

pvx Complete HTML Documentation

Total HTML pages included: **23**

References appendix: **143** papers (minimum enforced: 100).

Generated: 2026-02-20 16:45:18 UTC.

Generated by `scripts_generate_docs_pdf.py`.

Contents

1. [01. pvx HyperText Markup Language \(HTML\) Documentation](#) (index.html)
2. [02. pvx Architecture](#) (architecture.html)
3. [03. pvx Mathematical Foundations](#) (math.html)
4. [04. pvx Window Reference](#) (windows.html)
5. [05. pvx Technical Glossary](#) (glossary.html)
6. [06. pvx Algorithm Limitations](#) (limitations.html)
7. [07. pvx Benchmark Report](#) (benchmarks.html)
8. [08. pvx Pipeline Cookbook](#) (cookbook.html)
9. [09. pvx Command-Line Interface \(CLI\) Flag Index](#) (cli_flags.html)
10. [10. pvx Citation Quality](#) (citations.html)
11. [11. pvx Group: Analysis, QA, and Automation](#) (groups/analysis_qa_and_automation.html)
12. [12. pvx Group: Creative Spectral Effects](#) (groups/creative_spectral_effects.html)
13. [13. pvx Group: Denoise and Restoration](#) (groups/denoise_and_restoration.html)
14. [14. pvx Group: Dereverb and Room Correction](#) (dereverb_and_room_correction.html)
15. [15. pvx Group: Dynamics and Loudness](#) (groups/dynamics_and_loudness.html)
16. [16. pvx Group: Granular and Modulation](#) (groups/granular_and_modulation.html)
17. [17. pvx Group: Pitch Detection and Tracking](#) (groups/pitch_detection_and_tracking.html)
18. [18. pvx Group: Retune and Intonation](#) (groups/retune_and_intonation.html)
19. [19. pvx Group: Separation and Decomposition](#) (groups/separation_and_decomposition.html)
20. [20. pvx Group: Spatial and Multichannel](#) (groups/spatial_and_multichannel.html)
21. [21. pvx Group: Spectral and Time-Frequency Transforms](#) (groups/spectral_time_frequency_transforms.html)
22. [22. pvx Group: Time-Scale and Pitch Core](#) (groups/time_scale_and_pitch_core.html)
23. [23. pvx Research Bibliography \(Phase Vocoder and Related Digital Signal Processing \(DSP\)\)](#) (papers.html)

pvx HyperText Markup Language (HTML) Documentation

Source: index.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

This HTML documentation is organized by algorithm folder/theme and generated from `src/pvx/algorithms/registry.py` and `src/pvx/algorithms/base.py` dispatch parameters.

Totals:

- **111** algorithms
- **12** folders/themes
- **87** distinct algorithm parameter keys in dispatch
- **103** linked technical glossary concepts
- **143** bibliography references

Quick Setup (Install + PATH)

```
python3 -m venv .venv
source .venv/bin/activate
python3 -m pip install -e .
pvx --help
```

If `pvx` is not found, add the virtualenv binaries to your shell path (`zsh`):

```
printf 'export PATH="%s/.venv/bin:$PATH"\n' "$(pwd)" >> ~/.zshrc
source ~/.zshrc
pvx --help
```

No-PATH fallback: `python3 pvx.py voc input.wav --stretch 1.2 --output output.wav`

Research bibliography (143 references) - foundational and directly related literature for phase vocoder, time-scale/pitch, time-frequency transforms, separation, denoising, dereverberation, spatial audio, and mastering workflows.

Linked technical glossary (103 terms) - quick definitions and external references (Wikipedia, standards pages, docs, and papers).

Mathematical foundations and **window reference (all 50 windows)**.

Architecture diagrams, algorithm limitations, benchmark report, pipeline cookbook, CLI flag index, and **citation quality report**.

GitHub usually shows `.html` files as source. For browser-rendered math on GitHub itself, use the Markdown mirrors: [docs/MATHEMATICAL_FOUNDATIONS.md](#) and [docs/WINDOW_REFERENCE.md](#).

Algorithm Groups

Folder	Theme	Algorithms	Page
time_scale_and_pitch_core	Time-Scale and Pitch Core	7	Open
pitch_detection_and_tracking	Pitch Detection and Tracking	8	Open
retune_and_intonation	Retune and Intonation	8	Open

Folder	Theme	Algorithms	Page
spectral_time_frequency_transforms	Spectral and Time-Frequency Transforms	8	Open
separation_and_decomposition	Separation and Decomposition	8	Open
denoise_and_restoration	Denoise and Restoration	8	Open
dereverb_and_room_correction	Dereverb and Room Correction	8	Open
dynamics_and_loudness	Dynamics and Loudness	8	Open
creative_spectral_effects	Creative Spectral Effects	8	Open
granular_and_modulation	Granular and Modulation	8	Open
analysis_qa_and_automation	Analysis, QA, and Automation	8	Open
spatial_and_multichannel	Spatial and Multichannel	24	Open

pvx Architecture

Source: architecture.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Architecture overview for runtime processing, algorithm dispatch, documentation generation, and CI/Pages publication.

Runtime and CLI Flow

Algorithm Registry and Dispatch

```
flowchart TD
    R[src/pvx/algorithms/registry.py] --> B[src/pvx/algorithms/base.py]
    B --> M1[time_scale_and_pitch_core/*]
    B --> M2[retune_and_intonation/*]
    B --> M3[dynamics_and_loudness/*]
    B --> M4[spatial_and_multichannel/*]
```

Documentation Build Graph

```
flowchart LR
    G1[scripts_generate_python_docs.py] --> D[docs/*]
    G2[scripts_generate_theory_docs.py] --> D
    G3[scripts_generate_docs_extras.py] --> D
    G4[scripts_generate_html_docs.py] --> H[docs/html/*]
    D --> H
```

CI and Pages

```
flowchart LR  
PR[Push / PR] --> CI[Doc and test workflow]  
CI --> S[Generation + drift checks]  
S --> T[Unit tests + docs coverage tests]  
T --> P[GitHub Pages deploy workflow]  
P --> SITE[Published docs site]
```

pvx Mathematical Foundations

Source: math.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Mathematical summary of the core signal-processing model used across pvx. Equations are rendered with MathJax when this page is opened in a normal browser.

GitHub usually displays HTML source text. For GitHub-native math rendering, use

[docs/MATHEMATICAL_FOUNDATIONS.md](#).

STFT Analysis and Synthesis

Analysis:

$$X_t[k] = \sum_{n=0}^{N-1} x[n + tH_a]w[n]e^{-j2\pi kn/N}$$

where `x` represents the signal (and `x[n]` represents sample index `n`), `t` represents frame index, `k` represents frequency-bin index, `w[n]` represents the analysis window, `H_a` / `H_s` represent analysis/synthesis hop sizes, `\phi` represents phase, `\omega` represents angular frequency, and `\varepsilon` represents a small numerical-stability constant.

Each frame `t` is windowed by `w[n]` and transformed to complex bins `k`.

Synthesis:

$$\hat{x}[n] = \frac{\sum_t \hat{x}_t[n - tH_s]w[n - tH_s]}{\sum_t w^2[n - tH_s] + \varepsilon}$$

where `x` represents the signal (and `x[n]` represents sample index `n`), `t` represents frame index, `k` represents frequency-bin index, `w[n]` represents the analysis window, `H_a` / `H_s` represent analysis/synthesis hop sizes, `\phi` represents phase, `\omega` represents angular frequency, and `\varepsilon` represents a small numerical-stability constant.

Overlap-add with energy normalization preserves level stability.

Transform Backend Selection (`--transform`)

pvx lets you choose the per-frame transform backend used in analysis/resynthesis: `fft`, `dft`, `czt`, `dct`, `dst`, and `hartley`.

Fourier family (`fft`, `dft`):

$$X_t[k] = \sum_{n=0}^{N-1} x_t[n]e^{-j2\pi kn/N}, \quad x_t[n] = \frac{1}{N} \sum_{k=0}^{N-1} X_t[k]e^{j2\pi kn/N}$$

where `x` represents the signal (and `x[n]` represents sample index `n`), `t` represents frame index, `k` represents frequency-bin index, `w[n]` represents the analysis window, `H_a` / `H_s` represent analysis/synthesis hop sizes, `\phi` represents phase, `\omega` represents angular frequency, and `\varepsilon` represents a small numerical-stability constant.

Chirp-Z (`czt`):

$$X_t[k] = \sum_{n=0}^{N-1} x_t[n]A^{-n}W^{nk}$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

With ppx defaults $A = 1$ and $W = e^{-j2\pi/N}$, CZT samples the DFT contour via a different numerical path.

DCT-II (`dct`):

$$C_t[k] = \alpha_k \sum_{n=0}^{N-1} x_t[n] \cos \left(\frac{\pi}{N} (n + \frac{1}{2}) k \right)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

DST-II (`dst`):

$$S_t[k] = \beta_k \sum_{n=0}^{N-1} x_t[n] \sin \left(\frac{\pi}{N} (n + \frac{1}{2})(k + 1) \right)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

Hartley (`hartley`):

$$H_t[k] = \sum_{n=0}^{N-1} x_t[n] \text{cas} \left(\frac{2\pi kn}{N} \right), \quad \text{cas}(\theta) = \cos \theta + \sin \theta$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

Transform	Why use it	Tradeoffs	Example use case
<code>fft</code>	Fastest and most robust default; best runtime and CUDA support.	Typical STFT leakage/time-resolution tradeoffs still apply.	General production stretch/pitch workflows.

Transform	Why use it	Tradeoffs	Example use case
dft	Reference Fourier baseline for transform parity checks.	Usually slower than fft with little audible benefit.	Verification and algorithm A/B testing.
czt	Alternative numerical path for awkward or prime frame sizes.	Requires SciPy; generally slower and CPU-oriented.	Edge-case frame-size validation and diagnostics.
dct	Real-basis energy compaction, useful for envelope-focused shaping.	No explicit complex phase bins; less transparent for strict phase coherence.	Creative timbre shaping and coefficient-domain experiments.
dst	Odd-symmetry real-basis alternative with distinct coloration.	Same phase limitations as DCT; artifacts are content-dependent.	Experimental percussive/spectral texture variants.
hartley	Real transform with Fourier-related basis for exploratory comparisons.	Different bin semantics from complex STFT can change artifact character.	Pedagogical and creative real-domain phase-vocoder tests.

Sample use cases

```
python3 pvxvoc.py dialog.wav --transform fft --time-stretch 1.08 --transient-preserve
--output-dir out
python3 pvxvoc.py tone_sweep.wav --transform dft --time-stretch 1.00 --output-dir out
python3 pvxvoc.py archival_take.wav --transform czt --n-fft 1531 --win-length 1531 --
hop-size 382 --output-dir out
python3 pvxvoc.py strings.wav --transform dct --pitch-shift-cents -17 --output-dir out
python3 pvxvoc.py percussion.wav --transform dst --time-stretch 0.92 --phase-locking
off --output-dir out
python3 pvxvoc.py synth_pad.wav --transform hartley --time-stretch 1.30 --output-dir
out
```

Phase-Vocoder Propagation

$$\Delta\phi_t[k] = \text{princarg}\left(\phi_t[k] - \phi_{t-1}[k] - \omega_k H_a\right)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

$$\hat{\omega}_t[k] = \omega_k + \frac{\Delta\phi_t[k]}{H_a}$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

$$\hat{\phi}_t[k] = \hat{\phi}_{t-1}[k] + \hat{\omega}_t[k]H_s$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

pvx uses these relationships to estimate true instantaneous frequency and re-accumulate phase under a new synthesis hop.

Pitch and Microtonal Mapping

$$r_{\text{pitch}} = 2^{\Delta s/12} = 2^{\Delta c/1200}$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

Semitone shift Δs and cents shift Δc are equivalent ratio controls. Scale-constrained retuning maps detected F0 to the nearest permitted scale target.

Dynamics and Loudness

$$g_{\text{LUFs}} = 10^{(L_{\text{target}} - L_{\text{in}})/20}$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

$$y[n] = x[n] \cdot g_{\text{LUFS}}$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

Target-loudness gain is applied in linear amplitude domain after dynamics stages, before limiting/clipping safety stages.

Spatial and Multichannel Highlights

Inter-channel phase difference:

$$\Delta\phi_{ij}(k, t) = \phi_i(k, t) - \phi_j(k, t)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

Coherence-preserving objective:

$$J = \sum_{k,t} |\Delta\phi_{ij}^{out}(k, t) - \Delta\phi_{ij}^{in}(k, t)|$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

Delay estimate:

$$\tau^* = \arg \max_{\tau} \mathcal{F}^{-1} \left\{ \frac{X_i(\omega) X_j^*(\omega)}{|X_i(\omega) X_j^*(\omega)| + \epsilon} \right\}(\tau)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

pvx spatial modules focus on channel coherence, stable image cues, and phase-vocoder-consistent multichannel processing for robust chain composition.

pvx Window Reference

Source: windows.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Complete pvx analysis-window reference. This page covers all **50** supported windows from `pvx.core.voc.WINDOW_CHOICES`, with formula-family mapping and practical interpretation.

For GitHub-native equation rendering, use [docs/WINDOW_REFERENCE.md](#).

Quantitative metrics and per-window SVG plots are generated from [docs/window_metrics.json](#) and [docs/assets/windows/*](#).

Window Formula Key

Let $n = 0, \dots, N - 1$, center index $m = (N - 1)/2$, and normalized coordinate $x_n = (n - m)/m$.

(W0) Rectangular:

$$w[n] = 1$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ε represents a small numerical-stability constant.

(W1) Cosine series:

$$w[n] = \sum_{k=0}^K a_k \cos\left(\frac{2\pi kn}{N-1}\right)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ε represents a small numerical-stability constant.

(W2) Sine/Cosine:

$$w[n] = \sin\left(\frac{\pi n}{N-1}\right)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ε represents a small numerical-stability constant.

(W3) Bartlett:

$$w[n] = 1 - \left| \frac{n-m}{m} \right|$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ε represents a small numerical-stability constant.

(W4) Triangular:

$$w[n] = \max\left(1 - \left| \frac{n-m}{(N+1)/2} \right|, 0\right)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

(W5) Bartlett-Hann:

$$x = \frac{n}{N-1} - \frac{1}{2}, \quad w[n] = 0.62 - 0.48|x| + 0.38 \cos(2\pi x)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

(W6) Tukey:

$$x = \frac{n}{N-1}, \quad w[n] = \begin{cases} \frac{1}{2}(1 + \cos(\pi(\frac{2x}{\alpha} - 1))), & 0 \leq x < \frac{\alpha}{2} \\ 1, & \frac{\alpha}{2} \leq x < 1 - \frac{\alpha}{2} \\ \frac{1}{2}(1 + \cos(\pi(\frac{2x}{\alpha} - \frac{2}{\alpha} + 1))), & 1 - \frac{\alpha}{2} \leq x \leq 1 \end{cases}$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

pvx special cases: $\alpha \leq 0$ is rectangular and $\alpha \geq 1$ becomes Hann.

