Techniques for visualisation of combinatorial maps

Tingyu Huang

18999855

Noam Zeilberger

Theory

Project aim

The core purpose of the project is visualizing the combinatorial maps via variant approaches. In order to share the achievement with others and improve the usability of the solutions, the project will be presented as a website. The visualization of graphs and tree have caused a heated discussion and gradually formed a professional researching area, however, due to the concept of combinatorial maps introduced formally by A. Jacques[Jac70] at 1970, which called "Constellations" at that time, there is quite a few eyes focus on the visualization of it. Actually, an effective and vivid representation of combinatorial maps could help the beginners learn it directly, as well as, improve the efficiency and productivity for researchers to conduct the related research. Accordingly, it is vital to create a tool for visualizing the maps with some interactive operations with which users can customize some parameters and observe the different states of the maps. It can be seen that the entire project is based on the combinatorial maps. Combinatorial maps is a typically topological constructors base on the graphs, which contains many features not only relying on the underlying graph theory but also on itself, therefore, exploring the characters and sub-structures of it will bring strongly theoretical support.

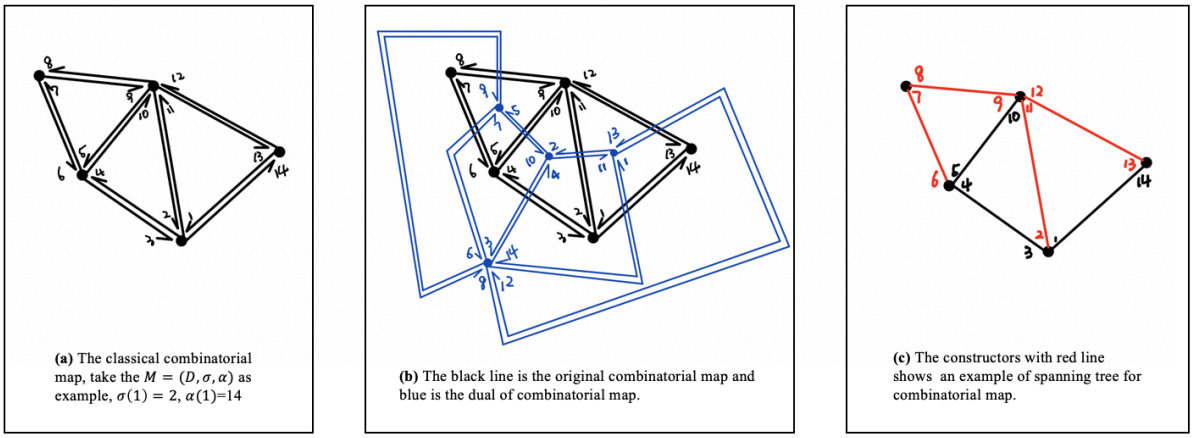
Related work

Basic concepts

- The chief distinction of combinatorial maps is its notation, unless the traditional planner graph which comprises the nodes and edges, the edges of it subdivided into two part and named as darts or half-edges. This alteration bring a new definition of components faces and vertices, so that its notation either shown as a set of permutations $M = (D, \sigma, \alpha)$[Jac70] in which D is the finite set of half-edges, α is the permutation of darts, as well as σ is the permutation of vertices, or expressed as $M = (D, \phi, \alpha)$[1] where $\phi = \sigma$ o α is the permutation of faces.

- Since the combinatorial map is the variation of the planar map that is an embedded graph on the surface, it has a **dual map** on the same surface and define as $M^* = (D, (\phi^{-1}, \alpha^{-1}, \sigma^{-1}))$[nLa19]. The vertices of the dual map are the center of the each faces, and the connections between arbitrary two vertices are dividing the edges which shared by corresponding faces[BK99].

- Similarly, in graph theory, a planar graph belongs to the undirected graph. In the mathematical research, a subgraph performs as a tree which includes all of the vertices of the undirected graph called **spanning tree**, and it is obvious that it is involved in the planer map which means that it is possible to abstract the spanning tree from the combinatorial map. Obviously, a conclusion can be derived that there is not only one spanning tree for a certain map.



(a) The classical combinatorial map, take the $M = (D, \sigma, \alpha)$ as example, $\sigma(1) = 2$, $\alpha(1)$=14

(b) The black line is the original combinatorial map and blue is the dual of combinatorial map.

(c) The constructors with red line shows an example of spanning tree for combinatorial map.

Figure 1 Examples of given concepts

Three structures mentioned above is shown in Figure 1, and there is no doubt that the orientation of each component is not consistent. The ordering of the face is mostly clockwise while the outside one is contrary which is the same with the vertices.

