2D Wavelet Animation Script User Manual

Daniel Smania

Version: 1.0 – Last updated: July 20, 2025

Contents

1 Introduction

This script generates a 2D animation of wavelet-based reconstructions of synthetic surfaces. It offers configurable color schemes, surface styles, and layout orientations to facilitate visualizations in educational and research contexts.

Key features:

- Animates 2D wavelet reconstructions incrementally.
- Offers various synthetic function types including smooth, discontinuous, Dirac, and mixed types.
- Configurable color schemes and visual styles.
- Produces MP4 or GIF outputs optionally.
- Logs command-line usage for reproducibility.

2 Prerequisites

- **Python** ≥ 3.8
- Required packages: `numpy`, `matplotlib`, `pywavelets`, `pillow`, `argparse`, `tkinter`, `scipy`.
- For MP4 export: FFmpeg in system `PATH`.

Install dependencies via:

```
pip install numpy matplotlib pywavelets pillow scipy
sudo apt install ffmpeg # For MP4 export
```

3 Quick Start

```
python wavelet2d.py
```

to generate an animation with default parameters.

```
python wavelet2d.py --wavelet_type haar --function_type smooth \
  --color_scheme vibrant --orientation portrait --save gif
```

This command produces a vibrant-colored portrait layout animation from a smooth synthetic function using `haar` wavelets.

4 Command-Line Options

`--frames_per_wavelet` Animation speed (frames per wavelet). Default: 3.

`--wavelet_type` Wavelet type from PyWavelets discrete wavelets (e.g., `haar`, `db4`, `coif4`). Default: `haar`.

`--function_type` Function type: `smooth`, `piecewise_linear`, `discontinuous`, `smooth_periodic`, `mix`, `dirac`. Default: `smooth`.

`--function_seed` Random seed for function generation. Default: 38324.

`--number_wavelets` Number of wavelets to animate. Default: 256.

`--save` Save animation format: `gif`, `mp4`. Default: `none` (display only).

`--grid_size` Grid size for 2D function (NxN). Default: 64.

`--color_scheme` Color scheme for reconstruction: `vibrant`, `gray`, `marine`, `ocean`. Default: `marine`.

`--original_style` Original function style: `transparent`, `wireframe`, `solid_contrast`, `gradient`, `oscillating_transparent`, `none`. Default: `wireframe`.

`--original_alpha` Base transparency of original function. Range: [0.0, 1.0]. Default: `0.9`.

`--oscillating_speed` Oscillation speed for original function transparency. Default: `0.05`.

`--orientation` Layout orientation: `landscape` (side-by-side), `portrait` (vertical stack). Default: `landscape`.

`--camera` Camera movement: `static_overview`, `rotating`. Default: `rotating`.

`--single_wavelet` Single wavelet animation type: 2d, 3d. Default: 3d.

Animation

When `-save` is `gif` or `mp4`, the script writes a file named:

`wavelet2d-<YYYY-MM-DD-HH-MM-SS>-<parameters>.(gif|mp4)`

Run History

Each execution appends a timestamped entry to `wavelet2d_history.log` with the command used and its parameters. Only the last 100 entries are preserved.

5 Examples

1. Marine palette preview

```
python wavelet2d.py --wavelet_type haar --color_scheme marine --save gif
```

6 Workflow Internals

1. Generate a 2D synthetic function `f_2d`.
2. Compute its wavelet decomposition with `pywt.wavedec2()`.
3. Initialize 3D plots using `matplotlib`'s surface tools.
4. Update the partial reconstruction frame-by-frame.
5. Display the original, cumulative, and single-wavelet surfaces.
6. Animate camera position and lighting.
7. Save result if requested.

7 Troubleshooting

No display Use `-save` to bypass GUI requirements.

Low performance Try `-grid_size 32` or a simpler color scheme.

No MP4 Ensure `ffmpeg` is in your system `PATH`.

8 Extending the Script

- Add new colormaps in `get_colormaps()`.
- Use Tkinter to build a GUI around this script.
- Load real datasets in place of `random_function_2d()`.

License