(W7) Parzen:

$$\$u=\left|\frac{2n}{N-1}-1\right|, \quad w[n]=\begin{cases} 1-6u^2+6u^3, & 0 \leq u \leq \frac{1}{2} \\ 2(1-u)^3, & \frac{1}{2} < u \leq 1 \\ \frac{1}{2}, & 1 < u \leq 1 \end{cases}$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

(W8) Lanczos:

$$w[n] = \text{sinc}\left(\frac{2n}{N-1} - 1\right)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

(W9) Welch:

$$w[n] = \max(1 - x_n^2, 0)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

(W10) Gaussian:

$$w[n] = \exp\left(-\frac{1}{2}\left(\frac{n-m}{\sigma}\right)^2\right), \sigma = r_\sigma m$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

(W11) General Gaussian:

$$w[n] = \exp\left(-\frac{1}{2}\left|\frac{n-m}{\sigma}\right|^{2p}\right)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

(W12) Exponential:

$$w[n] = \exp\left(-\frac{|n-m|}{\tau}\right), \tau = r_\tau m$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

(W13) Cauchy:

$$w[n] = \frac{1}{1 + \left(\frac{n-m}{\gamma}\right)^2}, \gamma = r_\gamma m$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent

analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

(W14) Cosine power:

$$w[n] = \sin\left(\frac{\pi n}{N-1}\right)^p$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

(W15) Hann-Poisson:

$$w[n] = w_{\text{Hann}}[n] \exp\left(-\alpha \frac{|n-m|}{m}\right)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

(W16) General Hamming:

$$w[n] = \alpha - (1 - \alpha) \cos\left(\frac{2\pi n}{N-1}\right)$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

(W17) Bohman:

$$x = \left| \frac{2n}{N-1} - 1 \right|, \quad w[n] = (1-x) \cos(\pi x) + \frac{\sin(\pi x)}{\pi}$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ϵ represents a small numerical-stability constant.

(W18) Kaiser:

$$w[n] = \frac{I_0 \left(\beta \sqrt{1 - r_n^2} \right)}{I_0(\beta)}, \quad r_n = \frac{n - m}{m}$$

where x represents the signal (and $x[n]$ represents sample index n), t represents frame index, k represents frequency-bin index, $w[n]$ represents the analysis window, H_a / H_s represent analysis/synthesis hop sizes, ϕ represents phase, ω represents angular frequency, and ε represents a small numerical-stability constant.

Each window name in the table maps to one formula family above plus fixed constants/shape parameters.

Window	Family	Parameters	F	C	E	S	M	P	Plots	Plain-English behavior	Pros	Cons	Usage advice
hann	Cosine series (0.5, -0.5)	co-effs=(0.5, -0.5)	W1	0.49	1.50	-1.07	4.22	0.00	1.46	taper balancing leakage and resolution.	Balanced taper	Not optimal for amplitude suppression and frequency resolution.	Default choice for most pvx time-stretch and pitch-shift workflows.

Window	Friendly	Parameters	Form	Coeff	EN	S	M	P	Plot	Plain-English behavior	Pros	Cons	Usage advice
			mul	h	B	ca	ai	ea	s				
hamming	Cosine series	co-eff s=(0. 54, -0. 46)	W1	0.51	1.36339775	-1.70500000	4.00000000	-4.20000000	timetfreq	Cosine-sum taper balancing leakage and resolution.	Balanced leakage suppression and frequency resolution.	Not optimal for amplitude and extreme side-lobe rejection.	Default choice for most pvx time-stretch and pitch-shift workflows.
blackman	Cosine series	co-eff s=(0. 42, -0. 5, 0.0 8)	W1	0.41990179	1.76010790	-1.00980000	6.00000000	-5.80000000	timetfreq	Cosine-sum taper balancing leakage and resolution.	Strong sidelobe suppression for cleaner spectral separation.	Wide main lobe than Hann/Hamming.	Used for dense harmonic material when leakage artifacts dominate.

Window	Friendly	Parameters	Form	Center	Energy	Sigma	Mean	Peak	Plot	Plain-English behavior	Pros	Cons	Usage advice
bl ac km an ha rr is	Cosin eff s= (0. rie s 75, -0. 488 29, 0.1 412 8, -0. 011 68)	co- eff s= (0. rie s 75, -0. 488 29, 0.1 412 8, -0. 011 68)	W 1	0. 3 5 8 358 75,	2. 00 53 32 5	-0. .8 25 0 01 5	8. 00 0 1 /	-9. 2. 01 1 fr e q	ti m e / fr e q	Cosine-sum taper balancing leakage and resolution.	Strong side-lobe suppression for cleaner spectral separation.	Wide main lobe than Hann/Hamming.	Used for dense harmonic material when leakage artifacts dominate.

Window	Friendly	Parameters	Form	Center	Energy	Sigma	Major	Peak	Plot	Plain-English behavior	Pros	Cons	Usage advice
			mul	h	B	II	n-	lo	si				
nuttal	Cosine series	co-eff = (0.355768, -0.487396, 0.144232, -0.012604)	W1	0.355	2.20	-0.85	8.0	-9.325	timre/q	Cosine-sum taper balancing leakage and resolution.	Strong side-lobe suppression for cleaner spectral separation.	Wide main lobe than Hann/Hamming.	Used for dense harmonic material when leakage artifacts dominate.

Window	Friendly	Parameters	For mul a ke y	C o h e n g ai n	E N B (b in s)	S ca ll g lo ss (d B)	M ai lo h (bi ns)	P ea k o b e (d B)	Pl ot s	Plain-English behavior	Pros	Cons	Usage advice
flat-top	Cosine series	co-eff = (1, -1. 93, 1.2 9, -0. 388 , 0.0 322)	W 1	0. 9 9 9 5 1 4	3. 77 21 17 .0 .0 00	-0 .0 16	10 .0 31 1	-6 8. / fr e q	taper balancing leakage and resolution.	Very accurate amplitude estimation in FFT bins.	Very wide main lobe and reduced frequency discrimination.	Not fine pitch separation.	

Window	Friendly	Parameters	Form	Center	Energy	Sigma	Major	Peak	Plot	Plain-English behavior	Pros	Cons	Usage advice
			multiple	height	Bins	Width	Number of bins	Bin width	Overlap				
bl ac km an _n ut ta ll	Cosin e se (0. rie 363 s 582 , -0. 489 177 , 0.1 366 , -0. 010 641 1)	co- eff s= ()	W 1	0. 3 6 3 4 0 5	1. 97 70 73 .	-0 .8 50 0	8. 00 0 17 4	-9 8. 0 /	ti m e / fr e q	Cosine-sum taper balancing leakage and resolution.	Strong side-lobe suppression for cleaner spectral separation.	Wide main lobe than Hann/Hamming.	Use for dens e harmonic material when leakage artifacts dominate .

Window	Friendly	Parameters	For mul a ke y	C o h e n g ai n	E N B (b in s)	S ca ll g lo ss (d B)	M ai lo h (bi ns)	P ea k o b e (d B)	Pl ot s	Plain-English behavior	Pros	Cons	Usage advice
exact_bla ckman	Cosine coefficients	cosine coefficients (0.426591, -0.496561, 0.0768487)	W1	0.426591	1.694500	-1.2386	6.0000	-6.236	timereq	Cosine-sum taper balancing leakage and resolution.	Strong sidelobe suppression for cleaner spectral separation.	Wide main lobe than Hann/Hamming.	Used for dense harmonic material when leakage artifacts dominate.
sine	Sinusoidal	none	W2	0.623436309	1.234304	-2.0096	3.0000	-2.9999	timereq	Raised-sine taper; smooth edges with moderate main-lobe width.	Smooth endpoint behavior and straightforward implementation.	Less configurable than Kaiser/Tukey families.	Used for stable, low-complexity alternatives to Hann.

Window	Friendly	Parameters	Form	Coeff.	EN	Sigma	M	P	Plot	Plain-English behavior	Pros	Cons	Usage advice
bartlett	Triangular	none	W3	0.49756	1.39856	-1.82200	4.00000	-2.65230	timetfreq	Linear triangular taper; simple and fast.	Cheap linear taper with intuitive behavior.	Higher sidelobes than stronger cosine-sum windows.	Use for low-cost processing or quick exploratory runs.
boxicar	Rectangular	none	W00000	1.00000	1.00000	-3.92200	2.00000	-1.32640	timetfreq	No tapering; best frequency resolution but strongest leakage.	Narrowest main lobe and maximal bin sharpness.	Highest sidelobes and strongest leakage/phasiness on non-bin-centered content.	Used only for controlled test tones or when leakage is acceptable.

Window	Friendly	Parameters	Form	Center	EN	Sigma	Mean	Peak	Plot	Plain-English behavior	Pros	Cons	Usage advice
triangular	Triangular	none	W4	0.500244	1.332683	-1.802600	4.000000	-2.6523	timetfreq	Triangular taper variant with denominator $(N + 1)/2$.	Cheaper linear taper with intuitive behavior.	Higher sidelobes than stronger cosine-sum windows.	Use for low-cost processing or quick exploratory runs.
barttanner	Hybrid components	fixed	W5	0.499756	1.456559	-1.501700	4.000000	-3.5874	timetfreq	Blend of Bartlett-like slope and cosine curvature.	Moderate leakage suppression with light-weight computation.	Less common and leakageless interpretation than standard Hann/Hamming.	Used as a middle-ground taper when Bartlett feels too sharp.

Window	Friendly	Parameters	For m	C o ul a ke y	E h e nt g ai n	S h (b in s)	M ill o pi n g lo ss (d B)	P ai lo wi h (bi ns)	Pl ea k el o b e (d B)	Plain-English behavior	Pros	Cons	Usage advice
takey	Tuke y	alpha =0.5	W6	0.749634	1.228636	-2.2656	2.06	-1.5123	timereq	Cosine tapers at edges with flat center region.	Interpolates between rectangular and Hann behaviors.	Behaviors change strongly with alpha and can be inconsistent across presets.	Use when you need a controllable flat center with soft edges.

Window	Familly	Parameters	For m	C o ul a ke y	E h e nt g ai n	S h (b in s)	M ill o pi n g lo ss (d B)	P ai lo dt h (bi ns)	Pl ea k el o b e (d B)	Plain-English behavior	Pros	Cons	Usage advice
take_y_0p_1	Tuke_y	alpha=0.1	W6	0.949536	1.039289	-3.505	2.094	-1.309	tim/e/q	Cosine tapers at edges with flat center region.	Interpolates between rectangular and Hann behaviors.	Behavior changes strongly with alpha and can be inconsistent across soft edge sets.	Use when you need a controllable flat center with soft edges.

Window	Friendly	Parameters	For mul a ke y	C o h e n g ai n	E N B (b in s)	S ca ll o pi n g lo ss (d B)	M ai lo be wi h (bi ns)	P ea k d el o b e (d B)	Plot	Plain-English behavior	Pros	Cons	Usage advice
take_y_0p25	Tuke_y	alpha=0.25	W6	0.874573	1.102579	-2.9601	2.28601	-1.31601	timetaper/q	Cosine tapers at edges with flat center region.	Interpolates between rectangular and Hann behaviors.	Behavior changes strongly with alpha and can be inconsistent across presets.	Use when you need a controllable flat center with soft edges.

Window	Friendly	Parameters	Format	Color	EN	S	M	P	Plot	Plain-English behavior	Pros	Cons	Usage advice
take_y_0p75	Tuke_y	alpha=0.75	W6	0.624695	1.360628	-1.7188	3.95	-1.395	timetaper/q	Cosine tapers at edges with flat center region.	Interpolates between rectangular and Hann behaviors.	Behavior changes strongly with alpha and can be inconsistent across different pre-sets.	Use when you need a controllable flat center with soft edges.

Window	Friendly	Parameters	Form	Co	E	S	M	P	Plot	Plain-English behavior	Pros	Cons	Usage advice
			mul	h	B	II	ai	e	k				
tu ke y_0p 9	Tu ke y	al-pha =0. 9	W 6	0. 5 4 9 7 3 1	1. 44 69 88 .	-1 .5 21 5	3. 62 97 2	-2 4. 5 /	ti m e /fr e q	Cosine tapers at edges with flat center region.	Interpolates between rectangular and Hann behavior.	Behavior changes strongly with alpha and can be inconsistent across pre-sets.	Use when you need a controllable flat center with soft edge s.
pa rz en	Poly- no mi al	pie ce- wis e cu- bic	W 7	0. 3 7 4 8 1 7	1. 91 83 97 .	-0 .8 97 0	8. 00 0 04 6	-5 3. /fr e q	Smooth piecewise cubic window with strong sidelobe suppression.	Smooth taper with good sidelobe control.	Broad main lobe than light-weight windows.	Use for spectral analysis where sidelobe cleanup is critical.	

Window	Family	Parameters	Form	Coeff.	EN	Sigma	M	P	Plot	Plain-English behavior	Pros	Cons	Usage advice
lanczos	Sinc	none	W8	0.588202	1.299668	-1.896820	3.284052	-2.652025	timetfreq	Sinc-based taper with useful compromise for spectral analysis.	Sinc-derived shape with useful compromise behavior.	Can exhibit oscillatory behavior versus cosine-sum defaults.	Use for interpolation-adjacent analysis experiments.
welch	Quadratric	none	W9	0.666344	1.200587	-2.2355	2.871295	-2.15295	timetfreq	Parabolic taper emphasizing center samples.	Center-emphasizing parabolic with simple form.	Not as strong at side-lobe suppression as Blackman-family windows.	Used when center weighting is desired with minimal complexity.

Window	Friendly	Parameters	Form	Center	Energy	Sigma	Min	Peak	Plot	Plain-English behavior	Pros	Cons	Usage advice
			mul	h	B	II	ai	lo	si				
gausssi_an_0_p25	Gaussi안	sigma_ma_ratio=0.25	W10	0.31	2.25	-0.6	11.68	-8.06	tim	Bell-shaped taper; smaller sigma gives stronger edge attenuation.	Smooth bell taper with low ringing and predictable decay.	Frequent changes noticeably with sigma.	Use sigma presets to tune transient locality versus spectral leakage.
gausssi_an_0_p35	Gaussi안	sigma_ma_ratio=0.35	W10	0.43	1.62	-1.2	6.71	-5.29	tim	Bell-shaped taper; smaller sigma gives stronger edge attenuation.	Smooth bell taper with low ringing and predictable decay.	Frequent changes noticeably with sigma.	Use sigma presets to tune transient locality versus spectral leakage.