Visualization

- In the study of Dainis Zeps and Paulis Kikusts[ZK12], they found an interesting way to pursue the visualization of combinatorial maps by introducing a new concept of the corner. This idea assigned a different representation of each component in the map. However, they just came up with the theory research and did the visualization work by a drawer given before, which led to the loss of deliberating on how to generate the combinatorial map automatically, for example, they did not talk about the method of layout and switch the entire problem to a useful algorithm.

- The handbook of Pigale [FP02] which is both the library and a graph editor told that its data structure was the combinatorial map. From the article, they came up some details about the combinatorial map, for instance, the instructive operators of the map, and the basic algorithm deep first searching algorithm(DFS) of it. It is worth mentioning that the algorithm is helpful for finding a spanning tree of the map, meanwhile, by visualizing the tree with the help of tree drawing algorithms, it is easy to achieve the target.

- There are many heuristic algorithms for visualization of planer map like force-directed drawing algorithms summarized by Stephen G. Kobourov[K13]. From these algorithms, some practical layout algorithms can be drawn.

- A specific idea has been implemented just relying on the classical components of the combinatorial map and the force-directed algorithm. This is the first attempt for visualization with the Python.

Objectives

1. Computation of faces by given the permutations of vertices and darts. This particularly depends on the equivalent $\phi = \sigma$ o α which means, the faces are surrounding by the darts where any two of them are adjacent the same vertex, but one is incidence and another one is exit.

Correctness: The composition of the permutations is the identity: $\phi\alpha\sigma = id$, namely, any component could be educed by others.

2. Calculating the three components to define a random combinatorial map with given the length of D and signed as N. Firstly, creating vertex permutations with size of N by a randomly generating permutation algorithm—Fisher-Yates shuffle[FY43], simultaneously, this algorithm also used to generate the half-edge permutations with the constrains that there is no cross between the loops and the connecting two different vertices edges, in another word, the half-edges of a loop need to next to each other around the same vertices, as well as, reducing the cross between two faces as much as possible. Finally, the reasoning of faces still uses the method in objective 1.

Correctness: The DFS algorithm is used to test the correctness of the results, so that, if it can compute a path through the whole vertices successfully, it would be a true consequence.

3. Parsing the combinatorial map as an arbitrary minimum spanning tree with the DFS. It is well known that the DFS is used to find a simplify route between any two nodes. Here, two vertices are fixed, as the result, the path begins from the first node to the last one, and the path would be the minimal set of edges, which also can be seen as the spanning tree.

Correctness: The results would pass all vertices without duplication.

4. The combinatorial map can be drawn by the spanning tree in objective 3. In the article, Tree Drawing Algorithm[R13], supports plenty of algorithms to draw the trees. To find a suitable one and implement it, after that, adding the rest half-edges to complete the map structure.

Correctness: The outcomes based on the tree structure, hence, it needs to show the tree constructor accurately and the whole structure must follow all permutations of the map.

5. Finding the dual of combinatorial map. The vertices of dual map are the middle of faces and the surrounding darts of each node are the ordering half-edges around the faces in primitive maps.

Correctness: The number of vertices in dual is the same with the faces of original combinatorial maps, the permutation of them is the same as well, as the same time, the permutation of darts in both are similar.

6. Visualizing the combinatorial map under dual maps built above. The faces of combinatorial map could be generated by the corresponding nodes in dual, hereafter, growing each face together according to the same darts and vertices.

Correctness: Each element has the right ordering of darts as the same as the permutations provided.

7. The simple frame for displaying the interface of website. Designing the style of system and confirming the functions and operations when users utilize the tool.

Correctness: Each element can show and present successfully, the structure of the interfaces shows clearly, and the basic interactions with users are frequently.

8. Converting the previous methods from the python to the JavaScript and accomplish all part of the system. The format for the image can be variable.

Correctness: The images with different methods show clearly and correctly, the structure of the system is simple and clear, and the interaction or operation is smooth and convenient.

Methodology

The project pays more attention to finding the methods to visualize the combinatorial map. Hence, the first task is concerning what is the combinatorial map and what characters and feature it has. Through learning the definitions and theorems, and researching the topological structures, the basic directions of approaches are proposed. Beginning with the classical representations of the map combining with the typical graph drawing algorithm. The second and third method prefer finding the isomorphism, variant, and subgraph of the map, the dual maps and spanning trees.

For the visualization of graphs, there are many mature algorithms that can be used and referenced after understanding their core ideas. In order to do a better performance for users, an interactive interface is necessary. Website is more easy to access and convenient to interact, and this is the reason why building the system as a website.

