

GRASP: A MATLAB TOOLBOX FOR GRAPH SIGNAL PROCESSING

Benjamin Girault^{*, \diamond , \dagger} , Shrikanth S. Narayanan \diamond , Antonio Ortega \diamond , Paulo Gonçalves^{*}, Éric Fleury^{*}

^{*} Univ Lyon, Inria, ENS de Lyon, CNRS, UCB Lyon 1, 69342, Lyon, FRANCE

\diamond University of Southern California, Los Angeles, CA 90089, USA

\dagger Corresponding author.

ABSTRACT

The GraSP toolbox aims at processing and visualizing graphs and graphs signal with ease. In the demo, we show those capabilities using several examples from the literature and from our own experiments.

Index Terms— graph signal processing.

1. THE GRASP TOOLBOX

The emerging field of graph signal processing aims at studying signals on (possibly) irregular discrete supporting domains. Whereas classical signal processing relies on well founded mathematical properties of Euclidean spaces, we here rely on the algebraic properties of graphs and the derivation of a graph Fourier transform and harmonic analysis of graph signals found in the literature [1], where the signal is a function associating a value to each vertex, and edges code relations between those values. The goal of *GraSP*, for *Graph Signal Processing*¹, is to assemble in a unified framework a series of analysis and visualization tools for graphs and graph signals. Developed for the Matlab environment, a popular choice for the signal processing community, it benefits from its advanced plotting capabilities to yield an efficient and complete toolkit for graphs signal processing. Moreover, the code is well documented, versatile, and easy to understand and contribute to.

1.1. Graph and Plotting

Several functions generate known graph structures, from the very simple cycle graph and grid graphs (underlying Euclidean domain), to the Barabási-Albert and Watts-Strogatz graphs (useful for simulation of social networks without communities). Some are weighted such as the graph of Fig. 1 (Gaussian kernel of the Euclidean distance). Importing and exporting graph structures using CSV files is also available

This work was supported in part by NSF under grants CCF-1410009, CCF-1527874, CCF-1029373, and by Labex Milyon.

This toolbox has been developed by B. Girault during his Ph.D. at École Normale Supérieure de Lyon.

¹The toolbox is a free software (GPL-compatible license) available at <http://grasp.gforge.inria.fr>.

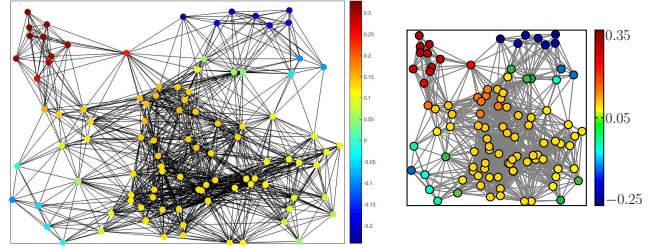


Fig. 1: A graph and a low pass graph signal depicted using Matlab (left) and the L^AT_EX package (right).

and allows interacting with other systems. Graphs are stored in a Matlab structure holding all pertinent data to perform efficiently graph signal processing.

The centerpiece of the toolbox is the function `grasp_show_graph` plotting a graph and a graph signal using a color scale (see Fig. 1). This function has been optimized for later modification of the signal, for example in animations. Examples of animations are given with a GUI iterating an operator, and through a function generating a GIF file from a sequence of graph signals (this can be for example a time varying graph signal).

1.2. Graph Signal Processing

GraSP is able to compute the matrix of the graph Fourier transform based on the three different approaches found in

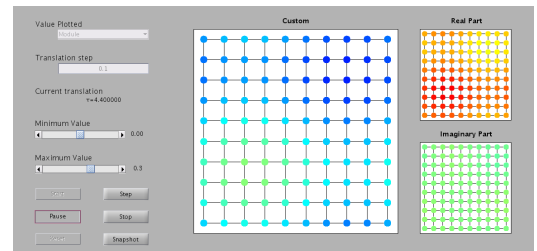


Fig. 2: An example of a Matlab GUI taking advantage of the plotting efficiency of the toolbox to visualize the iteration of an operator on a graph signal.

```

% Build a graph and a graph signal
g = grasp_plane_rnd(100);
g.A = grasp_adjacency_thresh(g, 0.01);
g = grasp_eigendecomposition(g);
x = grasp_heat_kernel(g, 2);

% Display them
grasp_show_graph(gca, g,
                'node_values', x);

% Export them
grasp_exportcsv(g, 'nodes.csv', 'edges.csv');
grasp_exportcsv_signal(g, x, 'heat.csv');

