Graph Harvester

Julius Deynet
Universität Würzburg, Germany

Tim Hegemann
Universität Würzburg, Germany

Sebastian Kempf
Universität Würzburg, Germany

Alexander Wolff
Universität Würzburg, Germany

Abstract

We present *Graph Harvester*, a website for extracting graphs from illustrations in scientific papers. For every graph that has been extracted, Graph Harvester queries the graph database *House of Graphs*. If the graph is not already present there, the user can upload the graph into the database, possibly after modifying it, and with a reference to the paper that contains the drawing of the graph.

2012 ACM Subject Classification Applied computing → Graphics recognition and interpretation; Mathematics of computing → Graph theory; Information systems → Web applications

Keywords and phrases House of Graphs, Graph recognition, Information extraction

Digital Object Identifier 10.4230/LIPIcs.GD.2024.58

Category Software Abstract

Supplementary Material
Software (Source Code): `https://github.com/JuliusDeynet/graph_harvester` [5]
archived at `swh:1:dir:f390ce9e8201cb8d2c97848e8fb5170173dcb82b`

Funding *Tim Hegemann*: Supported by BMBF grant 01IS22012C.

1 Graph Harvester

In graph theory, certain graphs are pivotal in numerous publications, serving as key tools in proving or refuting theorems. Hence, having a collection of significant graphs previously used in publications is highly beneficial. This is the idea behind the website *House of Graphs* (HoG) set up by Coolsaet et al. [3, 4]. It explicitly does not aim at making all possible graphs (of a certain size) available, but only "interesting" ones. On the site, one can search graphs using many criteria, draw graphs, and upload new graphs. Still, from a graph drawing perspective, the HoG database is far from being complete. Out of 293 potentially interesting graphs (with at least seven vertices and maximum vertex degree of at least 3) that we extracted from figures in papers presented at GD 2023, the HoG database contained only 57.

Unfortunately, extending the HoG collection is time-consuming. Currently, graphs must be uploaded in one of two formats: either as an adjacency matrix or in graph6 string representation. HoG users, however, may have their graphs represented differently, e.g., as a drawing. Converting a drawing to the required formats by hand is prone to errors for smaller graphs, and practically impossible for larger ones.

To address this problem, we present *Graph Harvester* [6], a website that extracts graphs from drawings of graphs in PDF files. Whereas there has been work on extracting graphs from bitmap images [1], we focus on the simpler problem of extracting graphs from vector data, which is the primary format used in publications nowadays. Graph Harvester works as follows. A user uploads a publication as a PDF file. Then, the website displays each detected figure in the file next to a drawing of the extracted graph for review. Furthermore, the website computes a representation in the format accepted by HoG and checks whether the graph is already present in the HoG database. To this end, the following steps are performed.

The initial extraction process is based on modules of the KIETA [7] pipeline. Graphical and textual elements within the PDF file are extracted using PyMuPDF, a Python library for data extraction, analysis, conversion, and manipulation of PDF documents. The graphical elements are clustered to find a cohesive group that contains a figure in vector format (SVG). Then, the corresponding figure caption is determined using a keyword-based approach.

Next, the graphs are extracted from the figures. For this step, four geometric objects are of importance: circles and rectangles, which represent potential vertices; and line segments and curves, which represent potential edges. We call two line segments or Béziers curves *connected* if they share an endpoint (i.e., their endpoints are closer than a given threshold).

Because the PDF format does not support circle objects natively, these are typically represented by cubic Béziers curves. Therefore, any connected sequence of three or more curves forming a cycle, with endpoints at similar distances from their joint barycenter, is considered a circle. The latter condition is used to filter out elliptic shapes, which typically do not represent vertices (but are rather used for highlighting).

Next, the potential vertices are filtered for duplicates and by size. Curves that form cycles but have not been classified as vertices are retained for further consideration, as they may represent edges. Rectangles whose four corners coincide with potential vertices are split into four line segments. Then, edges are handled. Any sequence of connected line segments or Béziers curves is considered an edge candidate if both endpoints match vertex candidates. (Otherwise, they usually represent axes or other annotations.) Edge candidates are subdivided if they contain vertex candidates in their relative interior.