Window	Friendly	Parameters	Format	Color	EN	Sigma	Min	Peak	Plot	Plain-English behavior	Pros	Cons	Usage advice
gaussi안_0	Gau	sigma_ratio=0.	W10	0.5484	1.32058	-1.89	4.125	-3.636	timereq	Bell-shaped taper; smaller sigma gives stronger edge attenuation.	Smooth bell taper with low ringing and predictable decay.	Frequent changes noticeably with sigma.	Use sigma presets to tune transient locality versus spectral leakage.
gaussi안_0	Gau	sigma_ratio=0.	W10	0.6455	1.17181	-2.315	3.000	-2.890	timereq	Bell-shaped taper; smaller sigma gives stronger edge attenuation.	Smooth bell taper with low ringing and predictable decay.	Frequent changes noticeably with sigma.	Use sigma presets to tune transient locality versus spectral leakage.

Window	Friendly	Parameters	Format	Corner	EN	Slope	Margin	Poles	Plots	Plain-English behavior	Pros	Cons	Usage advice	
gausssi_an_0_p65	Gaussi_an	sig_ma_ratio=0.65	W10	0.713490	1.097655	-2.055749	2.625555	-2.575555	tim/q	Bell-shaped taper; smaller sigma gives stronger edge attenuation.	Smooth bell-taper with low ringing and predictable decay.	Freq	Use	sigma ma pre-sets to tune transient locality versus spectral leakage.
general_gausssi_an_-1_p5_0_p35	General_gaussian	power=1.5,sig_ma_ratio=0.35	W11	0.39355	2.01658	-0.784	5.15689	-2.4.5/q	tim/q	Adjustable shape exponent controls shoulder steepness.	Additional shape control beyond standard Gaussian.	Extra parameters increase tuning complexity.	Use for re-search/tuning tasks where shoulder steepness matters.	

Window	Familly	Parameters	For mul a ke y	C o h e n g ai n	E N B (b in s)	S ca ll g lo ss (d B)	M ai o pi n dt h (bi ns)	P ea k wi b e (d B)	Pl ot s	Plain-English behavior	Pros	Cons	Usage advice
general_gaussian_2_p0_0_p35	General Gaussian	power=2, sigma=0.35	W11	0.37	2.230016	-0.6281	5.91717	-1.71/q	timereq	Adjustable shape exponent controls shoulder steepness.	Additional shape control beyond standard Gaussian.	Extra parameters increase tuning complexity.	Use for research/tuning tasks where shoulder steepness matters.
general_gaussian_3_p0_0_p35	General Gaussian	power=3, sigma=0.35	W11	0.3644	2.5593	-0.532	5.9299	-1.6299/q	timereq	Adjustable shape exponent controls shoulder steepness.	Additional shape control beyond standard Gaussian.	Extra parameters increase tuning complexity.	Use for research/tuning tasks where shoulder steepness matters.

Window	Friendly	Parameters	For mul a ke y	C o h e n g ai n	E N B (b in s)	S ca ll g lo ss (d B)	M ai o pi n dt h (bi ns)	P ea k wi b e (d B)	Pl ot s	Plain-English behavior	Pros	Cons	Usage advice
general_gaussian_4_p0_0_p35	General Gaussian	power=4, sigma_ratio=0.	W11	0.35	2.529	-0.496	5.520	-1.060	timereq	Adjustable shape exponent controls shoulder steepness.	Additional shape control beyond standard Gaussian.	Extra parameters increase tuning complexity.	Use for research/tuning tasks where shoulder steepleness matters.
expontial_0p25	Exponential	tau_raiontial=0.25	W12	0.245	2.054	-1.022	15.62885	-3.1885	timereq	Symmetric exponential decay away from center.	Simple edge decay with fast computation.	Can produce less-uniform center weighting than cosine families.	Use for experiments emphasizing center-heavy weighting.

Window	Familly	Parameters	For mul a ke y	C o h e n g a i n	E N B (b in s)	S ca ll g n ss (d B)	M ai lo h (bi ns)	P ea k o b e (d B)	Pl ot s	Plain-English behavior	Pros	Cons	Usage advice
exponentia l_0p5	Exponential	tau _ratio =0.5	W 12	0.432187	1.332318	-2.0625	3.6250	-1.9100	tim /freq	Symmetric exponential decay away from center.	Simple edge decay with fast computation.	Can produce less-uniform center-weighting than cosine families.	Use for experiments emphasizing center-heav y weighting.
exponentia l_1p0	Exponential	tau _ratio =1	W 12	0.631991	0.08205599	-2.8544	2.5914	-2.01	tim /freq	Symmetric exponential decay away from center.	Simple edge decay with fast computation.	Can produce less-uniform center-weighting than cosine families.	Use for experiments emphasizing center-heav y weighting.

Window	Familly	Parameters	For mul a ke y	C o h e n g ai n	E N B (b in s)	S ca ll o g n ss (d B)	M ai lo h (bi ns)	P ea k o b e (d B)	Pl ot s	Plain-English behavior	Pros	Cons	Usage advice
ca uc hy _0 p5	Ca uc hy / Lo rent zia n	gam ma_ ra-tio =0.	W 13	0.5 3 4 0 2	1.22 97 75 0 2	-2.2 .09 2 3 3	3.56 2 73 3 / 3	-2 2.73 / fr e q	tim	Heavy-tailed taper with slower side decay than Gaussian.	Heavy tails pre-serv e more peripheral samples than Gaussian.	Tail energy can increase relative to steep taper s.	Use when you want softer attenuation of far-window samples.
ca uc hy _1 p0	Ca uc hy / Lo rent zia n	gam ma_ ra-tio =1	W 13	0.7 8 5 2 5 9	1.04 19 63 5 5 9	-3.1 08 5 4 4	2.37 5 03 4 4	-1 9.03 4 / fr e q	tim	Heavy-tailed taper with slower side decay than Gaussian.	Heavy tails pre-serv e more peripheral samples than Gaussian.	Tail energy can increase relative to steep taper s.	Use when you want softer attenuation of far-window samples.

Window	Friendly	Parameters	For mul a ke y	C o h e n g ai n	E N B (b in s)	S ca ll g lo ss (d B)	M ai lo dt h (bi ns)	P ea k o b e (d B)	Pl ot s	Plain-English behavior	Pros	Cons	Usage advice	
ca uc hy _2 p0	Ca uc hy / Lo rent zia n	gam ma _ratio =2	W 13	0.9 2 7 2 3 3	1.00 43 93 2 3 3	-3 .6 48 4 9	2.09 4.68 68 4 9	-1 4. / fr e q	tim	Heavy-tailed taper with slower side decay than Gaussian.	Heavy tails preserve more peripheral samples than Gaussian.	Tail energy can increase relative to steep taper.	Tail energy can increase relative to steep taper.	Use when you want softer attenuation of far-window samples.
co sine _pow er _2	Co sine power	pow er=2	W 14	0.4 9 9 7 5 6	1.50 07 33 22 0	-1 .4 22 0	4.00 1.46 8	-3 1.46 8	tim	Raises sine taper to power p ; higher p narrows effective support.	Simple powers can over-control edge steepness.	Higher powers can overly narrow effective support.	Used to smoothly increase edge attenuation versus basic sine.	

Window	Friendly	Parameters	Form	Center	EN	Slope	Main	Pearce	Plot	Plain-English behavior	Pros	Cons	Usage advice
			multi- agent key	center gain	(bin gai n)	(b in g lo ss (d B))	width loss (bi ns)	peak dt lo h ss (d B))	ots				
cosine_power_3	Cosine power	power=3	W14	0.42406	1.73572	-1.00740	5.00000	-3.92955	timetfreq	Raises sine taper to power p ; higher p narrows effective support.	Simple parameter controls edge steepness.	Higher powers can over-ly narrow effective support.	Used to smoothly increase edge attenuation versus basic sine.
cosine_power_4	Cosine power	power=4	W14	0.37481	1.94534	-0.86208	6.00000	-4.67411	timetfreq	Raises sine taper to power p ; higher p narrows effective support.	Simple parameter controls edge steepness.	Higher powers can over-ly narrow effective support.	Used to smoothly increase edge attenuation versus basic sine.

Window	Friendly	Parameters	Form	Coef	EN	Sig	Mul	Pow	Plot	Plain-English behavior	Pros	Cons	Usage advice
hann_poson_0_p5	Hann-Poisson	alpha=0.5	W15	0.432	1.609944	-1.2588	5.1824	-3.5245	timetfreq	Hann multiplied by exponential envelope for stronger edge decay.	Combines smooth Hann center with exponential edge suppression.	Can over-attenuate edges for high alpha values.	Use when you need stronger edge decay than Hann without full flat-top cost.

Window	Friendly	Parameters	Form	Coef	EN	Sigma	Mean	Peak	Plot	Plain-English behavior	Pros	Cons	Usage advice
hann_poson_1_p0	Hann-Poisson	alpha=1	W15	0.378	1.73419	-1.01750	10.307	-8.017	timetfreq	Hann multiplied by exponential envelope for stronger edge decay.	Combines smooth Hann center with exponential edge suppression.	Can over-attenuate edges for high alpha values.	Use when you need stronger edge decay than Hann without full flat-top cost.

Window	Familly	Parameters	For mul a ke y	C o m h e n g a i n	E N B (b in s)	S ca ll o g n ss (d B)	M ai lo h (bi ns)	P ea k d o b e (d B)	Pl ot s	Plain-English behavior	Pros	Cons	Usage advice
hann_poson_2_p0	Hann-Poisson	alpha=2	W15	0.297	2.0231787	-0.870719	16.4709	-8.059	timereq	Hann multiplied by exponential envelope for stronger edge decay.	Combines smooth Hann center with exponential edge suppression.	Can over-attenuate edges for high alpha values.	Use when you need stronger edge decay than Hann without full flat-top cost.

Window	Friendly	Parameters	For mul a ke y	C o m h e n g a i n	E N B (b in s)	S ca ll o g n ss (d B)	M ai lo h (bi ns)	P ea k d o b e (d B)	Pl ot s	Plain-English behavior	Pros	Cons	Usage advice
general_hammering_0p50	General Hammering	alpha=0.50	W16	0.49	1.50733	-1.40220	4.00468	-3.1.468	timereq	Hamming family with tunable cosine weight.	Tunable Hamming-style cosine weighting.	Hamming-classic Hann/Hamming choices.	Less standard than classic Hann/Hamming behavior near Hamming family defaults.

Window	Familly	Parameters	For mul a ke y	C o m h e n g ai n	E N B (b in s)	S ca ll o g n ss (d B)	M ai lo h (bi ns)	P ea k d o b e (d B)	Pl ot s	Plain-English behavior	Pros	Cons	Usage advice
general_hamm ing_0p 60	General Hamming	alpha =0.60	W 16	0.5 9 9 8 0 5	1.22 24 75	-2.1 79	3.46 9	-3.1 60 0	tim /fraq	Hamming family with tunable cosine weight.	Tunable Hamming - style cosine weighting.	Hamming than classic Hann /Hamming choices.	Less standardized than classic Hamming behavior near Hamming family defaults .

Window	Friendly	Parameters	For mul ake y	C o h e n g ai n	E N B (b in s)	S ca ll o pi n g lo ss (d B)	M ai lo be wi h (bi ns)	P ea k d o b e (d B)	Pl ot s	Plain-English behavior	Pros	Cons	Usage advice	
general_hammering_0p70	General Hammering	alpha=0.70	W16	0.699	1.0919854	-2.7626	2.656	-2.078	timere/q	Hamming family with tunable cosine weight.	Tunable Hamming-style cosine weighting.	Standardized than classic Hann/Hamming choices.	Less standard than classic Hann/Hamming behaviors.	Use to sweep side-lobes vs-width near Hamming family defaults.

Window	Familly	Parameters	For mul a ke y	C o h e n g ai n	E N B (b in s)	S ca ll o g lo ss (d B)	M ai lo h (bi ns)	P ea k o b e (d B)	Pl ot s	Plain-English behavior	Pros	Cons	Usage advice
general_hamm ing_0p 80	General Hamming	alpha =0.80	W 16	0.79902	1.03129	-3.272	2.31649	-1.649	ti me /fr e q	Hamming family with tunable cosine weight.	Tunable Hamming-style cosine weighting.	Less standard than classic Hann/Hamming choices.	Use to sweep side-lobes vs-width behavior near Hamming family defaults.
bohan	Bohaman	none	W 17	0.405087	1.786613	-1.00220	6.00997	-4.5.7	ti me /fr e q	Continuous slope window with cosine plus sine correction.	Continuous slope taper with good qualitative leakage control.	Less common in audio tooling and harder to tune by intuition.	Used for exploratory spectral work need ing smooth derivatives.

Window	Friendly	Parameters	Form	Coef	EN	S	M	P	Plot	Plain-English behavior	Pros	Cons	Usage advice
			m <ul style="list-style-type: none">akey	h <ul style="list-style-type: none">entgain	B <ul style="list-style-type: none">(bins)	W <ul style="list-style-type: none">opigloss(dB)	I <ul style="list-style-type: none">n-lodthns)	C <ul style="list-style-type: none">aibewibe(dB)	ea <ul style="list-style-type: none">kdelbe)	Plain-English behavior			

Window	Friendly	Parameters	For mul a ke y	C o h e n g ai n	E N B (b in s)	S ca ll o g n ss (d B)	M ai lo h (bi ns)	P ea k o b e (d B)	Pl ot s	Plain-English behavior	Pros	Cons	Usage advice
ka is er	Ka ise r- Be ss el	bet a fro m – kai ser – bet a	W 18	0. 3 3 1 7 0 8	2. 16 21 81	-0 .7 12 5	9. 05 .9 21	-1 / fr e q	ti m e	Adjustable trade-off window using modified Bessel function.	Cont inuously tunable width/ sidelobes	Need s beta tuning; poor choices can overblur or undersuppress sidelobes.	Use when you need explicit control over leakage versus resolution.

