Fully Automatic Visualisation of Overlapping Sets

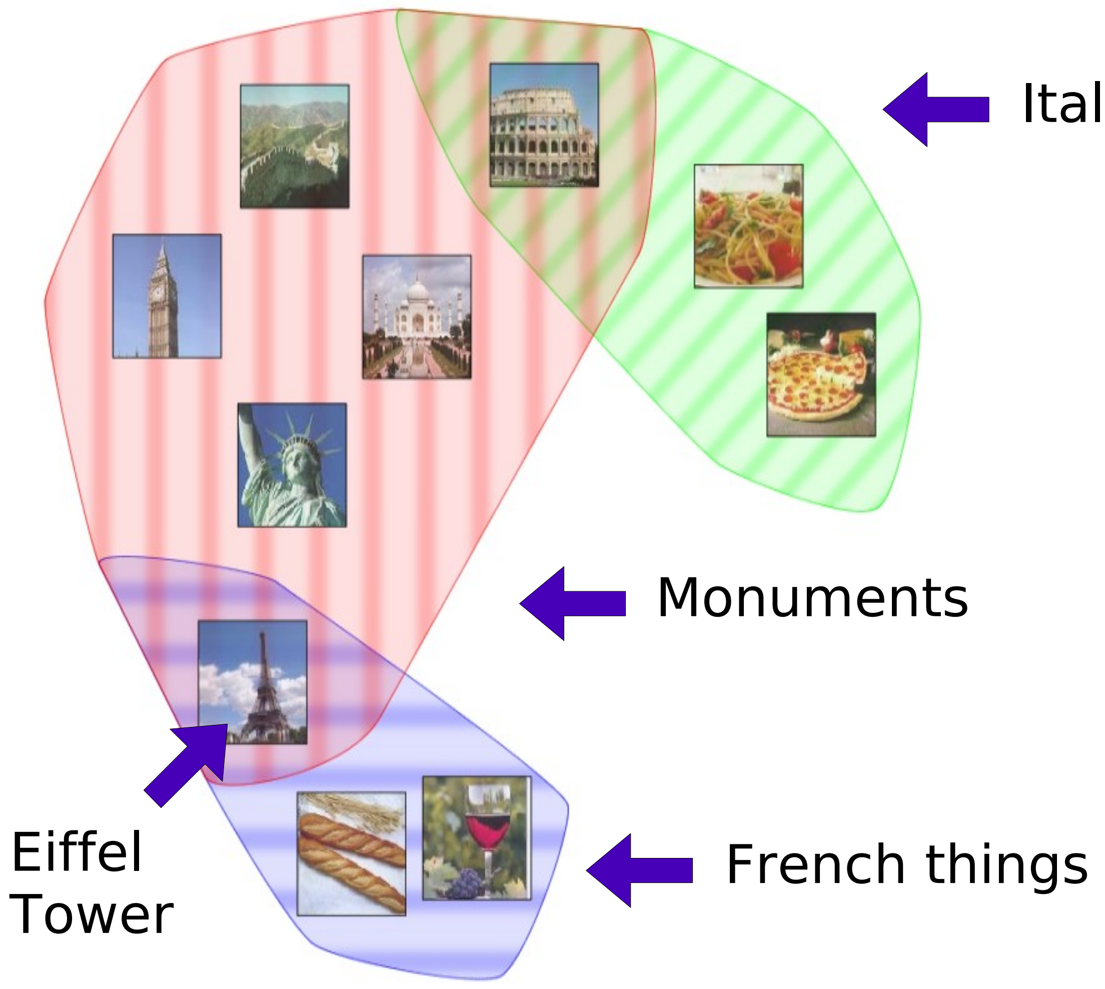
Authors: Paolo Simonetto
David Auber
Daniel Archambault

LaBRI

INRIA Bordeaux Sud-Ouest

Université Bordeaux 1, France

The Problem



- Collection of sets and elements.
- Elements might be shared by different sets.
  - overlapping clusters
  - taxonomies
  - query results
  - ...

How do we represent the sets, elements, and intersections clearly?

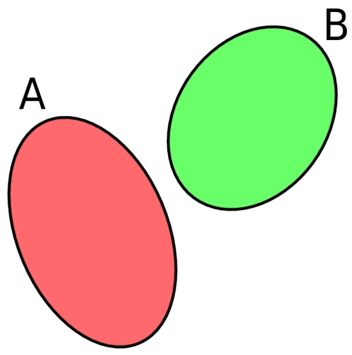
Euler Diagrams

Euler diagrams represent sets and their relationships.

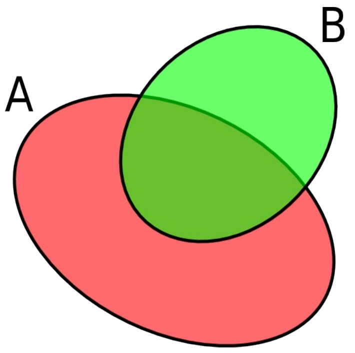
Each closed region is associated with a set.

Overlap, inclusion, and exclusion of regions depict analogue set concepts.

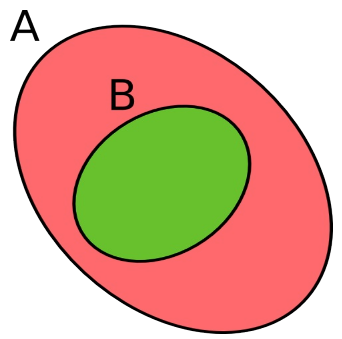
Elements are inserted into regions to depict which set(s) they belong to.



$A \cap B = \emptyset$



$A \cap B \neq \emptyset$
$A \cap B \neq A, B$



$A \supset B$

Euler Diagrams – General definitions

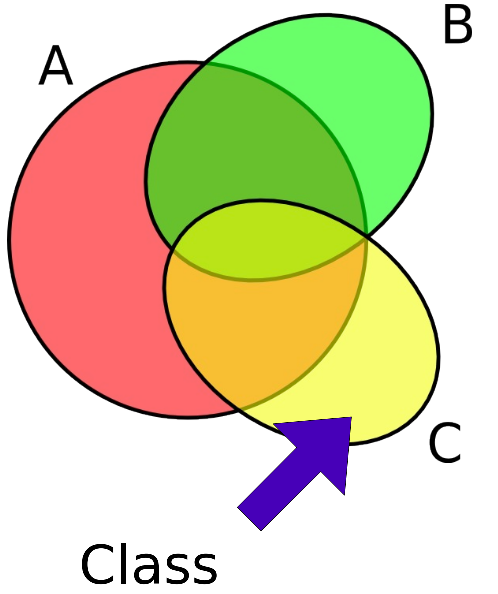
Classes are the sets to be depicted and their visual representations.

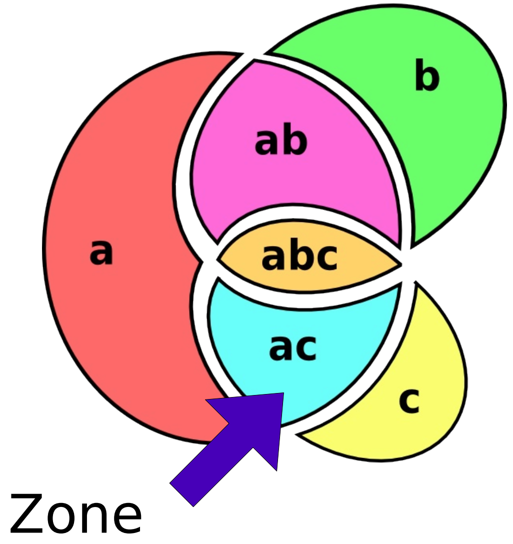
Zones are intersections between classes:

$$Z_S = \left(\bigcap_{C_x \in S} C_x \right) \cap \left(\bigcap_{C_x \notin S} \overline{C_x} \right)$$

S is a subset of classes.