Finally, the graph of all vertex and edge candidates is built and split into connected components. The adjacency matrix is set up separately for each component. Graphs (or components) with fewer than seven vertices or a maximum vertex degree of less than 3 are filtered out, as they are not considered interesting. The remaining matrices are converted to graph6 format and sent to the HoG interface to check whether the corresponding graphs are already present in the HoG database. If they are, their HoG identifiers are reported.

The graph6 representations, HoG identifiers, figure captions, and geometric objects used to represent the graphs are displayed by the Graph Harvester website. The latter are drawn as they were found in the paper. Additionally, a picture of the figure extracted from the original PDF is provided to allow the user to validate Graph Harvester's extraction. If a graph already exists in the HoG database, a link to the HoG entry is shown. Otherwise, users can choose to add the graph to the HoG database, possibly after modifying it with the HoG graph editor. Figure 1 shows a screenshot of the Graph Harvester website.

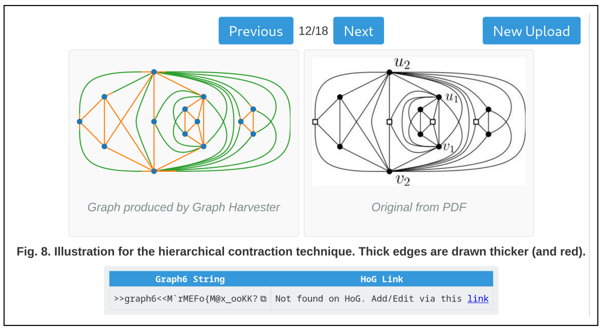

Fig. 8. Illustration for the hierarchical contraction technique. Thick edges are drawn thicker (and red).

Graph6 String	HoG Link
>>graph6<<M`rMEFo{M@x_ooKK?	Not found on HoG. Add/Edit via this link

Figure 1 Output of Graph Harvester for Figure 8(b) of a paper of Bekos et al. [2].

References

1 Christopher Auer, Christian Bachmaier, Franz Brandenburg, Andreas Gleißner, and Josef Reislhuber. Optical graph recognition. *J. Graph Algorithms Appl.*, 17(4):541–565, 2013. doi:10.7155/jgaa.00303.

2 Michael A. Bekos, Walter Didimo, Giuseppe Liotta, Saeed Mehrabi, and Fabrizio Montecchiani. On RAC drawings of 1-planar graphs. *Theoret. Comput. Sci.*, 689:48–57, 2017. doi:10.1016/j.tcs.2017.05.039.

3 Gunnar Brinkmann, Kris Coolsaet, Gauvain Devillez, Jan Goedgebeur, and Hadrien Mélot. House of Graphs. Website, accessed on Sep. 6, 2024. URL: https://houseofgraphs.org.

4 Kris Coolsaet, Sven D'hondt, and Jan Goedgebeur. House of Graphs 2.0: A database of interesting graphs and more. *Discret. Appl. Math.*, 325:97–107, 2023. doi:10.1016/J.DAM.2022.10.013.

5 Julius Deynet, Tim Hegemann, Sebastian Kempf, and Alexander Wolff. Graph Harvester. Software, swhId: swh:1:dir:f390ce9e8201cb8d2c97848e8fb5170173dcb82b (visited on 2024-10-14). URL: https://github.com/JuliusDeynet/graph_harvester.

6 Julius Deynet, Tim Hegemann, Sebastian Kempf, and Alexander Wolff. Graph Harvester. Website, accessed on Sep. 6, 2024. URL: https://go.uniwue.de/graph-harvester.

7 Sebastian Kempf, Markus Krug, and Frank Puppe. KIETA: Key-insight extraction from scientific tables. *Appl. Intell.*, 53:9512–9530, 2023. doi:10.1007/s10489-022-03957-8.