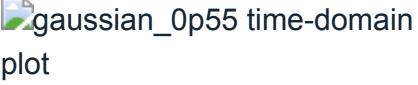
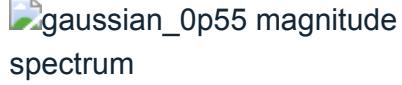
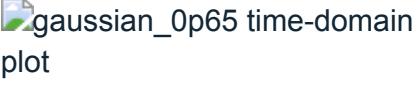
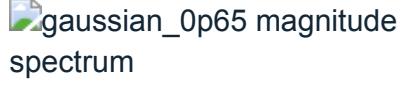
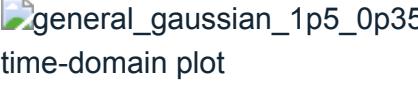
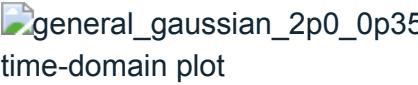
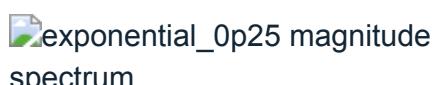
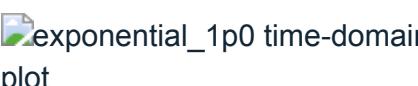
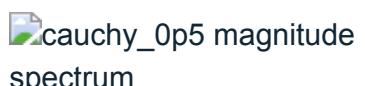
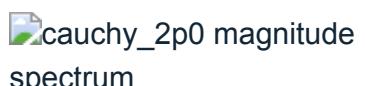
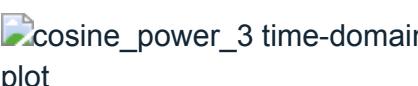
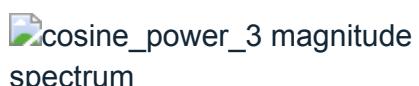
Window	Friendly	Parameters	F	C	E	S	M	P	Plots	Plain-English behavior	Pros	Cons	Usage advice
			for mul a ke y	co h er e nt g ai n	N B (b in s)	ca ll o pi n g lo ss (d B)	ai lo be wi dt h (bi ns)	ea k si el o b e (d B)					
rect	Rectangular	none	W0	1.00000	1.00000	-3.92200	2.00000	-1.32640	timetfreq	Alias of boxcar in pvx.	Narrowest main lobe and strongest leakage/bin sharpness.	Highest sidelobes and leakages or when leakage is acceptable.	Use only for controlled test tones or when leakage is acceptable.

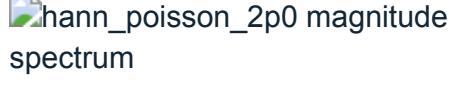
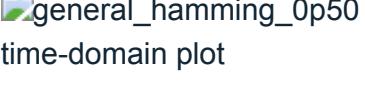
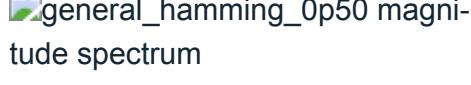
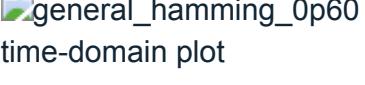
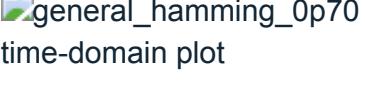
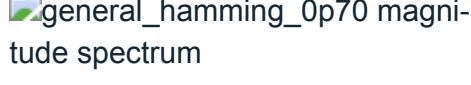
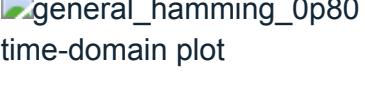
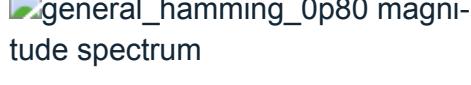
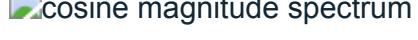
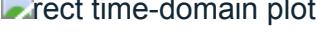
Complete Plot Gallery (All Windows)

Time-domain and frequency-magnitude plots for every supported window.

Window	Time-domain shape	Magnitude spectrum
hann		
hamming		
blackman		
blackmanharris		

Window	Time-domain shape	Magnitude spectrum
nuttall		
flattop		
blackman_nuttall		
exact_blackman		
sine		
bartlett		
boxcar		
triangular		
bartlett_hann		
tukey		
tukey_0p1		
tukey_0p25		
tukey_0p75		
tukey_0p9		
parzen		
lanczos		
welch		
gaussian_0p25		
gaussian_0p35		
gaussian_0p45		

Window	Time-domain shape	Magnitude spectrum
gaussian_0p55		
gaussian_0p65		
general_gaussian_1p5_0p35		
general_gaussian_2p0_0p35		
general_gaussian_3p0_0p35		
general_gaussian_4p0_0p35		
exponential_0p25		
exponential_0p5		
exponential_1p0		
cauchy_0p5		
cauchy_1p0		
cauchy_2p0		
cosine_power_2		
cosine_power_3		
cosine_power_4		
hann_poisson_0p5		

Window	Time-domain shape	Magnitude spectrum
<code>hann_poisson_1p0</code>		
<code>hann_poisson_2p0</code>		
<code>general_hamming_0p50</code>		
<code>general_hamming_0p60</code>		
<code>general_hamming_0p70</code>		
<code>general_hamming_0p80</code>		
<code>bohman</code>		
<code>cosine</code>		
<code>kaiser</code>		
<code>rect</code>		

pvx Technical Glossary

Source: glossary.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Linked glossary for core concepts used throughout pvx algorithms, CLIs, and research docs. Entries include concise definitions plus external references (Wikipedia, standards pages, project docs, and canonical papers).

Total terms: **103** across **10** categories.

Categories

- Core DSP Math (15)

- Phase and Time-Scale DSP (11)
- Pitch and Intonation (12)
- Time-Frequency Transforms (8)
- Separation and Decomposition (8)
- Denoising and Restoration (13)
- Dynamics and Mastering (10)
- Creative Spectral Effects (8)
- Spatial Audio and Multichannel (10)
- Analysis and QA (8)

Core DSP Math

Term	Description	External link
Aliasing	Frequency-domain folding caused by undersampling. Unchecked aliasing folds out-of-band content back into the audible band, often creating false partials or metallic artifacts.	Reference
Blackman window	Cosine-sum window with stronger sidelobe suppression. In pvx workflows this directly affects spectral resolution, leakage behavior, and overlap-add reconstruction accuracy in STFT-based processing.	Reference
Convolution	Linear filtering operation in time domain equivalent to multiplication in frequency domain. In pvx workflows this directly affects spectral resolution, leakage behavior, and overlap-add reconstruction accuracy in STFT-based processing.	Reference
Fast Fourier transform (FFT)	Efficient algorithm for computing the discrete Fourier transform. Its computational efficiency is critical because pvx repeatedly computes FFTs frame-by-frame across channels and files.	Reference
Fourier transform	Transforms a signal from time domain to frequency domain. In pvx workflows this directly affects spectral resolution, leakage behavior, and overlap-add reconstruction accuracy in STFT-based processing.	Reference
Hamming window	Cosine-sum window with reduced nearest sidelobe amplitude. Compared with Hann it slightly reduces nearest sidelobes, often at the cost of different amplitude bias characteristics.	Reference
Hann window	Raised cosine window commonly used in STFT analysis. It is a common default because it offers a strong practical balance between leakage control and frequency resolution.	Reference

Term	Description	External link
Inverse FFT (IFFT)	Inverse operation that reconstructs time-domain samples from spectral bins. Accurate inverse transforms are required to reconstruct low-artifact audio after spectral-domain edits.	Reference
Kaiser window	Parametric window controlled by beta parameter. The beta parameter gives explicit control over the width-versus-sidelobe tradeoff, which is useful for task-specific tuning.	Reference
Linear phase	Filter phase response that preserves waveform shape of band-limited signals. In ppx workflows this directly affects spectral resolution, leakage behavior, and overlap-add reconstruction accuracy in STFT-based processing.	Reference
Minimum phase	System with minimum group delay for a given magnitude response. In ppx workflows this directly affects spectral resolution, leakage behavior, and overlap-add reconstruction accuracy in STFT-based processing.	Reference
Nyquist frequency	Half the sample rate; highest representable frequency in sampled audio. In ppx workflows this directly affects spectral resolution, leakage behavior, and overlap-add reconstruction accuracy in STFT-based processing.	Reference
Overlap-add (OLA)	Frame-based reconstruction by summing overlapped windowed segments. In ppx workflows this directly affects spectral resolution, leakage behavior, and overlap-add reconstruction accuracy in STFT-based processing.	Reference
Spectral leakage	Energy spread across bins due to finite-duration analysis windows. In ppx workflows this directly affects spectral resolution, leakage behavior, and overlap-add reconstruction accuracy in STFT-based processing.	Reference
Window function	Tapering function applied before spectral analysis to control leakage. Window shape selection is one of the strongest controls on sidelobe suppression versus main-lobe width.	Reference

Phase and Time-Scale DSP

Term	Description	External link
Formant preservation	Preservation of spectral envelope while pitch is changed. This helps avoid chipmunk/monster timbre artifacts when pitch is shifted significantly.	Reference

Term	Description	External link
Granular synthesis	Sound synthesis/modification using short grains. In ppx this concept determines how frames are aligned, phases are propagated, and transients are preserved during time-stretch and pitch operations.	Reference
HPSS	Harmonic-percussive source separation in TF domain. Harmonic and percussive components can then be processed with different settings to preserve punch while shaping sustain.	Reference
Identity phase locking	Phase-locking variant that follows dominant peaks. In ppx this concept determines how frames are aligned, phases are propagated, and transients are preserved during time-stretch and pitch operations.	Reference
LP-PSOLA	Linear-prediction-assisted PSOLA variant. Linear-prediction support can improve source/filter control and reduce timbral drift in speech-like material.	Reference
Phase locking	Locking neighboring bin phases to preserve local waveform structure. Locking strategies reduce local phase incoherence, improving clarity and reducing chorus-like blur in stretched material.	Reference
Phase vocoder	STFT-based time-scale/pitch framework using phase propagation. It underpins most ppx core transforms by decoupling time-scale and pitch behavior through phase-consistent spectral resynthesis.	Reference
TD-PSOLA	Time-domain pitch-synchronous overlap-add technique. It is effective for voiced monophonic content where pitch periods are trackable and edits should remain time-domain natural.	Reference
Time warp map	Piecewise mapping from output time to input time. In ppx this concept determines how frames are aligned, phases are propagated, and transients are preserved during time-stretch and pitch operations.	Reference
Transient preservation	Strategies that protect attacks during spectral time modification. Without transient-aware handling, attacks can smear, so this is central for drums and articulation-critical material.	Reference
WSOLA	Waveform-similarity overlap-add time-scale modification. It is often preferred for moderate speech/music time changes when waveform continuity is more important than heavy spectral editing.	Reference

Pitch and Intonation

Term	Description	External link
Cents	Logarithmic pitch interval unit (1200 cents per octave). Because cents are logarithmic, fixed cent offsets represent musically consistent relative pitch intervals anywhere in the register.	Reference
Equal temperament	Tuning system dividing octave into equal semitone ratios. It is the default modern tuning grid but may be intentionally replaced for historical or microtonal work.	Reference
Harmonic product spectrum	Downsample-and-multiply spectrum for F0 evidence. In ppx retune pipelines this governs pitch-target decisions, microtonal mapping behavior, and the stability of note-to-note correction.	Reference
Just intonation	Tuning based on low-integer frequency ratios. Its pure-ratio intervals can sound more consonant in context but depend strongly on key and harmonic function.	Reference
MIDI Tuning Standard	Specification for non-12TET tuning in MIDI systems. This enables compatible systems to exchange alternate tunings without redefining note identities.	Reference
Portamento	Continuous slide between notes. Retune systems often need slide-aware logic so expressive glides are not flattened into abrupt quantized jumps.	Reference
pYIN	Probabilistic extension of YIN with voicing model. Its probabilistic voicing model can improve contour stability relative to raw framewise estimates.	Reference
RAPT	Robust algorithm for pitch tracking. In ppx retune pipelines this governs pitch-target decisions, microtonal mapping behavior, and the stability of note-to-note correction.	Reference
Scala scale format	Text format for microtonal scales. Using Scala files allows precise import of non-12TET interval maps into repeatable production workflows.	Reference
Subharmonic summation	Weighted subharmonic summation for F0 estimation. In ppx retune pipelines this governs pitch-target decisions, microtonal mapping behavior, and the stability of note-to-note correction.	Reference
SWIPE	Sawtooth-waveform inspired pitch estimator. It emphasizes sawtooth-like harmonic structure, which can improve robustness on certain voiced spectra.	Reference
YIN	Fundamental-frequency estimator using difference-function minima. It is commonly used as a robust baseline detector for monophonic F0 tracks.	Reference

Time-Frequency Transforms

Term	Description	External link
Chirplet transform	Transform using chirped atoms. In pvx transform modules this shapes the tradeoff between time localization, frequency precision, and invertibility of analysis/resynthesis steps.	Reference
Constant-Q transform (CQT)	Transform with logarithmic center frequencies and constant Q. Its logarithmic spacing is especially useful for music analysis where pitch perception is logarithmic.	Reference
Multiresolution analysis	Nested subspace structure for multi-scale representations. In pvx transform modules this shapes the tradeoff between time localization, frequency precision, and invertibility of analysis/resynthesis steps.	Reference
NSGT	Nonstationary Gabor transform with invertible adaptive filter banks. NSGT offers adaptive, invertible filter-bank behavior beyond fixed-resolution STFT framing.	Reference
Reassigned spectrogram	Sharper TF representation by moving energy to local centroids. Reassignment can sharpen time-frequency energy localization for clearer ridge and onset interpretation.	Reference
Synchrosqueezing	Post-processing that concentrates TF energy along ridges. It can concentrate ridge energy for improved instantaneous-frequency interpretation.	Reference
Variable-Q transform (VQT)	Generalized CQT with variable Q and bandwidth behavior. Compared to CQT, VQT broadens control over Q and bandwidth behavior across frequency ranges.	Reference
Wavelet packet	Hierarchical wavelet decomposition of both low/high branches. It supports multi-band decomposition where both approximation and detail branches are recursively analyzed.	Reference

Separation and Decomposition

Term	Description	External link
CP decomposition	Canonical polyadic tensor decomposition. In pvx decomposition workflows this assumption controls source isolation quality, bleed rejection, and the interpretability of separated components.	Reference
Demucs	Deep waveform-domain music source separation architecture. Neural stem separation quality can be high, but behavior depends heavily on training-domain match.	Reference

Term	Description	External link
ICA	Blind source separation based on statistical independence. In ppx decomposition workflows this assumption controls source isolation quality, bleed rejection, and the interpretability of separated components.	Reference
NMF	Nonnegative factorization of matrices into parts-based components. Factor count and constraints strongly influence whether components are musically meaningful or overly fragmented.	Reference
RPCA	Low-rank + sparse decomposition robust to outliers. It is frequently used to separate stable backgrounds from sparse foreground events.	Reference
Tensor decomposition	Higher-order extension of matrix factorization. In ppx decomposition workflows this assumption controls source isolation quality, bleed rejection, and the interpretability of separated components.	Reference
Tucker decomposition	Core tensor with mode matrices factorization. In ppx decomposition workflows this assumption controls source isolation quality, bleed rejection, and the interpretability of separated components.	Reference
U-Net	Encoder-decoder CNN with skip connections. In ppx decomposition workflows this assumption controls source isolation quality, bleed rejection, and the interpretability of separated components.	Reference

