Readme document for Piece-Wise Spline Wigner-Ville Distribution (PW-WVD) MATLAB Package V1.0 [1]

The additional material in this file is provided by the authors to supplement the paper in [1]. The PW-WVD MATLAB package can be obtained using the permanent GitHub link in [2] or from the paper media section. The MATLAB scripts, functions, and datasets listed below, supplied within the PW-WVD MATLAB package, are used to produce results, and supporting figures illustrated in the paper:

Demo Scripts:

The developed PW-WVD package contains the following demo scripts within its directory:

1. *Demo_1_example_simulation.m*
   - Description: This demo script produces the results that are depicted in Fig. 2 of the paper.
   - Process: It generates an example 2-component non-stationary signal and computes its PW-WVD, WVD, and the WVD cross-terms. In addition, it shows the signal IA, IF, and time-support functions.
2. *Demo_2_tfd_comparison.m*
   - Description: This demo script produces the results that are depicted in Fig. 3 of the paper.
   - Process: It generates an example 2-component non-stationary signal and computes the signal ideal TFD, PW-WVD, and the MPD using chirplet atoms. Note that the MPD is precomputed and saved in *MPD_example.mat*.
3. *Demo_3_resolution_visualization.m*
   - Description: This demo script produces the illustration in Fig. 4 of the paper.
   - Process: It yields a one-dimensional visual interpretation for the proposed TFD resolution measure.
4. *Demo_4_measure_comparison.m*
   - Description: This demo script produces the results that are depicted in Figs. 5 and 6 of the paper.
   - Process: It compares the proposed average TFD performance with the NIR and Reinhold measures using the CKD of the example signal generated in *Demo_2_tfd_comparison.m*. In addition, it illustrates the TFD performance progression by computing the CKD at four increasing performance levels. Note that the CKD optimization is precomputed.
5. *Demo_5_performance_results.m*
   - Description: This demo script produces the results that are depicted in Fig. 7 and Table I of the paper.
   - Process: It yields the TFD performance evaluation results using the proposed measures. It uses the database signal parameters saved in *pw_wvd_database.mat* and the 12 TFD evaluations saved in *perf_pwvd.mat*, *perf_spwvd.mat*, *perf_ed.mat*, *perf_bjd.mat*, *perf_bd.mat*, *perf_mbd.mat*, *perf_embd.mat*, *perf_ckd.mat*, *perf_rgd.mat*, *perf_mdd.mat*, *perf_dgf.mat*, and *perf_mpd.mat*.
6. *Demo_6_computational_complexity.m*
   - Description: This demo script produces the results that are depicted in Fig. 8 of the paper.
   - Process: It produces the TFD computational complexity measured in terms of averaged processing time. It uses the 12 TFD computational complexity evaluations saved in *comtime_pwvd.mat*, *comtime_spwvd.mat*, *comtime_ed.mat*, *comtime_bjd.mat*, *comtime_bd.mat*, *comtime_mbd.mat*, *comtime_embd.mat*, *comtime_ckd.mat*, *comtime_rgd.mat*, *comtime_mdd.mat*, *comtime_dgf.mat*, and *comtime_mpd.mat*.

7. *Demo_7_inner_terms.m*
   - Description: This demo script produces the results that are depicted in Fig. A.1 of the paper.
   - Process: It generates an example mono-component non-stationary signal with non-linear IF and computes the WVD auto-terms and inner-terms using the formulation in Appendix A.

Main Scripts:

The developed PW-WVD package contains the following main scripts within its directory:

1. *Main_1_database_generation.m*
   - Description: This main script generates the database signals and parameters and saves them in *pw_wvd_database.mat*.
   - Process: It produces 1000 multi-component non-stationary signals sampled at 1 Hz and characterized by random number of components between 1 and 4, polynomial IF laws with random orders between 1 and 3, random constant IA between 0.5 and 1, and random time support within 0 and 255. The signal random time support is constrained with minimum and maximum IF curve lengths to produce signals with realistic durations.
2. *Main_2_compute_mp_atoms.m*
   - Description: This main script decomposes the database signals using the matching pursuit approach with chirplet atoms and saves the results in *MP_atoms.mat*.
   - Process: It generates a chirplet dictionary with approximately 6 million atoms and decomposes each signal in the database to reach a 1% minimum relative L_2 error.
3. *Main_3_optimization.m*
   - Description: This main script optimizes each quadratic TFD using the Bayesian optimization algorithm.
   - Process: The optimization is executed with random initial kernel parameters, for 200 iterations, and by using the expected improvement plus acquisition function. The optimal TFD parameters are then saved in *opt_pwvd.mat*, *opt_spwvd.mat*, *opt_ed.mat*, *opt_bjd.mat*, *opt_bd.mat*, *opt_mbd.mat*, *opt_embd.mat*, *opt_ckd.mat*, *opt_rgd.mat*, *opt_mdd.mat*, and *opt_dgf.mat*.
4. *Main_4_evaluation.m*
   - Description: This main script evaluates each optimized TFD using the proposed accuracy and resolution measures.
   - Process: The computed evaluations are saved in *perf_pwvd.mat*, *perf_spwvd.mat*, *perf_ed.mat*, *perf_bjd.mat*, *perf_bd.mat*, *perf_mbd.mat*, *perf_embd.mat*, *perf_ckd.mat*, *perf_rgd.mat*, *perf_mdd.mat*, *perf_dgf.mat*, and *perf_mpd.mat*.
5. *Main_5_computational_time.m*
   - Description: This main script evaluates the computational complexity of each optimized TFD in terms of processing time.
   - Process: The TFD computational complexity is estimated by Monte-Carlo simulations where the processing time of each optimized TFD is judged by generating the TFR of the 1000 test signals and repeating the process for 10 times. The TFD computational complexity evaluations are then saved in *comtime_pwvd.mat*, *comtime_spwvd.mat*, *comtime_ed.mat*, *comtime_bjd.mat*, *comtime_bd.mat*, *comtime_mbd.mat*, *comtime_embd.mat*, *comtime_ckd.mat*, *comtime_rgd.mat*, *comtime_mdd.mat*, *comtime_dgf.mat*, and *comtime_mpd.mat*.

Functions:

The developed PW-WVD package is comprised of the following MATLAB functions that are in specific folders within *Functions* directory:

1. *MP*
   - *chirplet_dictionary.m*: It generates the chirplet dictionary atoms.
   - *chirplet_parameters.m*: It generates the chirplet dictionary parameters.
   - *mp_atoms.m*: It decomposes an input signal using the Orthogonal Matching Pursuit algorithm.
   - *mpd.m*: It computes the time-frequency Matching Pursuit distribution.

2. *Optimization*
   - *objective_fun.m*: It holds the Bayesian optimization cost function.
   - *prepare_opt_var.m*: It prepares the Bayesian optimization variables.
   - *qtfd_opt.m*: It computes TFDs for optimization.
   - *tfd_param.m*: Getting TFDs parameters from the Bayesian optimizer.
   - *optimize_ckd_example.m*: Bayesian optimization for the example CKD.

3. *Performance*
   - *adatnirperformance.m*: It is the Reinhold measure implementation by Dr Isabella Reinhold in [3].
   - *tfd_measure.p*: This is a protected MATLAB function that computes the NIR measure. It is obtained from the TFSAP toolbox in [4].
   - *tfd_accuracy.m*: It computes the proposed TFD accuracy measure.
   - *tfd_resolution.m*: It computes the proposed TFD resolution measure.
   - *tfd_perf.m*: It computes the proposed TFD performance measure.

4. *TFD*
   - *qtfd_wvd.m*: The Wigner-Ville distribution.
   - *qtfd_dgf.m*: The directional Gaussian filter distribution. This is a modified implementation of the original code found in the supplementary material of [5].
   - *qtfd.m*: It computes any quadratic TFD using the kernels listed in *tf_kernel.m*.
   - *tf_kernel_bjd.m*: The Born-Jordan Doppler-lag kernel.
   - *tf_kernel_ed.m*: The exponential Doppler-lag kernel.
   - *tf_kernel_pwvd.m*: The pseudo WVD Doppler-lag kernel.
   - *tf_kernel_spwvd.m*: The smoothed-pseudo WVD Doppler-lag kernel.
   - *tf_kernel_bd.m*: The B-distribution Doppler-lag kernel.
   - *tf_kernel_mbd.m*: The modified B-distribution Doppler-lag kernel.
   - *tf_kernel_embd.m*: The extended modified B-distribution Doppler-lag kernel.
   - *tf_kernel_ckd.m*: The compact Doppler-lag kernel.
   - *tf_kernel_mdd.m*: The multi-directional Doppler-lag kernel. This is a modified implementation of the original code found in the supplementary material of [6].
   - *tf_kernel_rgd.m*: The radial Gaussian Doppler-lag kernel. This is a modified implementation of the original code found in the supplementary material of [6].
   - *tf_kernel_dgf.m*: The directional Gaussian filter process; interpreted as a spatial kernel. This is a modified implementation of the original code found in the supplementary material of [5].
   - *tf_kernel.m*: It computes the TFDs Doppler-lag kernels.