The tool for doing the research is mainly relying on the Google scholar, by retrieving the key works “combinatorial maps”, “spanning tree”, “dual map”, “graph drawing” and “tree drawing”, and combining with the operations “and”, “ordering by year or citation”, etc.

Project plan

I have abundant background knowledge and know the technology for producing the web pages. The specific plan is shown as a Gantt chart in Figure 2, the whole process cost about 90 days from 10, June to 10, September. At the beginning of the project, the two essential tasks are discussing the detail of topics and writing the proposal. After that, stepping into the core part of the project, the researching and implementation of the ideas, and writing the dissertation to report the entire project.

The

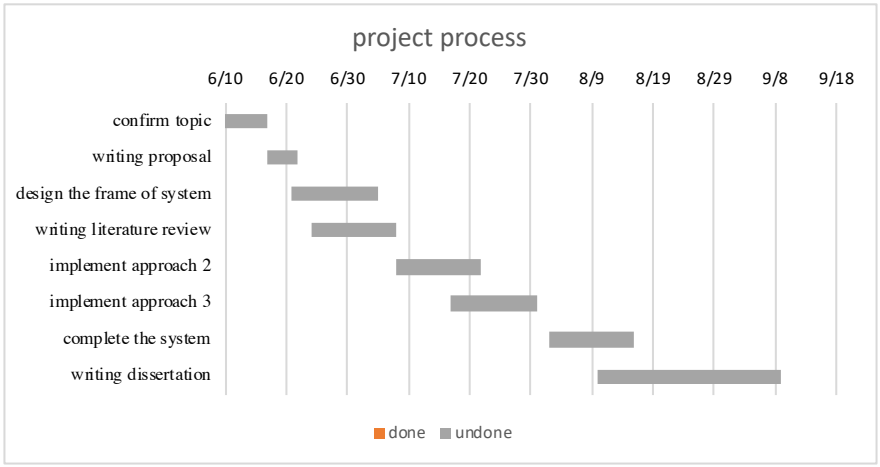


Figure 2 The project plans

implementation divided into 4 partitions. Firstly, designing a frame of the system from 21, June, cause the final representation of the project as a website. Whatever methods to implement all display in the web pages so that a clear style and operations of the interface can help improve the productivity. During this progress, the literature review of the report could be wrote based on the materials collected before. The second and third steps implement the second approach of visualization which begins at 8, July and spending 14 days and third method that spends the same number of days following the second one, respectively. Last but not least, completing the whole system, which includes two works, converting the first approach from Python to JavaScript and import it to the system, and refining the system by testing the frequency of the system and enhance the interaction between the system and the users. In fact, the whole system is nearly completed after the second and third stages, hence, the technique parts of dissertation can be done before the 25, August and the rest of time is writing the remaining part.

Risks and contingency plan

1. In objective 6, there is no good selection to growing each face. If it is difficult to grow them together, it could just display them separately.
2. If using the mean in 1, it would be necessary to find an appropriate layout algorithm to do the representation for saving time, otherwise, it could just divide the canvas by the number of faces n. Eventually, if n is too large, it is not quite clear to show the each face.
3. Embedding the images into the website and exporting with different formats. Searching more solutions, and choose the best.

Reference

[Jac70] Jacques, André. "Constellations et graphes topologiques." *Combinatorial Theory and Applications* 2 (1970): 657-673.

[nLa19] nLab authors. *combinatorial map (2019)*. [online] Available at: https://ncatlab.org/nlab/show/combinatorial+map [Accessed 19 Jun. 2019].

[BK99] Brun, Luc, and Walter G. Kropatsch. "Dual contraction of combinatorial maps." *Workshop on Graph based representations, Austria*. 1999.

[ZK12] Dainis Zeps and Paulis Kikusts. How to draw combinatorial map? from graphs and edges to corner rotations and permutations. arXiv preprint arXiv:1207.5673, 2012.

[FP02] de Fraysseix, H., and P. Ossona de Mendez. "PIGALE: Public Implementation of a Graph Algorithm Library and Editor. Free Software (GPL licence), 2002."

[K13] Kobourov, Stephe. "Force-Directed Algorithms."Handbook of Graph Drawing and Visualization(2013).

[R13]Rusu, Adrian. "Tree Drawing Algorithms." Handbook of Graph Drawing and Visualization(2013).

[FY43] Fisher, Ronald A., and Frank Yates. Statistical tables for biological, agricultural and medical research. Oliver and Boyd Ltd, London, 1943.

[Tam13] Roberto Tamassia. Handbook of graph drawing and visualization. Chapman and Hall/CRC, 2013.