Opera Network

Performances across Europe between 1775 and 1833

Thomas Depian ✉
TU Wien, Vienna, Austria

Michael Huber ✉
TU Wien, Vienna, Austria

Wilhelm Wanecek ✉
TU Wien, Vienna, Austria

1 Introduction

Our poster visualizes how operas, including their composers and librettists, travelled across Europe between 1775 and 1833. To do so, we use the provided dataset and group the performances by the triple *title*, *composer*, and *librettist*. This stems from the fact that an opera is not uniquely identified by its title, as it was not uncommon to have multiple composers for the same opera whom all based their work on a common libretto. In the following, unless otherwise specified, we use *Opera* as a synonym for the mentioned triple.

2 Components of the Poster

Since the dataset has an inherent geographical component, we want to emphasize the traveling aspect of operas, composers, and librettists at that time. To put that into more context, we use a map as the background of our poster. As the country borders that we are used to today did not exist back then, we omit them and just highlight topographic features like rivers and seas on the map. Apart from minor adjustments such as the QR-code, the poster is created programmatically using the library d3.js.

Positioning the Cities and Creating the Rings: Our visualization-process starts with the positioning of the cities, which is achieved through a force-based layout. The size of the city is linearly dependent on the number of performances in that city, and we assign to it attracting forces, pushing it towards its geographical location on the map, as well as colliding forces to avoid overlaps with other cities. For space constraints, we place the cities lying outside the main map cutout, namely *Sankt Petersburg*, *Carskoe Selo*, *London*, and *Valetta*, at the border of the map. Whenever the force-computed position deviates from the geographical location, we mark the true position of the city with a dashed line. Once the cities are placed, we create a ring in the corresponding city for each performance, where the rings at each city are ordered chronologically, with the innermost ring representing the first opera from the dataset performed in that city. In addition to the title and year of the performance, each ring contains a reference to the librettist and the composer. The composers and librettists are evenly distributed on the right and left side, respectively.

Connecting Cities with Paths: While the temporal relationship within a city is covered by the order of the rings, the temporal relationship between the cities is not yet established. To do that, we connect performances of the same opera in different cities in their chronological order by a directed path. The paths follow a routing graph based on the Voronoi graph of the approximation of each city-circle with an octagon. This means that we inscribe the outermost ring of each city in an octagon and use the endpoints of the octagons, in combination with evenly spaced points along the boundaries of the map, as input points for a Voronoi diagram.

The Voronoi edges serve as the base for the routing graph. To prevent the paths from going through a city, non-adjacent helper vertices are created for each Voronoi vertex that represents a city. These helper vertices are then connected to different Voronoi vertices outside of the city. Once we have established this naive version of the routing graph we use the *Floyd Warshall* algorithm to compute the shortest path through the routing graph for all sequences of performances.

Clearly, for the plethora of performances in the dataset, many edges of the Voronoi graph will be used by multiple paths. To distinguish different paths from each other, the paths are offset along the edges. This introduces many unnecessary crossings, which can be avoided by ordering the paths on each edge accordingly. We employ the bundle ordering approach proposed by Pupyrev et al. in [3]. Because our paths satisfy the terminal property, i.e., there exist no vertex in which some path ends while others continue, the ordering algorithm computes a crossing-optimal ordering. In general, Voronoi graphs look very angular and artificial, and so does our naive routing graph. As we have multiple points in a relatively small proximity, this results in multiple Voronoi vertices being very close to each other, yielding a zig-zag routing. Pupyrev et al. introduce *circular hubs*, which allow for a seamless transition between different orderings on two edges incident to the same vertex. In our setting, some vertices are so close to each other that the corresponding circular hubs overlap, making the proposed concept hard to realize. We tackle this challenge in two ways: First, we implement a hub-merging mechanism that merges vertices having overlapping hubs, thus eliminating the overlap. This mechanism is refined to not merge large chains of overlapping hubs into a single “mega”-hub, but instead to distribute them into multiple smaller, non-overlapping hubs. Secondly, we employ a mechanism detecting zig-zag paths that can be straightened. This helps in decluttering the visual appearance, giving the routing graph a more natural and (arguably) appealing look. The curves inside the circular hubs are defined using bi-arcs, as also done in [3], and described in [2]. Finally, we mark the direction of a path with small, equidistantly placed arrows.

Colors and the Collaboration Network: We want to aid the user with the way we color the paths and rings. To do this meaningfully, we assign each composer a base color and each librettist a shade. The color of an opera, and so of its paths and rings, is defined by the combination of base color and shade. The *Collaboration Network* above the legend serves as a lookup table for the used colors in the drawing. It is a modification of a classical chord diagram, where the central cutout is established with dummy entries that serve as a placeholder. Again, we place the librettists on the left side, now ordered by the number of operas they have written, and on the right side the composers. To minimize overlaps, we order the composers manually in an appealing way. The size of a chord is defined by the number of unique operas with this composer-librettist combination.

3 Interactive Version

As already mentioned at the beginning of Section 2, we created the poster programmatically using the library d3.js. Therefore, we create in addition to the poster also an interactive online-version to enhance the exploration of the data. While the poster should give an overview over the dataset, the interactivity should serve as the main tool for exploring it. The online-version can be accessed via `https://opera-network.netlify.app/` [1] and offers the following interaction possibilities.

Hovering over a Performance or a Path: By hovering over the path or the ring of a performance, the hovered opera will be highlighted in the poster. This allows the user to see the cities in which the opera performed and the path it took.

Hovering over a Composer or Librettist: By hovering over a composer or a librettist, all operas, i.e., rings and paths, of the composer or librettist will be highlighted in the visualization. This allows the user to see in which cities or regions a composer or librettist was active.

Clicking on a Performance or a Path: Clicking on a path or a ring of a performance has the same effect as hovering over it. However, the highlighting-effect gets locked and remains until the user clicks on an empty spot on the map. In this way, the user has the ability to conveniently analyze the travel aspect of an opera without having to constantly hover over it.

4 Answered Research Questions

Finally, we want to emphasize which research questions from the contest page can be answered in our visualization and to what extent. The question of how certain operas have traveled through Europe can be answered by the directions of the paths and colors of the rings. By identifying the color of an opera, we can start at any city where it was performed and follow its path through Europe. Using the performance years indicated in the city rings, we can also get a feeling for the temporal aspect and development of that performance. Furthermore, based on the dominance of a shaded color in a city or region, it can be seen whether the opera stayed at that place or went through Europe. Moreover, the dominance of a base color in a city can also be an indication of the popularity of that composer in that city. Lastly, the question of patterns in collaborations can be answered with the help of the collaboration network. Ultimately, it should not go unmentioned that we can use the visualization to answer other research questions. The size of a city, for example, is an indication of the popularity of that place in the opera scene at that time. Or the period of time during which a composer or librettist can be considered *active* in a city could be estimated from the position and frequency of performances of operas by that librettist or composer. These are shown with the points on the composer's or librettist's trajectory line.

References

1. Thomas Depian, Michael Huber, and Wilhelm Wanecek. Opera network, Sep 2022. URL: `https://opera-network.netlify.app/`.
2. Les A. Piegl and Wayne Tiller. Biarc approximation of NURBS curves. *Computer-Aided Design*, 34(11):807–814, September 2002. `doi:10.1016/s0010-4485(01)00160-9`.
3. Sergey Pupyrev, Lev Nachmanson, Sergey Bereg, and Alexander E. Holroyd. Edge routing with ordered bundles. *Computational Geometry*, 52:18–33, 2016. `doi:10.1016/j.comgeo.2015.10.005`.