Zones enclose elements that belong to exactly the same subset of classes.





Euler diagram generation

Drawing Euler diagrams is a problem of finding planar embeddings.

Typical approaches construct a planar graph and draw it. In our approach, this graph is an intersection graph.

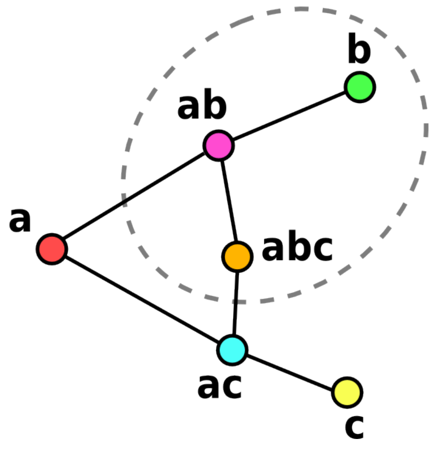
We do not describe the construction of the intersection graph in this work.

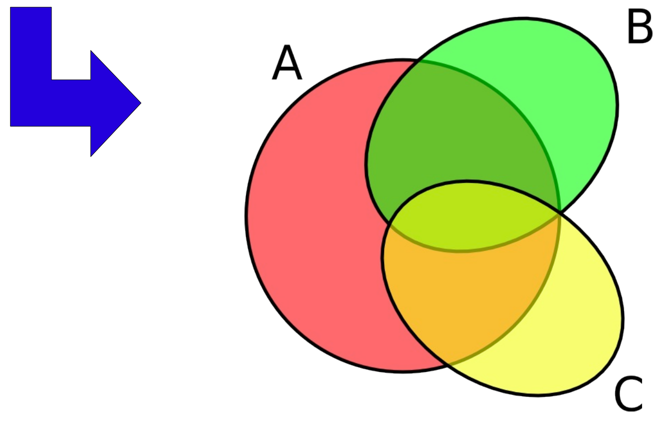
The intersection graph is generated by the algorithm described in:

Paolo Simonetto and David Auber

“An Heuristic for the Construction of Intersection Graphs”

13th International Conference on Information Visualisation (IV09). July 2009, Barcelona.

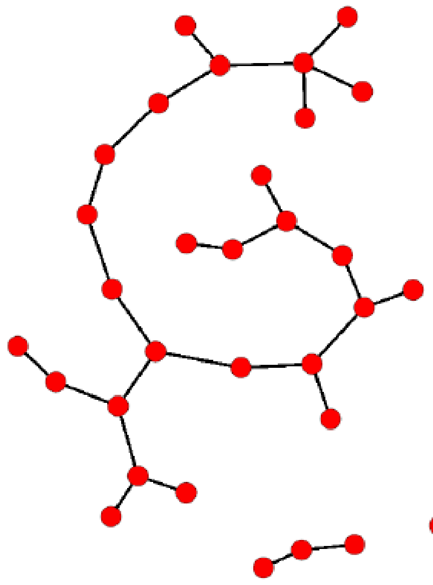




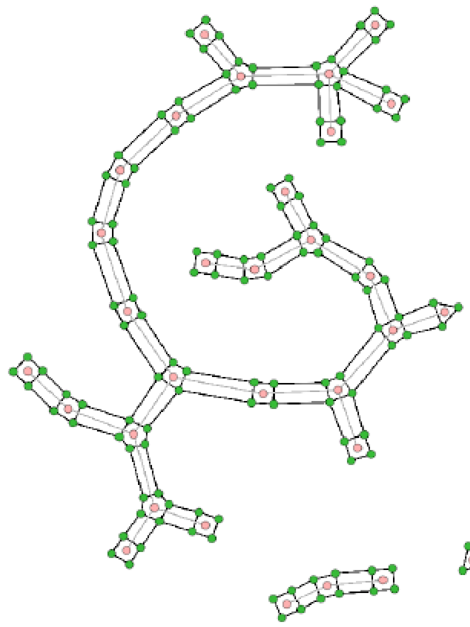
Euler diagram generation

The generation process goes through some intermediate phases:

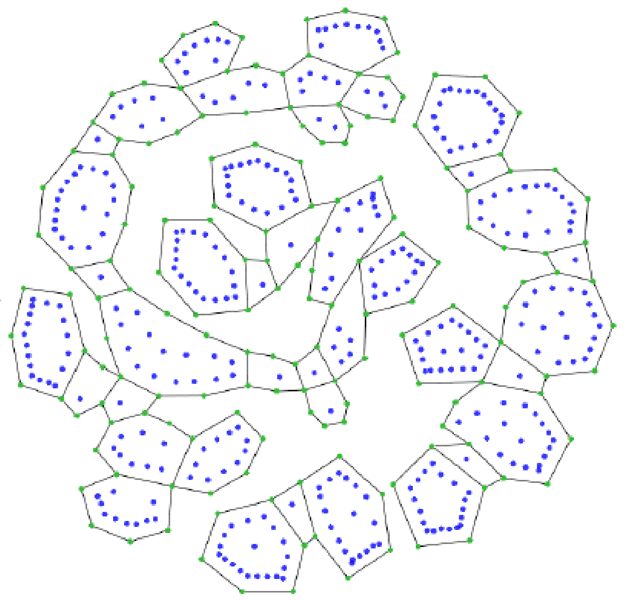
Drawing of the intersection graph



Construction of the grid graph



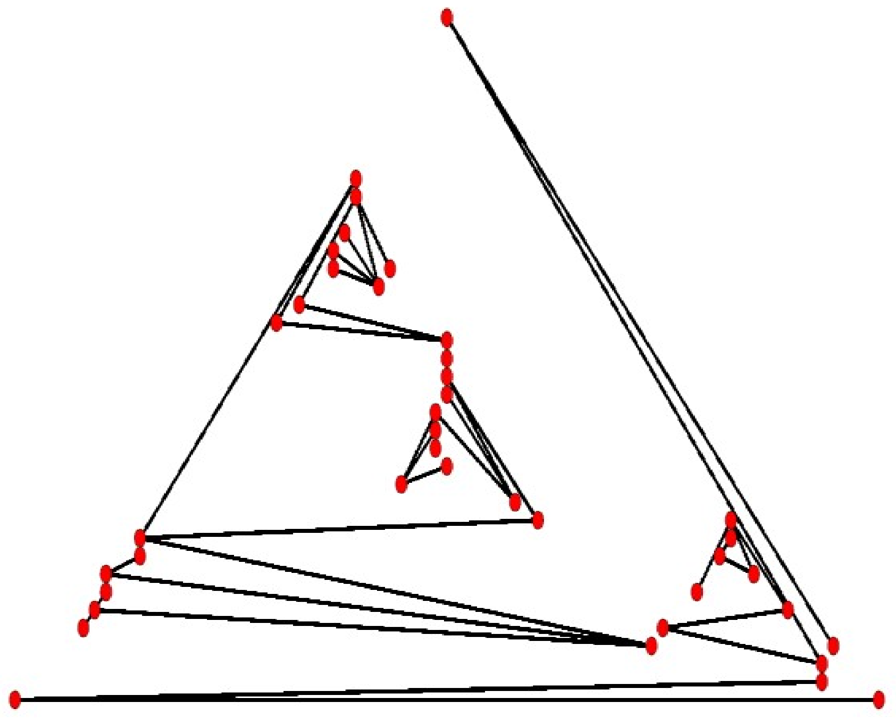
Insertion of the class elements

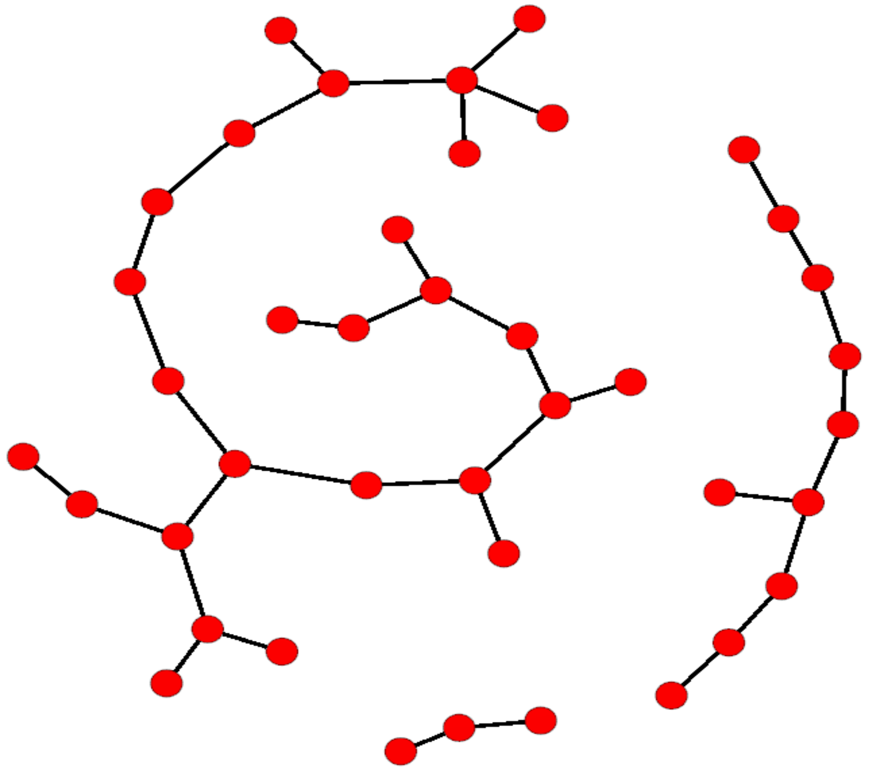


Drawing the intersection graph

An initial planar drawing is computed by the *FPP* algorithm (De Fraysseix, Pach, Pollack, 1990).

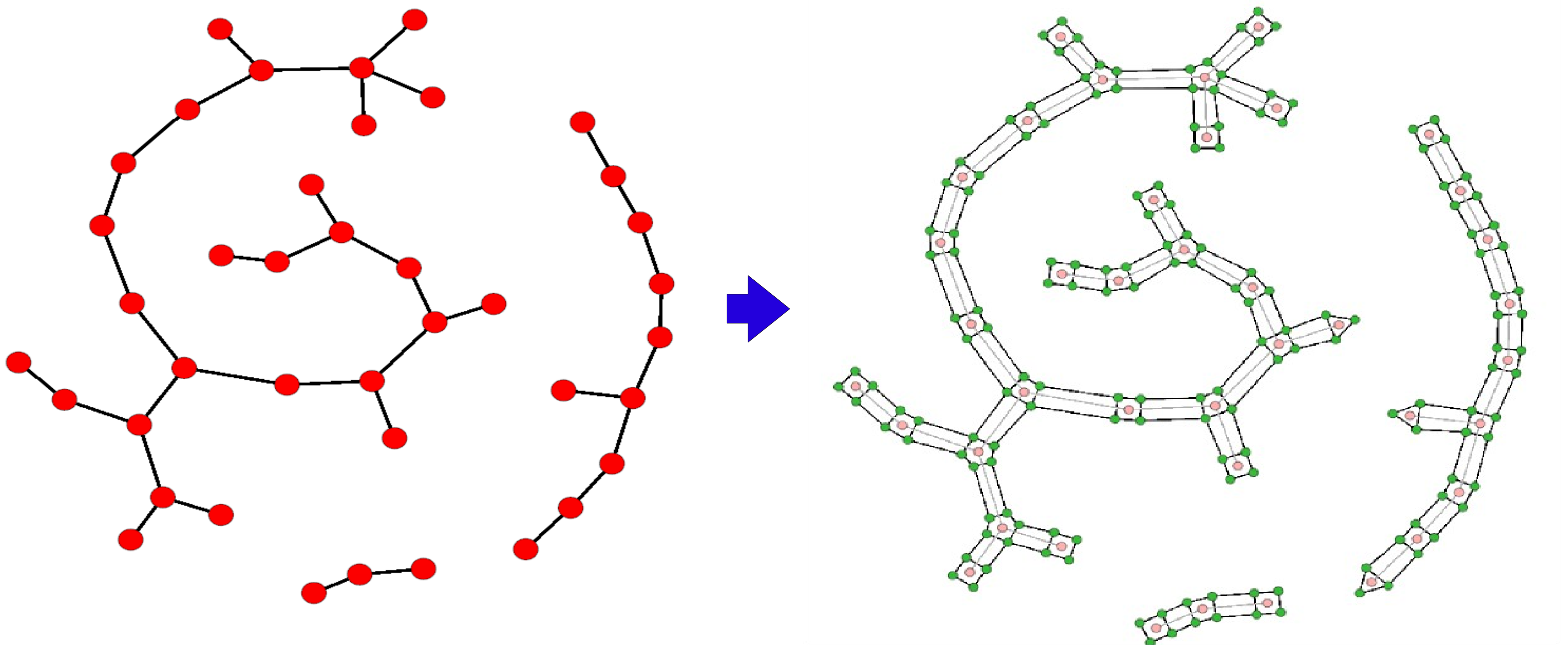
The layout is subsequently improved by *PrEd* (Bertault, 1999).





Grid graph construction - 1

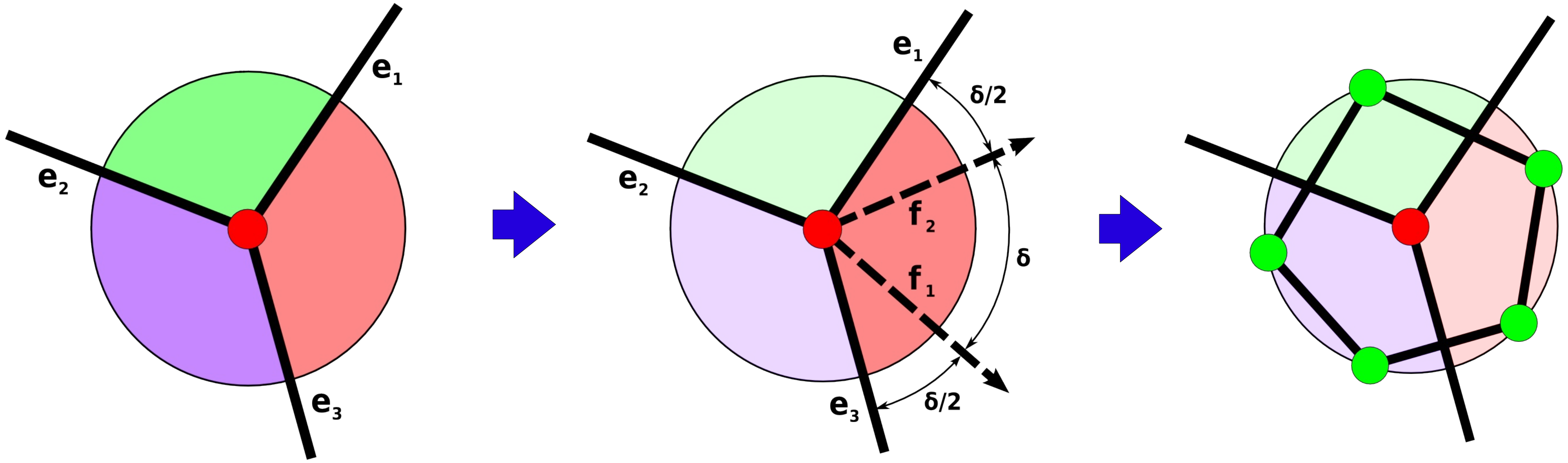
The drawing of the intersection graph is used to construct the grid graph. This graph transforms nodes and edges of the intersection graph into regions.