- *tf2af.m*: It transforms an input TFD to the Doppler-lag domain.
- *af2tf.m:* It transforms an input Ambiguity function to the TF domain.
- *filter_tfd.m*: It filters the WVD using the kernels listed in *tf_kernel.m*.
- *ideal_tfd.m*: It computes the ideal TFD for an input multi-component non-stationary signal.
- *pw_wvd.m*: It computes the proposed PW-WVD for an input multi-component non-stationary signal.

5. *Signal Generator*
   - *signal_parameters.m*: It produces the parameters of a mono-component non-stationary signal.
   - *signal_generator.m*: It generates a finite multi-component non-stationary signal with time-varying frequency and amplitude.

6. *Other*
   - *combs_rep.m*: It implements the multi-choose function; combinations with replacement. This function is a modified version of the code written by Matt Fig in https://mathworks.com/matlabcentral/fileexchange/24325-combinator-combinations-and-permutations.

Datasets:

The developed PW-WVD package is comprised of the following datasets that are in specific folders within *Data* directory:

1. *Database:* This folder holds the database signals and parameters in the file *pw_wvd_database.mat* that is generated by the main script *Main_1_database_generation.m*.
2. *MP:* This folder holds the MP parameters and MPD of the example signal generated by the demo script *Demo_2_tfd_comparison.m* in *MP_example.mat* and *MPD_example.mat*. Besides, it contains the MP decompositions of the database signals that are computed by the main script *Main_2_compute_mp_atoms.m* in *MP_atoms.mat*.
3. *Optimization:* This folder contains the optimal parameters for each TFD that are computed by the main script *Main_3_optimization.m*. The files are as follows: *opt_pwvd.mat*, *opt_spwvd.mat*, *opt_ed.mat*, *opt_bjd.mat*, *opt_bd.mat*, *opt_mbd.mat*, *opt_embd.mat*, *opt_ckd.mat*, *opt_rgd.mat*, *opt_mdd.mat*, and *opt_dgf.mat*.
4. *Evaluation:* This folder contains the TFD evaluations for each optimized TFDs that are computed by the main script *Main_4_evaluation.m*. The files are as follows: *perf_pwvd.mat*, *perf_spwvd.mat*, *perf_ed.mat*, *perf_bjd.mat*, *perf_bd.mat*, *perf_mbd.mat*, *perf_embd.mat*, *perf_ckd.mat*, *perf_rgd.mat*, *perf_mdd.mat*, *perf_dgf.mat*, and *perf_mpd.mat*.
5. *Computation time:* This folder contains the computational complexity evaluations for each optimized TFDs that are computed by the main script *Main_5_computational_time.m*. The files are as follows: *comtime_pwvd.mat*, *comtime_spwvd.mat*, *comtime_ed.mat*, *comtime_bjd.mat*, *comtime_bd.mat*, *comtime_mbd.mat*, *comtime_embd.mat*, *comtime_ckd.mat*, *comtime_rgd.mat*, *comtime_mdd.mat*, *comtime_dgf.mat*, and *comtime_mpd.mat*.

References

[1] M. Al-Sa'd, B. Boashash, and M. Gabbouj, "Design of an Optimal Piece-Wise Spline Wigner-Ville Distribution for TFD Performance Evaluation and Comparison", *IEEE Transactions on Signal Processing*, 2021.

[2] M. Al-Sa'd, B. Boashash, and M. Gabbouj, PW-WV MATLAB Package, Online (2021). URL: https://github.com/Al-Sad/PW-WVD.

[3] Reinhold, Isabella, and Maria Sandsten. "Optimal time–frequency distributions using a novel signal adaptive method for automatic component detection." *Signal Processing* 133 (2017): 250-259.

[4] Boualem Boashash, and Samir Ouelha. "Efficient software platform TFSAP 7.1 and MATLAB package to compute time–frequency distributions and related time-scale methods with extraction of signal characteristics." *SoftwareX* 8 (2018): 48-52.

[5] Boualem Boashash, Time-frequency signal analysis and processing toolbox, Online (2016). URL: http://booksite.elsevier.com/9780123984999/toolbox.php.

[6] Boashash, Boualem, and Samir Ouelha. "Designing high-resolution time–frequency and time–scale distributions for the analysis and classification of non-stationary signals: a tutorial review with a comparison of features performance." *Digital Signal Processing* 77 (2018): 120-152.