$ — Sweep-line & 2-Opt
Generators

- $\text{Rpg}$ — Various heuristics
- $\text{Srpg}$ — On the integer grid
- $\text{Koch, Sierpinski, Hilbert, and Lebesgue}$
- $\text{Fpg}$ — Triangulation Perturbation
  - $\text{Spg}$ — Sweep-line & 2-Opt
Generators

- $R_{pg}$ — Various heuristics
- $S_{rpg}$ — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- $F_{pg}$ — Triangulation Perturbation
  - $S_{pg}$ — Sweep-line & 2-Opt
Generators

- Rpg — Various heuristics
- Srpg — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- Fpg — Triangulation Perturbation
- Spg — Sweep-line & 2-Opt
Generators

- Rpg — Various heuristics
- Srpg — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- Fpg — Triangulation Perturbation
- Spg — Sweep-line & 2-Opt
Generators

- \texttt{Rpg} — Various heuristics
- \texttt{Srpg} — On the integer grid
- \texttt{Koch}, \texttt{Sierpinski}, \texttt{Hilbert}, and \texttt{Lebesgue}
- \texttt{Fpg} — Triangulation Perturbation
- \texttt{Spg} — Sweep-line & 2-Opt
Generators

- \textbf{Rpg} — Various heuristics
- \textbf{Srpg} — On the integer grid
- \textbf{Koch}, Sierpinski, Hilbert, and Lebesgue
- \textbf{Fpg} — Triangulation Perturbation
- \textbf{Spg} — Sweep-line & 2-Opt
Generators

- $\mathbf{Rpg}$ — Various heuristics
- $\mathbf{Srpg}$ — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- $\mathbf{Fpg}$ — Triangulation Perturbation
- $\mathbf{Spg}$ — Sweep-line & 2-Opt
Generators

- Rpg — Various heuristics
- Srpg — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- Fpg — Triangulation Perturbation
- Spg — Sweep-line & 2-Opt
Generators

- \textit{Rpg} — Various heuristics
- \textit{Srpg} — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- \textit{Fpg} — Triangulation Perturbation
- \textit{Spg} — Sweep-line & 2-Opt
Generators

- Rpg — Various heuristics
- Srpg — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- Fpg — Triangulation Perturbation
- Spg — Sweep-line & 2-Opt
Generators

- Rpg — Various heuristics
- Srpq — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- Fpq — Triangulation Perturbation
- Spg — Sweep-line & 2-Opt
Generators

- **Rpg** — Various heuristics
- **Srpg** — On the integer grid
- **Koch, Sierpinski, Hilbert, and Lebesgue**
- **Fpg** — Triangulation Perturbation
- **Spg** — Sweep-line & 2-Opt
Generators

- Rpg — Various heuristics
- Srpg — On the integer grid
- Koch, Sierpinski, Hilbert, and Lebesgue
- Fpg — Triangulation Perturbation
- Spg — Sweep-line & 2-Opt
Instance Classes
Instance Classes
Instance Classes

fpg with holes
Instance Classes
Instance Classes
Instance Classes

2-opt
Instance Classes

2-opt
Instance Classes
Instance Classes
Instance Classes
Instance Classes
Instance Classes
Instance Classes
Instance Classes
Instance Classes
Instance Classes
Instance Classes
Instance Classes
Instance Classes
Summary

Database https://sbgdb.cs.sbg.ac.at/

Format-Converter https://github.com/cgalab/format-converter

Call for Participation

Do you have interesting polygons?

What is missing?
(specific class, property, file format)

Contact
{geder,held,palfrader}@cs.sbg.ac.at