Grid graph construction - 2

The algorithm constructs simple polygons around each intersection graph node. These polygons are called cells.

Circles of fixed radius are circumscribed around each node. Incident edges to nodes define sectors. One or more grid nodes are placed on the circumference according to the sector angle.



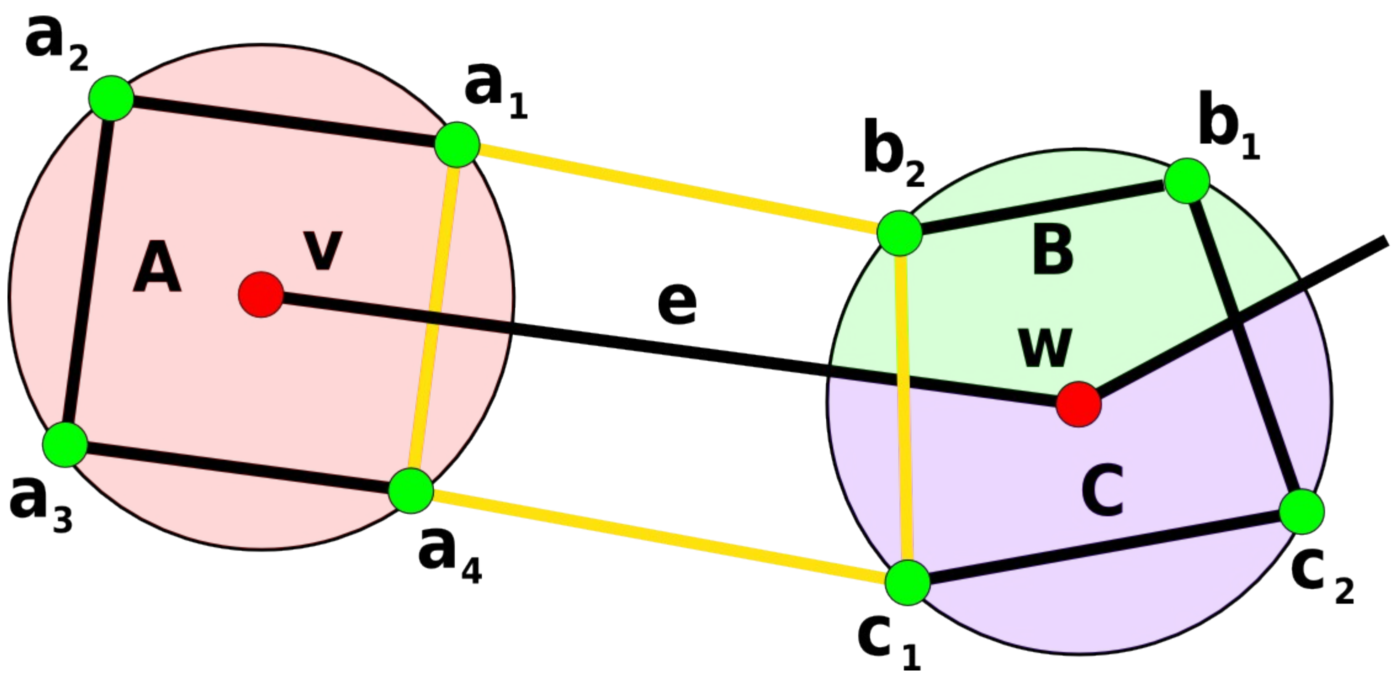
Grid graph construction - 3

The algorithm also constructs regions around intersection graph edges.

Given intersection graph edge, *e*, and its incident nodes, *v* and *w*, we add grid graph edges between:

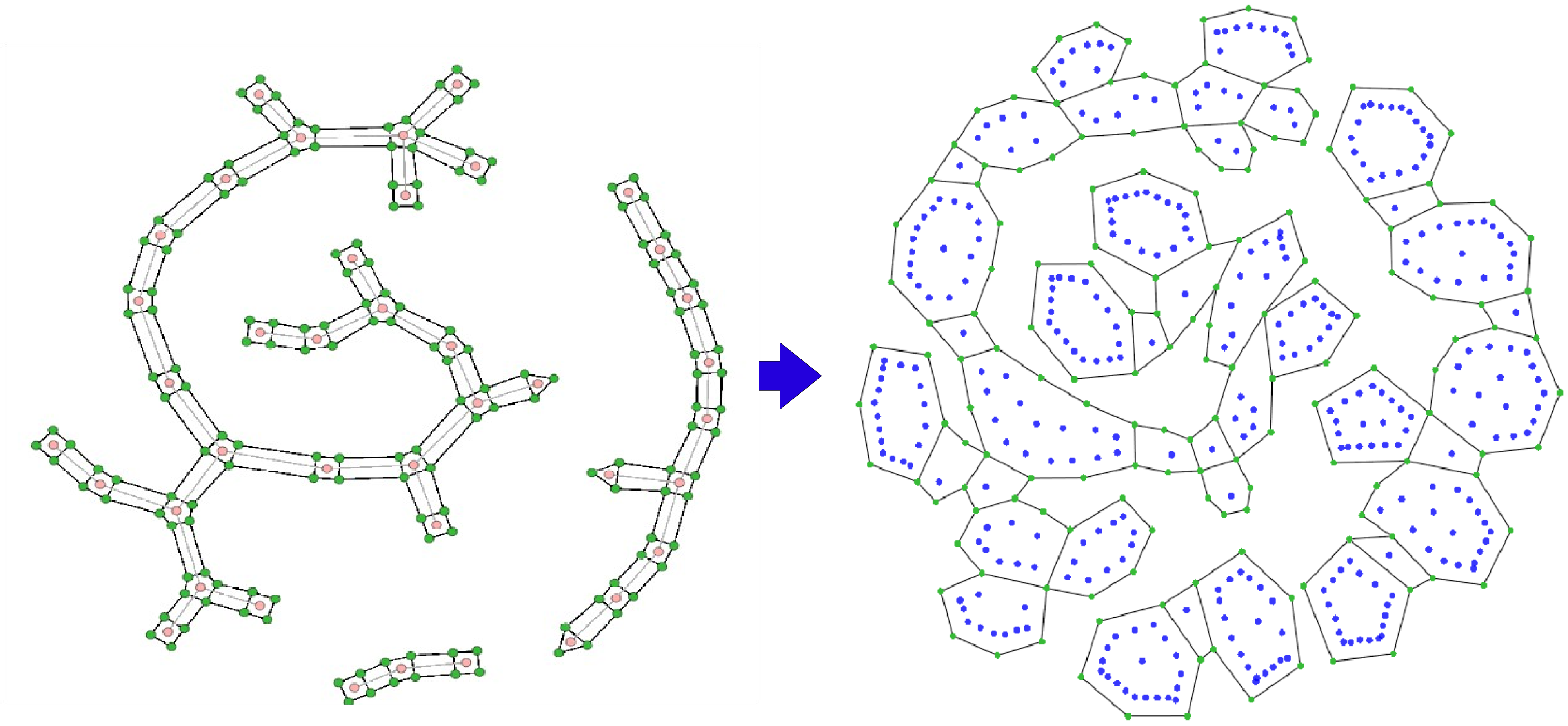
- the first CW node in *v*'s cell and the first CCW node in *w*'s cell

- the first CCW node in *v*'s cell and the first CW node in *w*'s cell



Insertion of the class elements

The algorithm inserts class elements into their corresponding zones. PrEd is applied a second time to expand zones to their appropriate size.