Denoising and Restoration

Term	Description	External link
Blind deconvolution	Joint estimation of source and room/filter response. In ppx restoration chains this controls denoise or dereverb strength versus artifact risk, helping balance intelligibility and texture retention.	Reference
Declick	Removal of impulsive click artifacts. In ppx restoration chains this controls denoise or dereverb strength versus artifact risk, helping balance intelligibility and texture retention.	Reference
Declip	Reconstruction of clipped waveform segments. In ppx restoration chains this controls denoise or dereverb strength versus artifact risk, helping balance intelligibility and texture retention.	Reference
Dereverberation	Suppression of late reverberation components. In ppx restoration chains this controls denoise or dereverb strength versus artifact risk, helping balance intelligibility and texture retention.	Reference

Term	Description	External link
Diffusion model	Generative model based on iterative denoising processes. In ppx restoration chains this controls denoise or dereverb strength versus artifact risk, helping balance intelligibility and texture retention.	Reference
Direct-to-reverberant ratio (DRR)	Ratio of direct sound energy to reverberant energy. In ppx restoration chains this controls denoise or dereverb strength versus artifact risk, helping balance intelligibility and texture retention.	Reference
Log-MMSE	Log-spectral MMSE enhancement estimator. Its log-domain objective often better tracks perceptual sensitivity than raw-amplitude objectives.	Reference
Minimum statistics	Noise tracking from low percentile/minimum spectral statistics. In ppx restoration chains this controls denoise or dereverb strength versus artifact risk, helping balance intelligibility and texture retention.	Reference
MMSE-STSA	MMSE short-time spectral amplitude estimator. It estimates clean spectral amplitudes from noisy observations while minimizing expected short-time error.	Reference
RNNoise	Hybrid recurrent neural + DSP speech denoiser. In ppx restoration chains this controls denoise or dereverb strength versus artifact risk, helping balance intelligibility and texture retention.	Reference
Spectral subtraction	Subtracts estimated noise spectrum from noisy signal. In ppx restoration chains this controls denoise or dereverb strength versus artifact risk, helping balance intelligibility and texture retention.	Reference
Weighted prediction error (WPE)	Multi-channel dereverb by delayed linear prediction. WPE is a standard dereverberation baseline for suppressing late reverberant tails in multi-channel settings.	Reference
Wiener filter	MMSE linear estimator for noisy observations. It is a classic linear MMSE approach and still valuable as a reliable baseline enhancer.	Reference

Dynamics and Mastering

Term	Description	External link
Compressor	Combined compressor and expander behavior. In ppx mastering stages this affects loudness compliance, peak containment, and perceived punch versus transparency across final renders.	Reference

Term	Description	External link
Compressor	Dynamic range processor that reduces gain above threshold. Threshold, ratio, attack, and release jointly define how quickly and how strongly level is controlled.	Reference
EBU R128	European loudness standard for production and broadcast. R128 operationalizes BS.1770 loudness measurement into practical broadcast and production targets.	Reference
Expander	Dynamic processor that reduces gain below threshold. It can reduce low-level noise floors and room tail audibility by attenuating content below threshold.	Reference
Hard clipping	Abrupt saturation by truncating waveform beyond threshold. In pvx mastering stages this affects loudness compliance, peak containment, and perceived punch versus transparency across final renders.	Reference
ITU-R BS.1770	International loudness measurement recommendation. Its weighting and gating model is the industry baseline for objective integrated loudness measurement.	Reference
Limiter	High-ratio compressor preventing peaks from exceeding threshold. In mastering chains it is commonly the final safety stage to prevent over-level peaks.	Reference
LUFS	Loudness unit relative to full scale. Targeting LUFS helps keep perceived program loudness consistent across tracks and delivery contexts.	Reference
Soft clipping	Smooth saturation curve reducing harsh distortion. In pvx mastering stages this affects loudness compliance, peak containment, and perceived punch versus transparency across final renders.	Reference
True peak	Peak estimate including inter-sample overs. True-peak metering estimates inter-sample peaks that sample-peak meters can miss.	Reference

Creative Spectral Effects

Term	Description	External link
Amplitude modulation (AM)	Modulating signal amplitude by a control waveform. In pvx creative processing this primarily controls intentional timbral coloration and motion, not transparent corrective behavior.	Reference
Cross synthesis	Combining spectral magnitude and phase or envelopes across signals. Different choices of magnitude/phase or envelope transfer produce very different timbral identities.	Reference

Term	Description	External link
Formant	Resonance region of spectral envelope. In pvx creative processing this primarily controls intentional timbral coloration and motion, not transparent corrective behavior.	Reference
Frequency modulation (FM)	Modulating oscillator frequency by another signal. In pvx creative processing this primarily controls intentional timbral coloration and motion, not transparent corrective behavior.	Reference
LFO	Low-frequency oscillator used for modulation. In pvx creative processing this primarily controls intentional timbral coloration and motion, not transparent corrective behavior.	Reference
Resonator	Filter structure emphasizing a narrow band. In pvx creative processing this primarily controls intentional timbral coloration and motion, not transparent corrective behavior.	Reference
Ring modulation	Multiplication by a carrier to produce sidebands. Sideband generation can create inharmonic or bell-like textures depending on carrier/modulator relationships.	Reference
Spectral freezing	Holding a short-time spectrum and resynthesizing over time. Freeze methods hold a selected spectral snapshot and are useful for drones, pads, and transitions.	Reference

Spatial Audio and Multichannel

Term	Description	External link
Binaural rendering	Headphone rendering preserving spatial localization cues. HRTF quality and listener assumptions strongly affect externalization and localization realism.	Reference
Crosstalk cancellation	Loudspeaker technique reducing opposite-ear leakage. In pvx multichannel workflows this directly influences localization cues, interchannel coherence, and translation to speaker or headphone playback.	Reference
DBAP	Distance-based amplitude panning. DBAP generalizes panning by amplitude weighting from geometric distance rather than simplex triplets.	Reference
Diffuse field	Sound field with many uncorrelated incoming directions. In pvx multichannel workflows this directly influences localization cues, interchannel coherence, and translation to speaker or headphone playback.	Reference

Term	Description	External link
HRTF	Head-related transfer function per source direction. In pvx multichannel workflows this directly influences localization cues, interchannel coherence, and translation to speaker or headphone playback.	Reference
Interaural level difference (ILD)	Level difference cue for localization. In pvx multichannel workflows this directly influences localization cues, interchannel coherence, and translation to speaker or headphone playback.	Reference
Interaural time difference (ITD)	Arrival-time difference cue for localization. In pvx multichannel workflows this directly influences localization cues, interchannel coherence, and translation to speaker or headphone playback.	Reference
Mid/side processing	Stereo representation as sum (mid) and difference (side). In pvx multichannel workflows this directly influences localization cues, interchannel coherence, and translation to speaker or headphone playback.	Reference
Room impulse response (RIR)	Acoustic response from source impulse to receiver. In pvx multichannel workflows this directly influences localization cues, interchannel coherence, and translation to speaker or headphone playback.	Reference
VBAP	Vector-base amplitude panning for loudspeaker arrays. VBAP is widely used for loudspeaker panning with controllable phantom-image placement.	Reference

Analysis and QA

Term	Description	External link
Bayesian optimization	Global optimization of expensive objective functions. It is useful when parameter spaces are expensive to evaluate and objective surfaces are noisy.	Reference
Beat tracking	Estimating temporal pulse positions from audio. In pvx analysis and automation tools this determines feature reliability and how confidently metrics can drive downstream decisions.	Reference
Chord estimation	Inferring harmonic labels from tonal features. In pvx analysis and automation tools this determines feature reliability and how confidently metrics can drive downstream decisions.	Reference

Term	Description	External link
Onset detection	Detection of transient musical events. In pvx analysis and automation tools this determines feature reliability and how confidently metrics can drive downstream decisions.	Reference
PESQ	Perceptual Evaluation of Speech Quality metric. PESQ is speech-focused and should be interpreted alongside task context and listening tests.	Reference
STOI	Short-Time Objective Intelligibility measure. STOI is an intelligibility-oriented metric and does not fully capture timbral preference.	Reference
Structure segmentation	Dividing music into sections like verse/chorus. In pvx analysis and automation tools this determines feature reliability and how confidently metrics can drive downstream decisions.	Reference
ViSQOL	Virtual Speech Quality Objective Listener metric family. ViSQOL provides perceptual similarity estimates and is often used with complementary quality metrics.	Reference

pvx Algorithm Limitations

Source: limitations.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Limitations guidance for all algorithm groups and algorithm IDs.

Group Summary

Group	Assumptions	Failure modes	When not to use
analysis_qa_and_automation	Feature extraction settings align with domain (speech vs music etc.).	False positives/negatives under domain shift.	Avoid treating single metrics as absolute quality verdicts.

Group	Assumptions	Failure modes	When not to use
cre- ative_spec- tral_ef- fects	Spectral manipulations are desired even with timbral coloration.	Can introduce intentional but strong coloration or temporal artifacts.	Avoid for transparent restoration/mastering paths.
de- noise_and_r estoration	Noise/artifacts are distinguishable from desired signal statistics.	Over-reduction can remove detail and create modulation artifacts.	Avoid high reduction settings on sparse acoustic sources without auditioning.
derever- b_and_- room_cor- rection	Late reverberation is separable from direct content under chosen model.	Speech/music clarity can drop if early reflections are over-suppressed.	Avoid strong dereverb when room character is part of artistic intent.
dynamic- s_and_loud- ness	Program dynamics fit compressor/limiter time constants and thresholds.	Pumping, breathing, or overs if thresholds and release are mis-set.	Avoid applying multiple strong dynamics stages without gain staging checks.
granu- lar_and_- modulation	Grain and modulation rates are musically matched to source texture.	Incoherent grain scheduling can produce choppiness or blur.	Avoid dense granular settings on speech intelligibility-critical content.
pitch_de- tec- tion_and_- tracking	F0 evidence is strong in the selected analysis band and frame size.	Octave errors and voicing flips under heavy noise/polyphony.	Avoid as the sole control signal for dense polyphonic mixtures.
re- tune_and_in tonation	Detected notes map cleanly to intended tonal center/scale.	Over-correction can flatten expressive vibrato or slides.	Avoid aggressive correction when preserving natural micro-intonation is required.
separa- tion_and_de composition	Sources have partially separable spectral or statistical structure.	Component bleeding and musical noise under overlap or model mismatch.	Avoid expecting perfect stems from strongly correlated or co-modulated sources.
spa- tial_and_- multichan- nel	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.

Group	Assumptions	Failure modes	When not to use
<code>spectral_time_frequency_transforms</code>	Transform parameterization matches target time-frequency structure.	Incorrect parameterization can smear events or over-fragment spectra.	Avoid default settings for highly nonstationary signals without tuning.
<code>time_stretch_and_pitch_core</code>	Frames are locally quasi-stationary and harmonic evolution is reasonably smooth.	High-ratio stretch can introduce phasiness and blurred transients.	Avoid for extreme percussive-only material when attack realism is critical.

analysis_qa_and_automation

Algorithm ID	Assumptions	Failure modes	When not to use
<code>analysis_qa_and_automation.auto_parameter_tuning_bayesian_optimization</code>	Feature extraction settings align with domain (speech vs music etc.).	False positives/negatives under domain shift.	Avoid treating single metrics as absolute quality verdicts.
<code>analysis_qa_and_automation.batch_preset_recommendation_based_on_source_features</code>	Feature extraction settings align with domain (speech vs music etc.).	False positives/negatives under domain shift.	Avoid treating single metrics as absolute quality verdicts.
<code>analysis_qa_and_automation.clip_hum_buzz_artifact_detection</code>	Feature extraction settings align with domain (speech vs music etc.).	False positives/negatives under domain shift.	Avoid treating single metrics as absolute quality verdicts.
<code>analysis_qa_and_automation.key_chord_detection</code>	Feature extraction settings align with domain (speech vs music etc.).	False positives/negatives under domain shift.	Avoid treating single metrics as absolute quality verdicts.
<code>analysis_qa_and_automation.onset_beat_downbeat_tracking</code>	Feature extraction settings align with domain (speech vs music etc.).	False positives/negatives under domain shift.	Avoid treating single metrics as absolute quality verdicts.
<code>analysis_qa_and_automation.pesq_stoi_visqol_quality_metrics</code>	Feature extraction settings align with domain (speech vs music etc.).	False positives/negatives under domain shift.	Avoid treating single metrics as absolute quality verdicts.

Algorithm ID	Assumptions	Failure modes	When not to use
<code>analysis_qa_and_automation.silence_speech_music_classifiers</code>	Feature extraction settings align with domain (speech vs music etc.).	False positives/negatives under domain shift.	Avoid treating single metrics as absolute quality verdicts.
<code>analysis_qa_and_automation.structure_segmentation_verse_chorus_sections</code>	Feature extraction settings align with domain (speech vs music etc.).	False positives/negatives under domain shift.	Avoid treating single metrics as absolute quality verdicts.

creative_spectral_effects

Algorithm ID	Assumptions	Failure modes	When not to use
<code>creative_spectral_effects.cross_synthesis_vocoder</code>	Spectral manipulations are desired even with timbral coloration.	Can introduce intentional but strong coloration or temporal artifacts.	Avoid for transparent restoration/mastering paths.
<code>creative_spectral_effects.formant_painting_warping</code>	Spectral manipulations are desired even with timbral coloration.	Can introduce intentional but strong coloration or temporal artifacts.	Avoid for transparent restoration/mastering paths.
<code>creative_spectral_effects.phase_randomization_textures</code>	Spectral manipulations are desired even with timbral coloration.	Can introduce intentional but strong coloration or temporal artifacts.	Avoid for transparent restoration/mastering paths.
<code>creative_spectral_effects.resonator_filterbank_morphing</code>	Spectral manipulations are desired even with timbral coloration.	Can introduce intentional but strong coloration or temporal artifacts.	Avoid for transparent restoration/mastering paths.
<code>creative_spectral_effects.spectral_blur_smear</code>	Spectral manipulations are desired even with timbral coloration.	Can introduce intentional but strong coloration or temporal artifacts.	Avoid for transparent restoration/mastering paths.
<code>creative_spectral_effects.spectral_contrast_exaggeration</code>	Spectral manipulations are desired even with timbral coloration.	Can introduce intentional but strong coloration or temporal artifacts.	Avoid for transparent restoration/mastering paths.

