Optical Graph Recognition on a Mobile Device

Christopher Auer, Christian Bachmaier, Franz J. Brandenburg,
Andreas Gleißner, and Josef Reislhuber

University of Passau, 94030 Passau, Germany
{auerc,bachmaier,brandenb,gleissner,reislhuber}@fim.uni-passau.de

In [1] we proposed Optical Graph Recognition (OGR) as the reversal of graph drawing. A drawing transforms the topological structure of a graph into a graphical representation. Primarily, it maps vertices to points and displays them by icons, and it maps edges to Jordan curves connecting the endpoints. In reverse, OGR transforms the digital image of a drawn graph into its topological structure. The recognition process is divided into the four phases preprocessing, segmentation, topology recognition, and postprocessing. In the preprocessing phase OGR detects which pixels of an image are part of the graph (graph pixels) and which pixels are not. The segmentation phase recognizes the vertices of the graph and classifies the graph pixels as vertex and edge pixels. The topology recognition phase first recognizes edge sections. Edge crossings divide edges into edge sections, i. e., the regions between crossings and vertices. The edge sections are merged into edges in the most probable way based on direction vectors. The postprocessing phase concludes OGR with tasks like converting the recognized graph into different file formats, like GraphML or adding coordinates to the vertices and edges, such that the recognized graph resembles the input graph.

Our OGR Java implementation OGR$^{\text{up}}$ is able to recognize drawings of undirected graphs with the following properties: Vertices are drawn as filled objects such as circles, edges are drawn as contiguous curves of a width significantly smaller than the diameter of the vertices, and they should exactly end at the vertices. Our desktop version of OGR$^{\text{up}}$ is of limited use, because it needs a camera as a second device to take a picture of the graph.

In contrast, the new Android version of OGR$^{\text{up}}$ needs only a single device. The picture of the graph, e. g., drawn on a whiteboard, is directly taken with the camera of the mobile device at hand. The part of the image that contains the graph can be selected via touch gestures, as seen in Fig. 1. Finally, the graph is recognized and used for further processing. It can be shared, visualized and edited on the mobile device, e. g., as proposed by Da Lozzo et al. [2].

For the Android version, we had to re-implement parts of the graph recognition algorithm and we developed a GUI that fits the capabilities of a mobile device. Whereas the computation time is acceptable in the desktop version of OGR$^{\text{up}}$ ($\approx$ 10 seconds for high resolution images), the computation time becomes unaccetably long in the Android version due to hardware limitations. The established digital image processing library OpenCV [3] helped to improve the computation time of OGR$^{\text{up}}$. To circumvent further performance issues, and to make OGR$^{\text{up}}$ available for different mobile operating system, like iOS or Windows Phone, we plan a web service implementation of OGR$^{\text{up}}$.

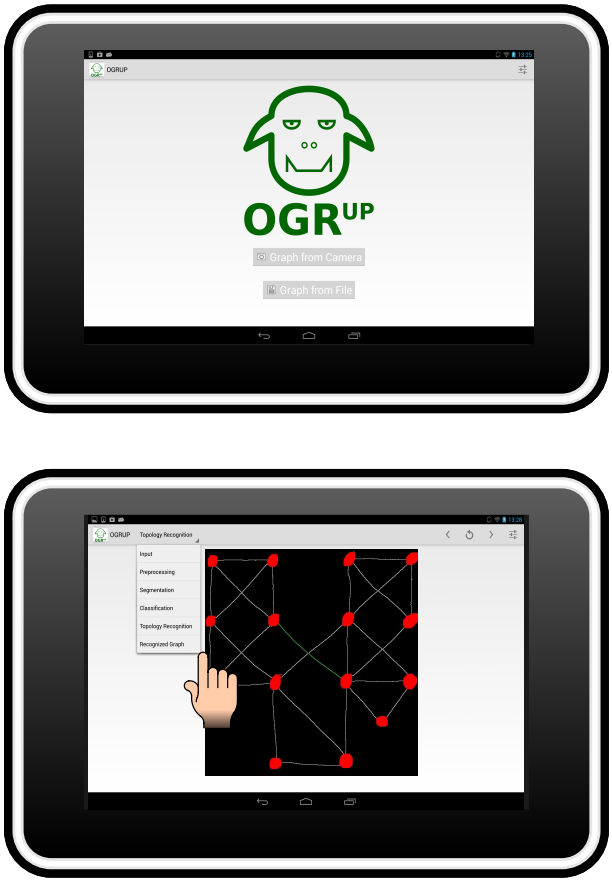

Fig. 1. Two screenshots of OGR^{up} on an Android tablet.

References

1. Auer, C., Bachmaier, C., Brandenburg, F.J., Gleißner, A., Reislhuber, J.: Optical graph recognition. J. Graph Alg. App. 17(4), 541–565 (2013)
2. Da Lozzo, G., Di Battista, G., Ingrassia, F.: Drawing graphs on a smartphone. J. Graph Alg. App. 16(1), 109–126 (2012)
3. Itseez: OpenCV. `http://www.opencv.org/`