Final diagram

To improve the clarity and the visual appeal of the final diagram, we introduce the following refinements:

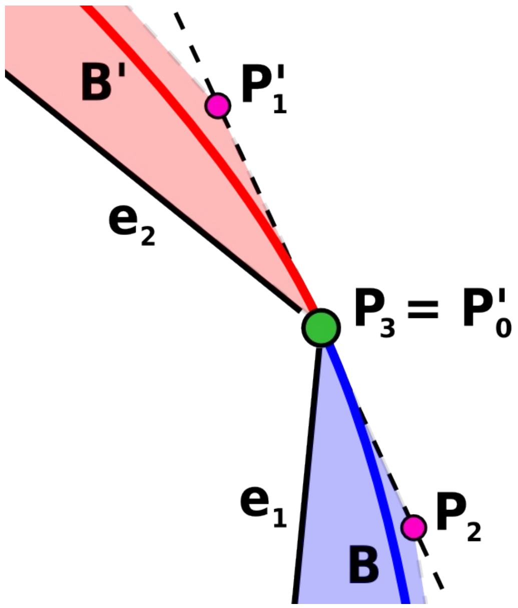
- Bezier curves to model class boundaries
- a colouring policy that minimises the number of necessary colours
- transparent textures that show participating classes in a zone

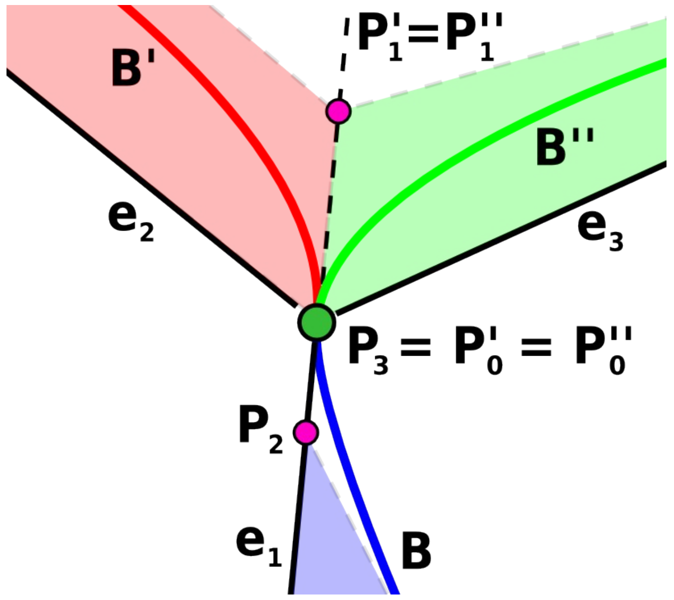
Bézier curves

To compute smooth boundaries, we transform each grid graph edge in a Bézier curve. Our algorithm ensures that:

- Bézier curves join smoothly at incident nodes
- No unwanted crossings between Bézier curves appear

More details can be found in the paper.





Colouring

In complex diagrams, it is difficult to distinguish many colours and the number of possible intersections between classes can be exponential.

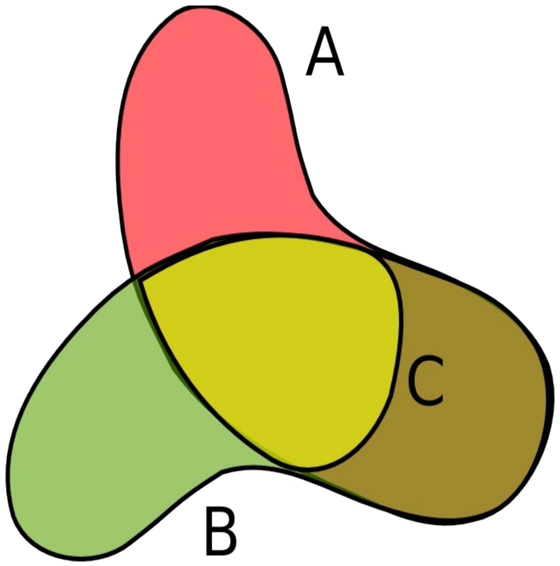
To assign colours to classes, we build a graph where each node is a class and edges connect overlapping classes.

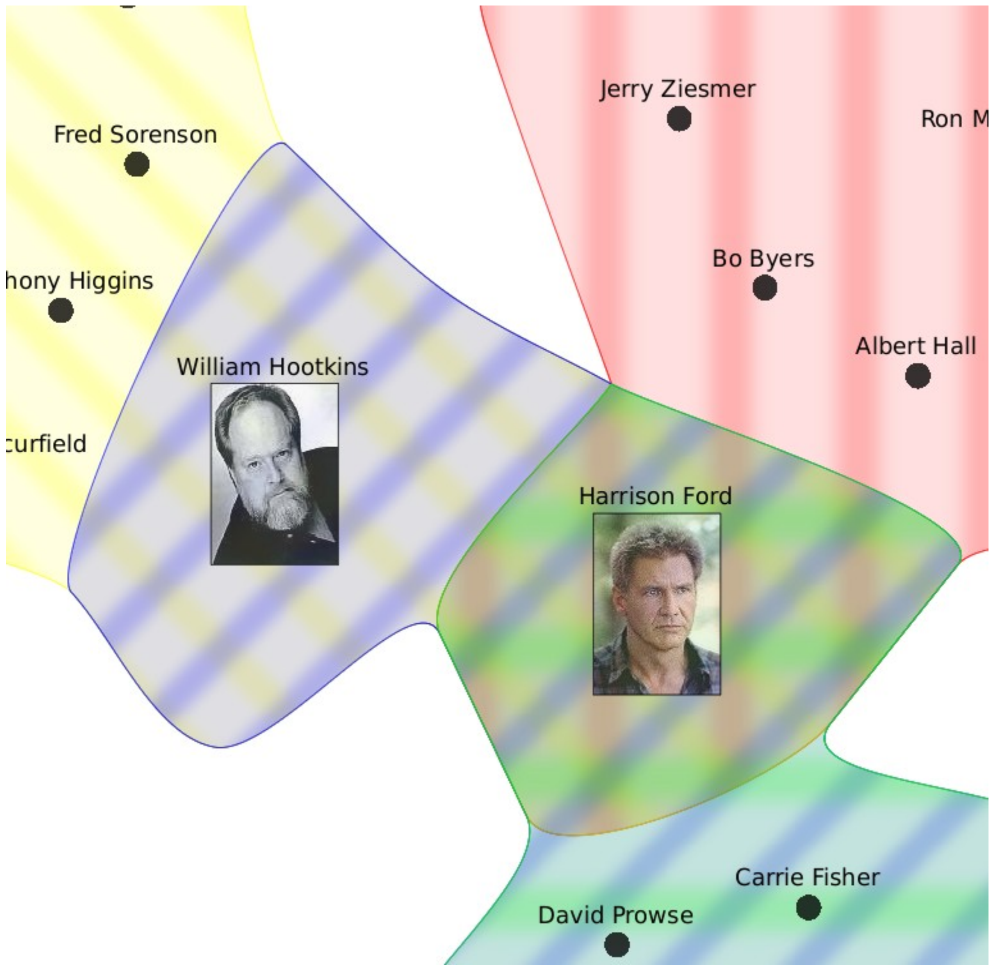
The Welsh and Powell (1967) heuristic for graph colouring is subsequently used to pick nice colour/texture combinations.