Algorithm ID	Assumptions	Failure modes	When not to use
<code>creative_spectral_effects.spectral_convolution_effects</code>	Spectral manipulations are desired even with timbral coloration.	Can introduce intentional but strong coloration or temporal artifacts.	Avoid for transparent restoration/mastering paths.
<code>creative_spectral_effects.spectral_freeze_banks</code>	Spectral manipulations are desired even with timbral coloration.	Can introduce intentional but strong coloration or temporal artifacts.	Avoid for transparent restoration/mastering paths.

denoise_and_restoration

Algorithm ID	Assumptions	Failure modes	When not to use
<code>de-noise_and_restoration.declick_decrackle_medium_wavelet_interpolation</code>	Noise/artifacts are distinguishable from desired signal statistics. Noise model should be representative of observed noise floor.	Over-reduction can remove detail and create modulation artifacts. Mismatched noise estimate leaves residue or damages detail.	Avoid high reduction settings on sparse acoustic sources without auditioning. Avoid static settings on rapidly varying non-stationary noise.
<code>de-noise_and_restoration.declip_via_sparse_reconstruction</code>	Noise/artifacts are distinguishable from desired signal statistics. Noise model should be representative of observed noise floor.	Over-reduction can remove detail and create modulation artifacts. Mismatched noise estimate leaves residue or damages detail.	Avoid high reduction settings on sparse acoustic sources without auditioning. Avoid static settings on rapidly varying non-stationary noise.
<code>de-noise_and_restoration.difusion_based_speech_audio_denoise</code>	Noise/artifacts are distinguishable from desired signal statistics. Noise model should be representative of observed noise floor.	Over-reduction can remove detail and create modulation artifacts. Mismatched noise estimate leaves residue or damages detail.	Avoid high reduction settings on sparse acoustic sources without auditioning. Avoid static settings on rapidly varying non-stationary noise.
<code>de-noise_and_restoration.log_mmse</code>	Noise/artifacts are distinguishable from desired signal statistics. Noise model should be representative of observed noise floor.	Over-reduction can remove detail and create modulation artifacts. Mismatched noise estimate leaves residue or damages detail.	Avoid high reduction settings on sparse acoustic sources without auditioning. Avoid static settings on rapidly varying non-stationary noise.

Algorithm ID	Assumptions	Failure modes	When not to use
de-noise_and_restitution.minimum_statistics_noise_tracking	Noise/artifacts are distinguishable from desired signal statistics. Noise model should be representative of observed noise floor.	Over-reduction can remove detail and create modulation artifacts. Mismatched noise estimate leaves residue or damages detail.	Avoid high reduction settings on sparse acoustic sources without auditioning. Avoid static settings on rapidly varying non-stationary noise.
de-noise_and_restitution.mmse_stsa	Noise/artifacts are distinguishable from desired signal statistics. Noise model should be representative of observed noise floor.	Over-reduction can remove detail and create modulation artifacts. Mismatched noise estimate leaves residue or damages detail.	Avoid high reduction settings on sparse acoustic sources without auditioning. Avoid static settings on rapidly varying non-stationary noise.
de-noise_and_restitution.rn-noise_style_denoiser	Noise/artifacts are distinguishable from desired signal statistics. Noise model should be representative of observed noise floor.	Over-reduction can remove detail and create modulation artifacts. Mismatched noise estimate leaves residue or damages detail.	Avoid high reduction settings on sparse acoustic sources without auditioning. Avoid static settings on rapidly varying non-stationary noise.
de-noise_and_restitution.wiener_denoising	Noise/artifacts are distinguishable from desired signal statistics. Noise model should be representative of observed noise floor.	Over-reduction can remove detail and create modulation artifacts. Mismatched noise estimate leaves residue or damages detail.	Avoid high reduction settings on sparse acoustic sources without auditioning. Avoid static settings on rapidly varying non-stationary noise.

dereverb_and_room_correction

Algorithm ID	Assumptions	Failure modes	When not to use
dereverb_and_room_correction.blind_deconvolution_dereverb	Late reverberation is separable from direct content under chosen model. Reverberation tail is assumed more diffuse than direct content.	Speech/music clarity can drop if early reflections are over-suppressed. Over-suppression can thin tonal body and ambience.	Avoid strong dereverb when room character is part of artistic intent. Avoid for intentionally wet effects unless mix preservation is planned.

Algorithm ID	Assumptions	Failure modes	When not to use
dereverb_and_-room_correction.-drr_guided_dereverb	<p>Late reverberation is separable from direct content under chosen model.</p> <p>Reverberation tail is assumed more diffuse than direct content.</p>	<p>Speech/music clarity can drop if early reflections are over-suppressed. Over-suppression can thin tonal body and ambience.</p>	<p>Avoid strong dereverb when room character is part of artistic intent. Avoid for intentionally wet effects unless mix preservation is planned.</p>
dereverb_and_-room_correction.late_reverb_suppression_via_coherence	<p>Late reverberation is separable from direct content under chosen model.</p> <p>Reverberation tail is assumed more diffuse than direct content.</p>	<p>Speech/music clarity can drop if early reflections are over-suppressed. Over-suppression can thin tonal body and ambience.</p>	<p>Avoid strong dereverb when room character is part of artistic intent. Avoid for intentionally wet effects unless mix preservation is planned.</p>
dereverb_and_-room_correction.-multi-band_adaptive_deverb	<p>Late reverberation is separable from direct content under chosen model.</p> <p>Reverberation tail is assumed more diffuse than direct content.</p>	<p>Speech/music clarity can drop if early reflections are over-suppressed. Over-suppression can thin tonal body and ambience.</p>	<p>Avoid strong dereverb when room character is part of artistic intent. Avoid for intentionally wet effects unless mix preservation is planned.</p>
dereverb_and_-room_correction.rec-tion.neural_dereverb_module	<p>Late reverberation is separable from direct content under chosen model.</p> <p>Model priors assume training-like signal statistics.</p> <p>Reverberation tail is assumed more diffuse than direct content.</p>	<p>Speech/music clarity can drop if early reflections are over-suppressed. Generalization gaps can produce unstable artifacts. Over-suppression can thin tonal body and ambience.</p>	<p>Avoid strong dereverb when room character is part of artistic intent. Avoid fully unattended use on out-of-domain material.</p> <p>Avoid for intentionally wet effects unless mix preservation is planned.</p>
dereverb_and_-room_correction.-room_impulse_inverse_filtering	<p>Late reverberation is separable from direct content under chosen model.</p> <p>Reverberation tail is assumed more diffuse than direct content.</p>	<p>Speech/music clarity can drop if early reflections are over-suppressed. Over-suppression can thin tonal body and ambience.</p>	<p>Avoid strong dereverb when room character is part of artistic intent. Avoid for intentionally wet effects unless mix preservation is planned.</p>

Algorithm ID	Assumptions	Failure modes	When not to use
dereverb_and_-room_correction.spec-tral_de-cay_sub-traction	Late reverberation is separable from direct content under chosen model. Reverberation tail is assumed more diffuse than direct content.	Speech/music clarity can drop if early reflections are over-suppressed. Over-suppression can thin tonal body and ambience.	Avoid strong dereverb when room character is part of artistic intent. Avoid for intentionally wet effects unless mix preservation is planned.
dereverb_and_-room_correction.w-pe_dere-verberation	Late reverberation is separable from direct content under chosen model. Reverberation tail is assumed more diffuse than direct content.	Speech/music clarity can drop if early reflections are over-suppressed. Over-suppression can thin tonal body and ambience.	Avoid strong dereverb when room character is part of artistic intent. Avoid for intentionally wet effects unless mix preservation is planned.

dynamics_and_loudness

Algorithm ID	Assumptions	Failure modes	When not to use
dynamics_and_loudness.ebu_r128_normalization	Program dynamics fit compressor/limiter time constants and thresholds.	Pumping, breathing, or overs if thresholds and release are mis-set.	Avoid applying multiple strong dynamics stages without gain staging checks.
dynamics_and_loudness.itu_bs_1770_loudness_measurement_gating	Program dynamics fit compressor/limiter time constants and thresholds.	Pumping, breathing, or overs if thresholds and release are mis-set.	Avoid applying multiple strong dynamics stages without gain staging checks.
dynamics_and_loudness.lufs_target_-mastering_chain	Program dynamics fit compressor/limiter time constants and thresholds.	Pumping, breathing, or overs if thresholds and release are mis-set.	Avoid applying multiple strong dynamics stages without gain staging checks.
dynamics_and_loudness.multi_band_compression	Program dynamics fit compressor/limiter time constants and thresholds.	Pumping, breathing, or overs if thresholds and release are mis-set.	Avoid applying multiple strong dynamics stages without gain staging checks.

Algorithm ID	Assumptions	Failure modes	When not to use
<code>dynamics_and_loudness.spectral_dynamics_bin_wise_compressor_expander</code>	Program dynamics fit compressor/limiter time constants and thresholds.	Pumping, breathing, or overs if thresholds and release are mis-set.	Avoid applying multiple strong dynamics stages without gain staging checks.
<code>dynamics_and_loudness.transient_shaping</code>	Program dynamics fit compressor/limiter time constants and thresholds.	Pumping, breathing, or overs if thresholds and release are mis-set.	Avoid applying multiple strong dynamics stages without gain staging checks.
<code>dynamics_and_loudness.true_peak_limiting</code>	Program dynamics fit compressor/limiter time constants and thresholds.	Pumping, breathing, or overs if thresholds and release are mis-set.	Avoid applying multiple strong dynamics stages without gain staging checks.
<code>dynamics_and_loudness.upward_compression</code>	Program dynamics fit compressor/limiter time constants and thresholds.	Pumping, breathing, or overs if thresholds and release are mis-set.	Avoid applying multiple strong dynamics stages without gain staging checks.

granular_and_modulation

Algorithm ID	Assumptions	Failure modes	When not to use
<code>granular_and_modulation.am_fm_ring_modulation_blocks</code>	Grain and modulation rates are musically matched to source texture.	Incoherent grain scheduling can produce choppiness or blur.	Avoid dense granular settings on speech intelligibility-critical content.
<code>granular_and_modulation.envelope_followed_modulation_routing</code>	Grain and modulation rates are musically matched to source texture.	Incoherent grain scheduling can produce choppiness or blur.	Avoid dense granular settings on speech intelligibility-critical content.
<code>granular_and_modulation.formant_lfo_modulation</code>	Grain and modulation rates are musically matched to source texture.	Incoherent grain scheduling can produce choppiness or blur.	Avoid dense granular settings on speech intelligibility-critical content.
<code>granular_and_modulation.freeze_grain_morphing</code>	Grain and modulation rates are musically matched to source texture.	Incoherent grain scheduling can produce choppiness or blur.	Avoid dense granular settings on speech intelligibility-critical content.

Algorithm ID	Assumptions	Failure modes	When not to use
<code>granular_and_modulation.grain_-_cloud_pitch_textures</code>	Grain and modulation rates are musically matched to source texture.	Incoherent grain scheduling can produce choppiness or blur.	Avoid dense granular settings on speech intelligibility-critical content.
<code>granular_and_modulation.granular_-_time_stretch_engine</code>	Grain and modulation rates are musically matched to source texture.	Incoherent grain scheduling can produce choppiness or blur.	Avoid dense granular settings on speech intelligibility-critical content.
<code>granular_and_modulation.rhythmic_-_gate_stutter_quantizer</code>	Grain and modulation rates are musically matched to source texture.	Incoherent grain scheduling can produce choppiness or blur.	Avoid dense granular settings on speech intelligibility-critical content.
<code>granular_and_modulation.spectral_tremolo</code>	Grain and modulation rates are musically matched to source texture.	Incoherent grain scheduling can produce choppiness or blur.	Avoid dense granular settings on speech intelligibility-critical content.

`pitch_detection_and_tracking`

Algorithm ID	Assumptions	Failure modes	When not to use
<code>pitch_detection_and_-track_in_g.crepe_style_neural_f0</code>	F0 evidence is strong in the selected analysis band and frame size. Model priors assume training-like signal statistics.	Octave errors and voicing flips under heavy noise/polyphony. Generalization gaps can produce unstable artifacts.	Avoid as the sole control signal for dense polyphonic mixtures. Avoid fully unattended use on out-of-domain material.
<code>pitch_detection_and_-tracking.harmonic_product_spectrum_hps</code>	F0 evidence is strong in the selected analysis band and frame size.	Octave errors and voicing flips under heavy noise/polyphony.	Avoid as the sole control signal for dense polyphonic mixtures.
<code>pitch_detection_and_-tracking.pyin</code>	F0 evidence is strong in the selected analysis band and frame size.	Octave errors and voicing flips under heavy noise/polyphony.	Avoid as the sole control signal for dense polyphonic mixtures.

Algorithm ID	Assumptions	Failure modes	When not to use
<code>pitch_detection_and_-tracking.rapt</code>	F0 evidence is strong in the selected analysis band and frame size.	Octave errors and voicing flips under heavy noise/polyphony.	Avoid as the sole control signal for dense polyphonic mixtures.
<code>pitch_detection_and_-tracking.subharmonic_summation</code>	F0 evidence is strong in the selected analysis band and frame size.	Octave errors and voicing flips under heavy noise/polyphony.	Avoid as the sole control signal for dense polyphonic mixtures.
<code>pitch_detection_and_-tracking.swipe</code>	F0 evidence is strong in the selected analysis band and frame size.	Octave errors and voicing flips under heavy noise/polyphony.	Avoid as the sole control signal for dense polyphonic mixtures.
<code>pitch_detection_and_-track-ing.viterbi_smoothed_pitch_contour_tracking</code>	F0 evidence is strong in the selected analysis band and frame size.	Octave errors and voicing flips under heavy noise/polyphony.	Avoid as the sole control signal for dense polyphonic mixtures.
<code>pitch_detection_and_-tracking.yin</code>	F0 evidence is strong in the selected analysis band and frame size.	Octave errors and voicing flips under heavy noise/polyphony.	Avoid as the sole control signal for dense polyphonic mixtures.

retune_and_intonation

Algorithm ID	Assumptions	Failure modes	When not to use
<code>re-tune_and_intonation.adaptive_intonation_context-sensitive_intervals</code>	Detected notes map cleanly to intended tonal center/scale. Pitch trajectory estimates should be continuous enough for retuning.	Over-correction can flatten expressive vibrato or slides. Fast F0 jumps can cause audible stepping.	Avoid aggressive correction when preserving natural micro-intonation is required. Avoid high-strength retune on breath/noise segments.