```

Fig. 3: Matlab code to generate Fig. 1.

the literature: from the standard Laplacian [1], the normalized Laplacian [1], or the graph shift [2]. Based on this matrix, the user can perform spectrum analysis of signals, spectrum filtering, or filter design, to cite only a few applications.

1.3. \LaTeX Packages

GraSP includes several \LaTeX packages to directly plot graphs and signals within a \LaTeX document using only CSV files. This achieves multiple goals. First of all, it allows to quickly tweak how figures look, without going back and forth between Matlab and \LaTeX . Second, it allows to generate animations in Beamer presentations in a very simple manner, without having to generate each figure of the animation separately. Finally, the packages act as PGF/TikZ macros, effectively allowing then to use the full power of PGF/TikZ to tweak figures with additional graphics. These packages have been used in [3] to generate all but one figures (the exception being a screenshot of Matlab).

2. DEMO

The demonstration shows the capabilities of the toolbox, especially when it comes to plotting. We show how quickly one can start using the toolbox and get results with several show-cases taken from the literature and our experiments. We also show how we build an environment for graph signal processing tools for Matlab with GraSP as a foundation and how we hope to build a community from that environment.

Requirements for the demo are very limited. Ideally, it would be shown on screen, or projected, but the demonstration can also be performed on a laptop.

3. COMPARISON WITH THE STATE OF THE ART

This abstract would not be complete without mentioning another toolbox released earlier, but developed during similar

```

\begin{tikzpicture}[scale=0.23]
  \flatgraph[colormap=jet,showcolormap=true,%
    minvalue=-0.25,maxvalue=0.35,%
    nodesize=13pt]%
    {nodes.csv}{edges.csv}{heat.csv}
\end{tikzpicture}

```

Fig. 4: \LaTeX code to generate Fig. 1.

time frames: The GSPbox² [4]. The goal of both toolbox are identical in the sense that they implement functions to easily perform graph signal processing. GraSP implements two functions to convert the graph structures from one toolbox to the other, making them compatible. We now list several additional functionalities that each toolbox present.

We begin by what GSPbox offers and is missing from GraSP. GSPbox is capable of making clusters within a graph better identifiable (useful when communities are known in advance). It is also better equipped to plot signals and filters in the Fourier domain, with the use of graph frequencies.

On the other hand, our toolbox is capable of using the third party library *GraphVIZ* to plot a graph whose vertices do not have 2D coordinates. Several algorithms are available to achieve this (see the documentation of GraphVIZ). When it comes to plotting, we can also plot values on edges using colors (as opposed to thickness for GSPbox). Graph plotting has been optimized for animation, thus allowing efficient GUI where signals may change. We also provide a function to plot matrices where entries of the matrix appear according to a colormap (instead of black and white for the builtin `imshow`). Finally, as shown in subsection 1.3, we provide \LaTeX packages to draw graphs, signals on graphs, and matrices using \LaTeX .

REFERENCES

- [1] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, “The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.
- [2] Aliaksei Sandryhaila and José M. F. Moura, “Discrete Signal Processing on Graphs,” *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.
- [3] Benjamin Girault, *Signal Processing on Graphs - Contributions to an Emerging Field*, Phd thesis, Ecole normale supérieure de lyon - ENS LYON, Dec. 2015.
- [4] Nathanaël Perraudin, Johan Paratte, David Shuman, Vasilis Kalofolias, Pierre Vandergheynst, and David K. Hammond, “GSPBOX: A toolbox for signal processing on graphs,” Aug. 2014.

²<https://lts2.epfl.ch/gsp/>.