Textures

Frequently, colours are not enough to disambiguate intersecting classes.

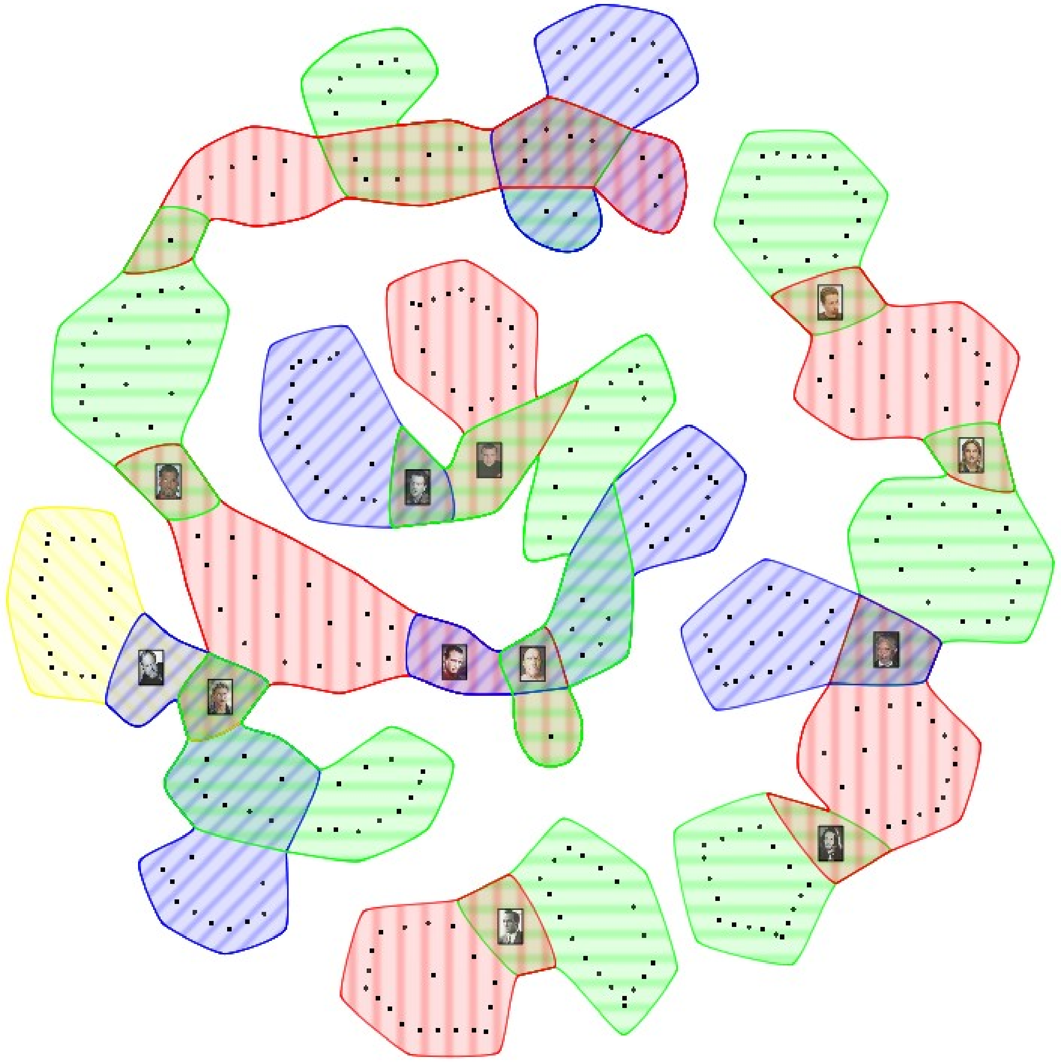
The algorithm uses transparent textures to help overcome this problem.





Examples – Cast of IMDB top 40

- The first twenty actors for each movie, in credit order, are selected.
- Intersections show how movies share cast members.
- Famous actors have pictures texture mapped to their nodes.
- As expected, these actors appear in intersections.



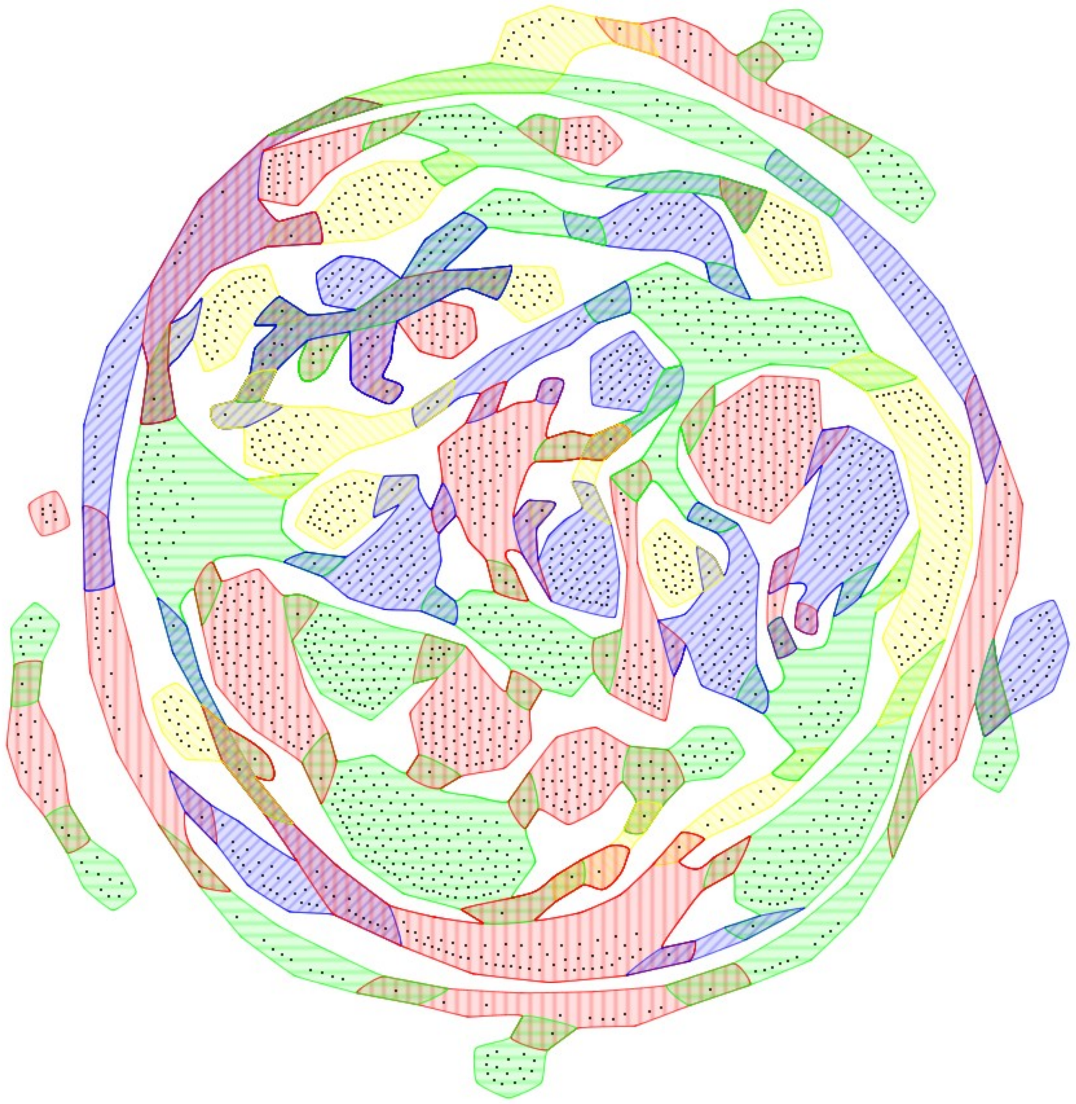
Examples – Cast of IMDB top 70

More complex diagram of the top rated IMDB movies.