Algorithm ID	Assumptions	Failure modes	When not to use
<code>re-tune_and_intonation.-chord_aware_retuning</code>	Detected notes map cleanly to intended tonal center/scale. Pitch trajectory estimates should be continuous enough for retuning.	Over-correction can flatten expressive vibrato or slides. Fast F0 jumps can cause audible stepping.	Avoid aggressive correction when preserving natural micro-intonation is required. Avoid high-strength retune on breath/noise segments.
<code>re-tune_and_intonation.just_in-tonation_map-ping_per_key_center</code>	Detected notes map cleanly to intended tonal center/scale. Pitch trajectory estimates should be continuous enough for retuning.	Over-correction can flatten expressive vibrato or slides. Fast F0 jumps can cause audible stepping.	Avoid aggressive correction when preserving natural micro-intonation is required. Avoid high-strength retune on breath/noise segments.
<code>re-tune_and_intonation.key_aware_retun-ing_with_con-fidence_weight-ing</code>	Detected notes map cleanly to intended tonal center/scale. Pitch trajectory estimates should be continuous enough for retuning.	Over-correction can flatten expressive vibrato or slides. Fast F0 jumps can cause audible stepping.	Avoid aggressive correction when preserving natural micro-intonation is required. Avoid high-strength retune on breath/noise segments.
<code>re-tune_and_intonation.porta-men-to_aware_re-tune_curves</code>	Detected notes map cleanly to intended tonal center/scale. Pitch trajectory estimates should be continuous enough for retuning.	Over-correction can flatten expressive vibrato or slides. Fast F0 jumps can cause audible stepping.	Avoid aggressive correction when preserving natural micro-intonation is required. Avoid high-strength retune on breath/noise segments.
<code>re-tune_and_intonation.s-cala_mts_s-cale_im-port_and_quan-tization</code>	Detected notes map cleanly to intended tonal center/scale. Pitch trajectory estimates should be continuous enough for retuning.	Over-correction can flatten expressive vibrato or slides. Fast F0 jumps can cause audible stepping.	Avoid aggressive correction when preserving natural micro-intonation is required. Avoid high-strength retune on breath/noise segments.

Algorithm ID	Assumptions	Failure modes	When not to use
re-tune_and_intonation.time_-varying_-cents_maps	Detected notes map cleanly to intended tonal center/scale. Pitch trajectory estimates should be continuous enough for retuning.	Over-correction can flatten expressive vibrato or slides. Fast F0 jumps can cause audible stepping.	Avoid aggressive correction when preserving natural micro-intonation is required. Avoid high-strength retune on breath/noise segments.
re-tune_and_intonation.vibrato_preserving_correction	Detected notes map cleanly to intended tonal center/scale. Pitch trajectory estimates should be continuous enough for retuning.	Over-correction can flatten expressive vibrato or slides. Fast F0 jumps can cause audible stepping.	Avoid aggressive correction when preserving natural micro-intonation is required. Avoid high-strength retune on breath/noise segments.

separation_and_decomposition

Algorithm ID	Assumptions	Failure modes	When not to use
separation_and_decomposition.demucs_style_stem_separation_backend	Sources have partially separable spectral or statistical structure.	Component bleeding and musical noise under overlap or model mismatch.	Avoid expecting perfect stems from strongly correlated or co-modulated sources.
separation_and_decomposition.ica_bss_for_multichannel_stems	Sources have partially separable spectral or statistical structure.	Component bleeding and musical noise under overlap or model mismatch.	Avoid expecting perfect stems from strongly correlated or co-modulated sources.
separation_and_decomposition.nmf_decomposition	Sources have partially separable spectral or statistical structure.	Component bleeding and musical noise under overlap or model mismatch.	Avoid expecting perfect stems from strongly correlated or co-modulated sources.
separation_and_decomposition.probabilistic_latent_component_separation	Sources have partially separable spectral or statistical structure.	Component bleeding and musical noise under overlap or model mismatch.	Avoid expecting perfect stems from strongly correlated or co-modulated sources.
separation_and_decomposition.rpcas_hps	Sources have partially separable spectral or statistical structure.	Component bleeding and musical noise under overlap or model mismatch.	Avoid expecting perfect stems from strongly correlated or co-modulated sources.

Algorithm ID	Assumptions	Failure modes	When not to use
separation_and_decomposition.sinusoidal_residual_transient_decomposition	Sources have partially separable spectral or statistical structure.	Component bleeding and musical noise under overlap or model mismatch.	Avoid expecting perfect stems from strongly correlated or co-modulated sources.
separation_and_decomposition.tenзор_decomposition_cp_tucker	Sources have partially separable spectral or statistical structure.	Component bleeding and musical noise under overlap or model mismatch.	Avoid expecting perfect stems from strongly correlated or co-modulated sources.
separation_and_decomposition.u_net_vocal_accompaniment_split	Sources have partially separable spectral or statistical structure.	Component bleeding and musical noise under overlap or model mismatch.	Avoid expecting perfect stems from strongly correlated or co-modulated sources.

spatial_and_multichannel

Algorithm ID	Assumptions	Failure modes	When not to use
spatial_and_multichannel.binaural_it-dILD_synthesis	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
spatial_and_multichannel.binaural_motion_trajectory_designer	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
spatial_and_multichannel.coherence_based_de_reverb_multichannel	Channel geometry/order and timing metadata are correct. Reverberation tail is assumed more diffuse than direct content.	Spatial collapse, combing, or localization bias from misalignment. Over-suppression can thin tonal body and ambience.	Avoid blind spatial processing when channel order/calibration is unknown. Avoid for intentionally wet effects unless mix preservation is planned.

Algorithm ID	Assumptions	Failure modes	When not to use
<code>spatial_and_multichannel.cross_channel_click_pop_repair</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.distance_based_amplitude_panning</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.decorrelated_reverb_upmix</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.microphone_array_calibration_tones</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.-multichannel_noise_psd_tracking</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.-multichannel_wiener_postfilter</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.

Algorithm ID	Assumptions	Failure modes	When not to use
<code>spatial_and_multichannel.phase_aligned_mid_side_field_rotation</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.phase_consistent_multichannel_denoise</code>	Channel geometry/order and timing metadata are correct. Noise model should be representative of observed noise floor.	Spatial collapse, combing, or localization bias from misalignment. Mismatched noise estimate leaves residue or damages detail.	Avoid blind spatial processing when channel order/calibration is unknown. Avoid static settings on rapidly varying nonstationary noise.
<code>spatial_and_multichannel.pvx_directional_spectral_warp</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.pvx_interaural_coherence_shaping</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.pvx_interchannel_phase_loc king</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.

Algorithm ID	Assumptions	Failure modes	When not to use
<code>spatial_and_multichannel.pvx_multichannel_time_alignment</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combining, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.pvx_spatial_freeze_and_trajectory</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combining, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.pvx_spatial_transient_preservation</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combining, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.rotating_speaker_doppler_field</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combining, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.spectral_freeze_resynthesis</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combining, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.spectral_spatial_granulator</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combining, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.

Algorithm ID	Assumptions	Failure modes	When not to use
<code>spatial_and_multichannel.stereo_width_frequency_dependent_control</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.stochastic_spatial_diffusion_cloud</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.-transaural_crossstalk_cancellation</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.
<code>spatial_and_multichannel.vbap_adaptive_panning</code>	Channel geometry/order and timing metadata are correct.	Spatial collapse, combing, or localization bias from misalignment.	Avoid blind spatial processing when channel order/calibration is unknown.

spectral_time_frequency_transforms

Algorithm ID	Assumptions	Failure modes	When not to use
<code>spectral_time_frequency_transforms.chirplet_transform_m_analysis</code>	Transform parameterization matches target time-frequency structure.	Incorrect parameterization can smear events or over-fragment spectra.	Avoid default settings for highly non-stationary signals without tuning.
<code>spectral_time_frequency_transforms.-constant_q_transform_m_cqt_processing</code>	Transform parameterization matches target time-frequency structure.	Incorrect parameterization can smear events or over-fragment spectra.	Avoid default settings for highly non-stationary signals without tuning.

Algorithm ID	Assumptions	Failure modes	When not to use
<code>spectral_time_frequency_transforms.-.multi_window_stft_fusion</code>	Transform parameterization matches target time-frequency structure.	Incorrect parameterization can smear events or over-fragment spectra.	Avoid default settings for highly non-stationary signals without tuning.
<code>spectral_time_frequency_transforms.ngt_based_processing</code>	Transform parameterization matches target time-frequency structure.	Incorrect parameterization can smear events or over-fragment spectra.	Avoid default settings for highly non-stationary signals without tuning.
<code>spectral_time_frequency_transforms.reassigned_spectrogram_methods</code>	Transform parameterization matches target time-frequency structure.	Incorrect parameterization can smear events or over-fragment spectra.	Avoid default settings for highly non-stationary signals without tuning.
<code>spectral_time_frequency_transforms.-.synchrosqueezed_stft</code>	Transform parameterization matches target time-frequency structure.	Incorrect parameterization can smear events or over-fragment spectra.	Avoid default settings for highly non-stationary signals without tuning.
<code>spectral_time_frequency_transforms.-.variable_q_transform.m_vqt</code>	Transform parameterization matches target time-frequency structure.	Incorrect parameterization can smear events or over-fragment spectra.	Avoid default settings for highly non-stationary signals without tuning.
<code>spectral_time_frequency_transforms.wavelet_packet_processing</code>	Transform parameterization matches target time-frequency structure.	Incorrect parameterization can smear events or over-fragment spectra.	Avoid default settings for highly non-stationary signals without tuning.

time_scale_and_pitch_core

Algorithm ID	Assumptions	Failure modes	When not to use
<code>time_scale_and_pitch_core.beat_syncronous_time_warping</code>	Frames are locally quasi-stationary and harmonic evolution is reasonably smooth.	High-ratio stretch can introduce phasiness and blurred transients.	Avoid for extreme percussive-only material when attack realism is critical.

Algorithm ID	Assumptions	Failure modes	When not to use
<code>time_s-scale_and_pitch_core.harmonic_percussive_split_tsm</code>	Frames are locally quasi-stationary and harmonic evolution is reasonably smooth.	High-ratio stretch can introduce phasiness and blurred transients.	Avoid for extreme percussive-only material when attack realism is critical.
<code>time_s-scale_and_pitch_core.lp_psolala</code>	Frames are locally quasi-stationary and harmonic evolution is reasonably smooth.	High-ratio stretch can introduce phasiness and blurred transients.	Avoid for extreme percussive-only material when attack realism is critical.
<code>time_s-scale_and_pitch_core.multi_resolution_phase_vocoder</code>	Frames are locally quasi-stationary and harmonic evolution is reasonably smooth. Phase continuity assumptions hold best for moderate stretch ratios.	High-ratio stretch can introduce phasiness and blurred transients. Extreme settings increase phasiness/transient blur risk.	Avoid for extreme percussive-only material when attack realism is critical. Avoid very large stretch+pitch shifts without transient controls.
<code>time_s-scale_and_pitch_core.non-linear_time_maps</code>	Frames are locally quasi-stationary and harmonic evolution is reasonably smooth.	High-ratio stretch can introduce phasiness and blurred transients.	Avoid for extreme percussive-only material when attack realism is critical.
<code>time_s-scale_and_pitch_core.td_psola</code>	Frames are locally quasi-stationary and harmonic evolution is reasonably smooth.	High-ratio stretch can introduce phasiness and blurred transients.	Avoid for extreme percussive-only material when attack realism is critical.
<code>time_s-scale_and_pitch_core.wsolala_waveform_similarity_overlap_add</code>	Frames are locally quasi-stationary and harmonic evolution is reasonably smooth.	High-ratio stretch can introduce phasiness and blurred transients.	Avoid for extreme percussive-only material when attack realism is critical.

pvx Benchmark Report

Source: benchmarks.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Reproducible STFT/ISTFT benchmark summary.

```
python3 scripts_generate_docs_extras.py --run-benchmarks
```

Spec: sample_rate=48000 Hz, duration=4.0 s

Host: macOS-15.1.1-arm64-arm-64bit-Mach-O | machine=arm64 | python=3.14.3

Backend	Status	Elapsed (ms)	Peak host memory (MB)	SNR vs input (dB)	Spectra I dist vs input (dB)	SNR vs CPU (dB)	Spectral dist vs CPU (dB)	Notes
cpu	ok	20.3 56	12.02	160.5 928	0.0	n/a	n/a	
cuda	unavailable	n/a	n/a	n/a	n/a	n/a	n/a	CUDA mode requires CuPy. Install a matching `cupy-cudaXX` package.
apple_silicon_na-na-tive_cpu	ok	20.2 76	12.02	160.5 928	0.0	160.5 928	0.0	

pvx Pipeline Cookbook

Source: [cookbook.html](#)

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Pipeline cookbook for practical one-liners, including Unix pipes and mastering chains.

Markdown source: [docs/PIPELINE_COOKBOOK.md](#).

Phase-vocoder core

Workflow	Command	Why
Moderate vocal stretch with formant preservation	<pre>python3 pvxvoc.py vocal.wav --time-stretch 1.15 --pitch-mode formant-preserving --output-dir out --suffix _pv</pre>	Retains speech-like vowel envelope while stretching timing.
Independent cents retune	<pre>python3 pvxvoc.py lead.wav --pitch-shift-cents -23 --time-stretch 1.0 --output-dir out --suffix _cents</pre>	Applies precise microtonal offset without tempo change.
Extreme stretch with multi-stage strategy	<pre>python3 pvxvoc.py ambience.wav --target-duration 600 --ambient-preset --n-fft 16384 --win-length 16384 --hop-size 2048 --window kaiser --kaiser-beta 18 --output-dir out --suffix _ambient600x</pre>	PaulStretch-style ambient profile for very large ratios using stochastic phase and onset time-credit controls.
Ultra-smooth speech stretch (600x)	<pre>python3 pvxvoc.py speech.wav --target-duration 600 --stretch-mode standard --phase-engine propagate --phase-locking identity --n-fft 8192 --win-length 8192 --hop-size 256 --window hann --normalize peak --peak-dbfs -1 --compressor-threshold-db -30 --compressor-ratio 2.0 --compressor-attack-ms 25 --compressor-release-ms 250 --compressor-makeup-db 4 --limiter-threshold 0.95 --output-dir out --suffix _speech600x</pre>	Prefers continuity and intelligibility over texture animation; avoids choppy stochastic artifacts on speech sources.
Auto-profile plan preview	<pre>python3 pvxvoc.py input.wav --auto-profile --auto-transform --explain-plan</pre>	Prints the resolved profile/config plan before long renders.
Multi-resolution fusion stretch	<pre>python3 pvxvoc.py input.wav --multires-fusion --multires-ffts 1024,2048,4096 --multires-weights 0.2,0.35,0.45 --time-stretch 1.4 --output-dir out --suffix _multires</pre>	Blends several FFT scales to reduce single-resolution bias on complex program material.

Workflow	Command	Why
Checkpointed long render with manifest	<pre>python3 pvxvoc.py long.wav --time-stretch 12 --auto-segment-seconds 0.5 --checkpoint-dir checkpoints --manifest-json reports/run_manifest.json --output-dir out --suffix _long</pre>	Caches segment renders for resume workflows and writes run metadata for reproducibility.