60 of the 70 top rated movies with full credited cast.

Over 2000 actors appear as set elements.

Although complex, the diagram is still readable.

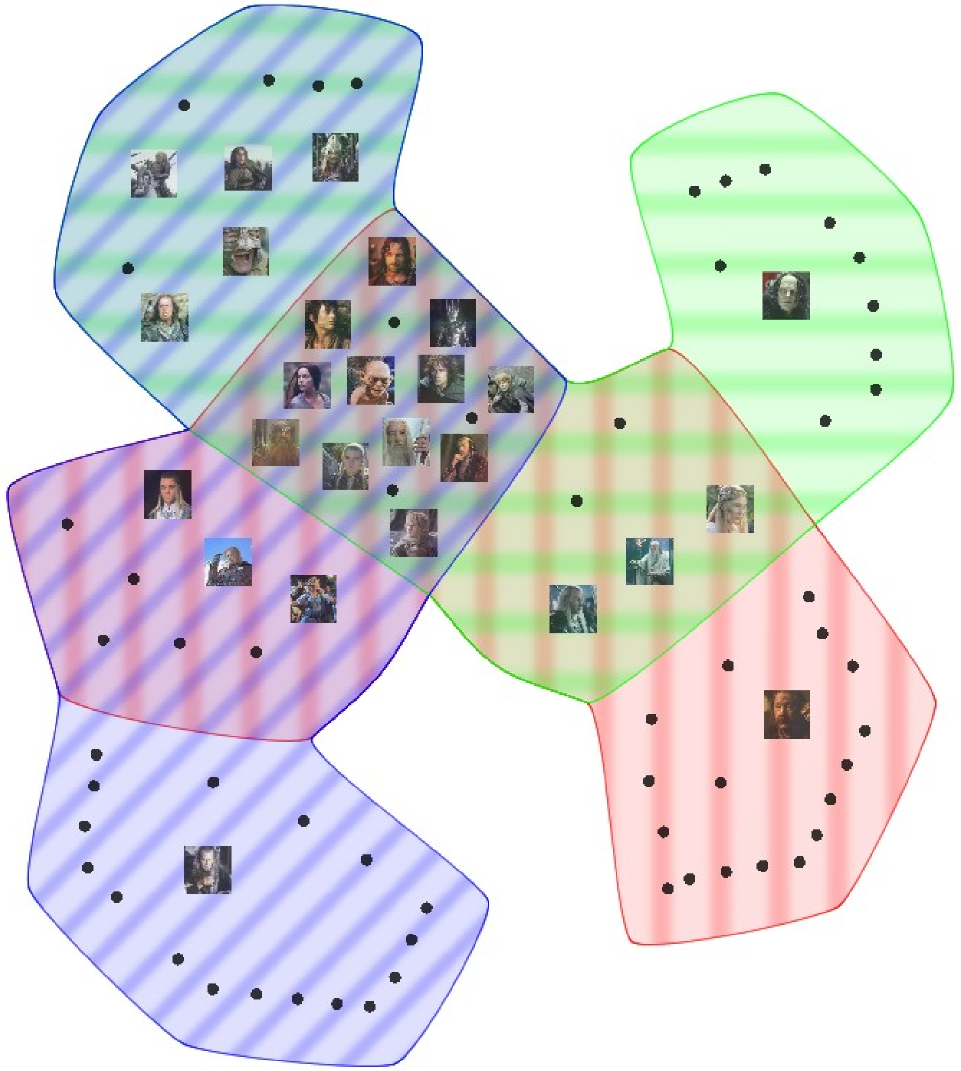


Examples – Cast of *The Lord of the Rings* trilogy - 1

Diagram of the credited cast of *The Lord of the Rings* trilogy.

Red:	*The Fellowship of the Ring*
Green:	*The Two Towers*
Blue:	*The Return of the King*

All the possible zones are present (Venn diagram).

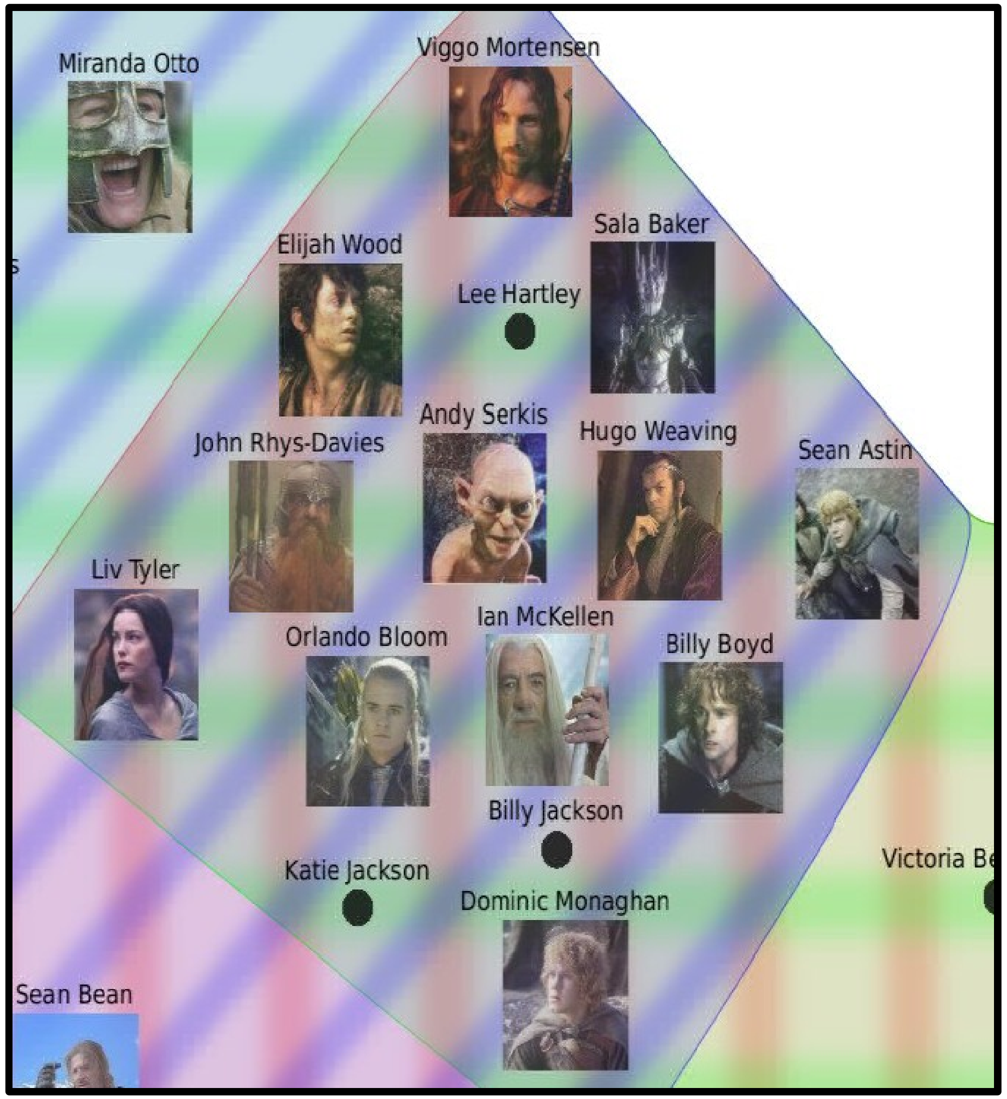


Examples – Cast of *The Lord of the Rings* trilogy - 2

A detail of the intersection of the three films.

The zone collects:

- The most important characters of the trilogy
- The children of the director, who made three cameo appearances
- Lee Hartley, who played orc characters in all the films



Conlcusions and future work

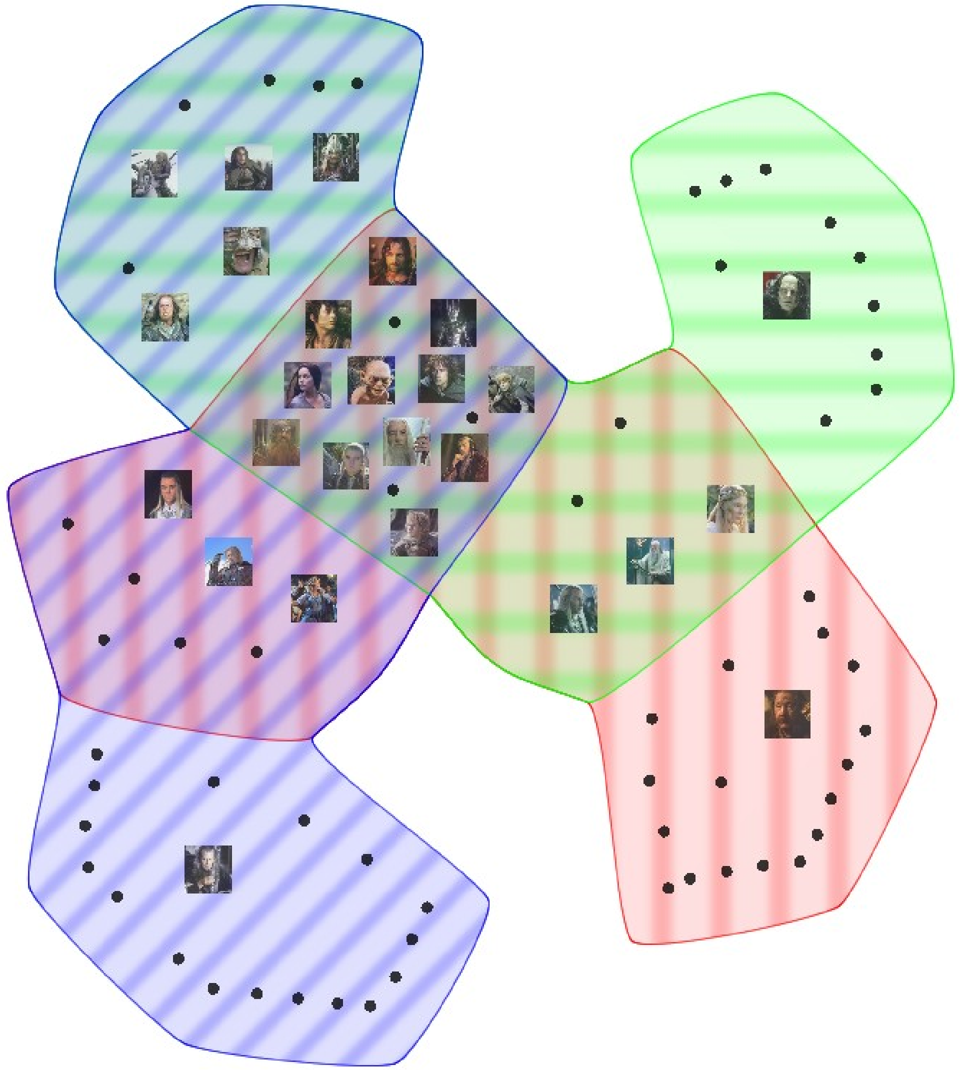
Our algorithm automatically generates a drawing of sets, their elements, and their intersections. It is based on the well-known concept of Euler diagrams.

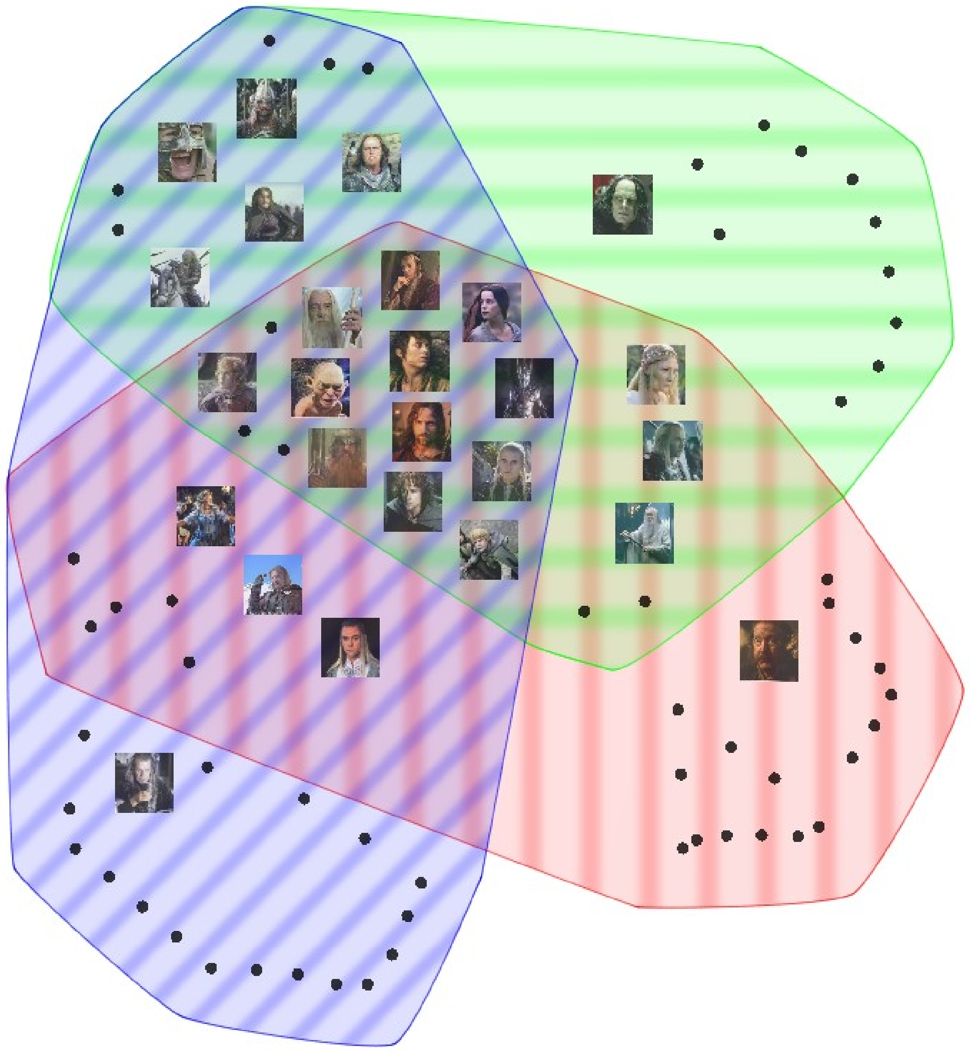
The method introduces techniques such as Bézier boundaries, textures, and colour minimisation in the generation of Euler diagrams.

Many improvements are currently under evaluation, including:

- simplification of the boundary curves
- algorithm execution time improvements
- user interaction
- multi-level diagram exploration
- ...

Future improvements





Thank you for your attention.