Transform selection

Workflow	Command	Why
Default production backend (FFT + transient protection)	<pre>python3 pvxvoc.py mix.wav --transform fft --time-stretch 1.07 --transient-preserve --phase-locking identity --output-dir out --suffix _fft</pre>	Use when you need the fastest and most stable general-purpose phase-vocoder path.
Reference Fourier baseline using explicit DFT mode	<pre>python3 pvxvoc.py tone_sweep.wav --transform dft --time-stretch 1.00 --pitch-shift-semitones 0 --output-dir out --suffix _dft_ref</pre>	Useful for parity checks and controlled transform-comparison experiments.
Prime-size frame experiment with CZT backend	<pre>python3 pvxvoc.py archival_take.wav --transform czt --n-fft 1531 --win-length 1531 --hop-size 382 --time-stretch 1.03 --output-dir out --suffix _czt</pre>	Alternative numerical path for awkward/prime frame sizes when validating edge cases.
DCT timbral compaction for smooth harmonic material	<pre>python3 pvxvoc.py strings.wav --transform dct --pitch-shift-cents -18 --soft-clip-level 0.95 --output-dir out --suffix _dct</pre>	Real-basis coefficients can emphasize envelope-like structure for creative reshaping.
DST odd-symmetry color pass	<pre>python3 pvxvoc.py snare_loop.wav --transform dst --time-stretch 0.92 --phase-locking off --output-dir out --suffix _dst</pre>	Provides an alternate real-basis artifact profile useful for creative percussive processing.
Hartley real-basis exploratory render	<pre>python3 pvxvoc.py synth_pad.wav --transform hartley --time-stretch 1.30 --pitch-shift-semitones 3 --output-dir out --suffix _hartley</pre>	Compares Hartley-domain behavior against complex FFT phase-vocoder output.

Workflow	Command	Why
A/B sweep of transform back-ends from shell loop	<pre>for t in fft dft czt dct dst hartley; do python3 ppxvoc.py voice.wav --transform "\$t" --time-stretch 1.1 --output-dir out --suffix "_\$t"; done</pre>	Fast listening workflow for selecting the least-artifact transform on your source.

Microtonal

Workflow	Command	Why
Custom cents map retune	<pre>python3 ppxretune.py vox.wav --root 60 --scale-cents 0,90,204,294,408,498,612,702,816,906,1020,110 --strength 0.8 --output-dir out</pre>	Maps incoming notes to a custom 12-degree microtonal scale.
Conform CSV with per-segment ratios	<pre>python3 ppxconform.py solo.wav map_conform.csv --pitch-mode ratio --output-dir out --suffix _conform</pre>	Applies timeline-specific time and pitch trajectories from CSV.

Pipelines

Workflow	Command	Why
Time-stretch -> denoise -> dereverb in one pipe	<pre>python3 ppxvoc.py input.wav --time-stretch 1.25 --stdout python3 ppxdenoise.py --reduction-db 10 --stdout python3 ppxdereverb.py --strength 0.45 --output-dir out --suffix _clean</pre>	Single-pass CLI chain for serial DSP in Unix pipes.
Morph -> formant -> unison	<pre>python3 ppxmorph.py a.wav b.wav -o - python3 ppxformant.py - --mode preserve --stdout python3 ppxunison.py - --voices 5 --detune-cents 8 --output-dir out --suffix _morph_stack</pre>	Builds a richer timbre chain with no intermediate files.
Pitch-follow sidechain map (A controls B)	<pre>python3 HPS-pitch-track.py A.wav python3 ppxvoc.py B.wav --pitch-follow-stdin --pitch-conf-min 0.75 --pitch-lowconf-mode hold --time-stretch-factor 1.0 --output output.wav</pre>	Tracks F0 contour from source A and applies it as a dynamic pitch-ratio control map on source B.

Mastering

Workflow	Command	Why
Integrated loudness targeting with limiter	<code>python3 pvxvoc.py mix.wav --time-stretch 1.0 --target-lufs -14 --compressor-threshold-db -20 --compressor-ratio 3 --limiter-threshold 0.98 --output-dir out --suffix _master</code>	Combines dynamics and loudness controls in shared mastering chain.
Soft clip and hard safety ceiling	<code>python3 pvxharmonize.py bus.wav --intervals 0,7,12 --mix 0.35 --soft-clip-level 0.92 --soft-clip-type tanh --hard-clip-level 0.99 --output-dir out</code>	Adds saturation while enforcing a strict final peak ceiling.

Batch

Workflow	Command	Why
Batch stretch over folder	<code>python3 pvxvoc.py stems/*.wav --time-stretch 1.08 --output-dir out/stems --overwrite</code>	Applies consistent transform to many files with one command.
Dry-run output validation	<code>python3 pvxdenoise.py takes/*.wav --reduction-db 8 --dry-run --output-dir out/preview</code>	Checks filename resolution and collisions without writing audio.

Automation

Workflow	Command	Why
A/B report generation	<code>python3 scripts_ab_COMPARE.py --input mix.wav --a-args "--time-stretch 1.1 --transform fft" --b-args "--time-stretch 1.1 --transform dct" --out-dir reports/ab --name fft_vs_dct</code>	Creates JSON/Markdown objective reports for fast algorithm and parameter comparisons.
Benchmark matrix sweep	<code>python3 scripts_benchmark_matrix.py --input mix.wav --transforms fft,dft,dct --windows hann,kaiser --n-ffts 1024,2048 --devices cpu --out-dir reports/bench</code>	Produces reproducible CSV/JSON runtime matrices across transform/window/FFT combinations.

Workflow	Command	Why
Quality regression check	<pre>python3 scripts_quality_regression.py --input mix.wav --output out/reg.wav --render-args "--time-stretch 1.2 --transform fft" --baseline-json reports/baseline.json --report-json reports/regression.json</pre>	Compares current renders against baseline objective metrics with configurable tolerances.

Spatial

Workflow	Command	Why
VBAP adaptive panning via algorithm dispatcher	<pre>python3 -m ppx.algorithms.spatial_and_multichannel.imaging_and_panning.vbap_adaptive_panning input.wav --output-channels 6 --azimuth-deg 35 --width 0.8 --output out/vbap.wav</pre>	Demonstrates algorithm-level spatial module invocation.

Analysis/QA

Workflow	Command	Why
Quality metrics on processed speech	<pre>python3 -m ppx.algorithms.analysis_qa_and_automation.pesq_stoi_visqol_quality_metrics clean.wav noisy.wav --output out/qa.json</pre>	Collects objective quality indicators for regression tracking.

pvx Command-Line Interface (CLI) Flag Index

Source: [cli_flags.html](#)

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Total unique long flags: **181**

Source index: [docs/CLI_FLAGS_REFERENCE.md](#) and [docs/cli_flags_reference.json](#).

hps_pitch_track.py

Flag	Required	Default	Choices	Action	Description	Source
--backend	False	auto	auto, pyin, acf		Pitch backend (default: auto -> pyin if available, else acf)	src/pvx/cli/hps_pitch_track.py
--confidence-floor	False	0.0			Set confidence below this floor to 0.0 (default: 0.0).	src/pvx/cli/hps_pitch_track.py
--fmax	False	1200.0			Maximum F0 in Hz (default: 1200)	src/pvx/cli/hps_pitch_track.py
--fmin	False	50.0			Minimum F0 in Hz (default: 50)	src/pvx/cli/hps_pitch_track.py
--frame-length	False	2048			Frame length in samples (default: 2048)	src/pvx/cli/hps_pitch_track.py
--hop-size	False	256			Hop size in samples (default: 256)	src/pvx/cli/hps_pitch_track.py
--output	False				Output CSV path (default: '-' for stdout)	src/pvx/cli/hps_pitch_track.py
--ratio-max	False	4.0			Upper clamp for emitted pitch_ratio (default: 4.0).	src/pvx/cli/hps_pitch_track.py
--ratio-min	False	0.25			Lower clamp for emitted pitch_ratio (default: 0.25).	src/pvx/cli/hps_pitch_track.py
--ratio-reference	False	median	median, mean, first, hz		Reference for emitted pitch_ratio values (default: median voiced f0).	src/pvx/cli/hps_pitch_track.py
--reference-hz	False				Reference frequency in Hz when --ratio-reference hz.	src/pvx/cli/hps_pitch_track.py

Flag	Required	Default	Choices	Action	Description	Source
--smooth-frames	False	5			Smoothing window for pitch_ratio frames (default: 5).	src/pvx/cli/hps_pitch_track.py
--stretch	False	1.0			Emit constant stretch column value (default: 1.0).	src/pvx/cli/hps_pitch_track.py

pvx.py

Flag	Required	Default	Choices	Action	Description	Source
--chunk-seconds	False	0.25			Chunk/segment duration for `--auto-segment-seconds` (default: 0.25)	src/pvx/cli/pvx.py
--context-ms	False				Optional stateful context window in milliseconds (default: auto from window/hop)	src/pvx/cli/pvx.py
--cross-fade-ms	False	0.0			Crossfade used for segment assembly in milliseconds (default: 0.0)	src/pvx/cli/pvx.py
--example	False			store_true	Print a copy-paste chain example and exit	src/pvx/cli/pvx.py
--keep-intermediate	False			store_true	Keep intermediate stage files after successful completion	src/pvx/cli/pvx.py
--mode	False	stateful	stateful, wrapper		Stream engine: stateful chunk processor (default) or wrapper compatibility mode	src/pvx/cli/pvx.py
--out	True				Final output path (or '-')	src/pvx/cli/pvx.py

Flag	Required	Default	Choices	Action	Description	Source
--output	True				Final output path (or '-')	src/pvx/cli/pvx.py
--pipeline	True				Pipeline string with stages separated by ' '. Example: "voc --stretch 1.2 formant --mode preserve"	src/pvx/cli/pvx.py
--work-dir	False				Optional directory for intermediate stage files	src/pvx/cli/pvx.py

pvxconform.py

Flag	Required	Default	Choices	Action	Description	Source
--cross-fade-ms	False	8.0			Segment cross-fade in milliseconds	src/pvx/cli/pvxconform.py
--map	True				CSV map path	src/pvx/cli/pvxconform.py
--resample-mode	False	auto	auto, fft, linear			src/pvx/cli/pvxconform.py

pvxdenoise.py

Flag	Required	Default	Choices	Action	Description	Source
--floor	False	0.1			Noise floor multiplier	src/pvx/cli/pvxdenoise.py
--noise-file	False				Optional external noise reference	src/pvx/cli/pvxdenoise.py

Flag	Required	Default	Choices	Action	Description	Source
--noise-seconds	False	0.35			Noise profile duration from start	src/pvx/cli/pvx-denoise.py
--reduction-db	False	12.0			Reduction strength in dB	src/pvx/cli/pvx-denoise.py
--smooth	False	5			Temporal smoothing frames	src/pvx/cli/pvx-denoise.py

pvxdeverb.py

Flag	Required	Default	Choices	Action	Description	Source
--decay	False	0.92			Tail memory decay 0..1	src/pvx/cli/pvxdeverb.py
--floor	False	0.12			Per-bin floor multiplier	src/pvx/cli/pvxdeverb.py
--strength	False	0.45			Tail suppression strength 0..1	src/pvx/cli/pvxdeverb.py

pvxformant.py

Flag	Required	Default	Choices	Action	Description	Source
--formant-lifter	False	32				src/pvx/cli/pvxformant.py
--formant-max-gain-db	False	12.0				src/pvx/cli/pvxformant.py
--formant-shift-ratio	False	1.0			Formant ratio (>1 up, <1 down)	src/pvx/cli/pvxformant.py

Flag	Required	Default	Choices	Action	Description	Source
--mode	False	shift	shift, pre-s erve			src/pvx/cli/pvxformant.py
--pitch-shift-cents	False	0.0			Additional microtonal pitch shift in cents before formant stage	src/pvx/cli/pvxformant.py
--pitch-shift-semitones	False	0.0			Optional pitch shift before formant stage	src/pvx/cli/pvxformant.py
--resample-mode	False	auto	auto, fft, linear			src/pvx/cli/pvxformant.py

pvxfreeze.py

Flag	Required	Default	Choices	Action	Description	Source
--duration	False	3.0			Output freeze duration in seconds	src/pvx/cli/pvxfreeze.py
--freeze-time	False	0.2			Freeze anchor time in seconds	src/pvx/cli/pvxfreeze.py
--random-phase	False			store_true	Add subtle phase randomization per frame	src/pvx/cli/pvxfreeze.py

pvxharmonize.py

Flag	Required	Default	Choices	Action	Description	Source
--force-stereo	False			store_true	Mix result as stereo with panning	src/pvx/cli/pvxharmonize.py

Flag	Required	Default	Choices	Action	Description	Source
--gains	False				Optional comma-separated linear gain per voice	src/pvx/cli/pvxharmonize.py
--intervals	False	0, 4, 7			Comma-separated semitone intervals per voice (supports fractional values)	src/pvx/cli/pvxharmonize.py
--intervals-cents	False				Optional cents offsets per voice, added to --intervals (e.g. 0,14,-12)	src/pvx/cli/pvxharmonize.py
--pans	False				Optional comma-separated pan per voice [-1..1]	src/pvx/cli/pvxharmonize.py
--resample-mode	False	auto	auto, fft, linear			src/pvx/cli/pvxharmonize.py

pvxlayer.py

Flag	Required	Default	Choices	Action	Description	Source
--harmonic-gain	False	1. 0				src/pvx/cli/pvxlayer.py
--harmonic-kernel	False	31				src/pvx/cli/pvxlayer.py
--harmonic-pitch-cents	False	0. 0				src/pvx/cli/pvxlayer.py
--harmonic-pitch-semitones	False	0. 0				src/pvx/cli/pvxlayer.py
--harmonic-stretch	False	1. 0				src/pvx/cli/pvxlayer.py
--percussive-gain	False	1. 0				src/pvx/cli/pvxlayer.py

Flag	Required	Default	Choices	Action	Description	Source
--percussive-kernel	False	31				src/pvx/cli/pvxlayer.py
--percussive-pitch-cents	False	0. 0				src/pvx/cli/pvxlayer.py
--percussive-pitch-semitones	False	0. 0				src/pvx/cli/pvxlayer.py
--percussive-stretch	False	1. 0				src/pvx/cli/pvxlayer.py
--resample-mode	False	aut o	auto, fft, linear			src/pvx/cli/pvxlayer.py

pvxmorph.py

Flag	Required	Default	Choices	Action	Description	Source
--alpha	False	0. 5			Morph amount 0..1 (0=A, 1=B)	src/pvx/cli/pvx-morph.py
--output	False				Output file path	src/pvx/cli/pvx-morph.py
--out-format	False				Output extension/format; for --stdout defaults to wav	src/pvx/cli/pvx-morph.py
--over-write	False			store _true		src/pvx/cli/pvx-morph.py
--stdout	False			store _true	Write processed audio to stdout stream (for piping); equivalent to -o -	src/pvx/cli/pvx-morph.py

pvxretune.py

Flag	Required	Default	Choices	Action	Description	Source
--chunk-ms	False	80.0			Analysis/process chunk duration in ms	src/pvx/cli/pvxre-tune.py
--f0-max	False	120.0				src/pvx/cli/pvxre-tune.py
--f0-min	False	60.0				src/pvx/cli/pvxre-tune.py
--overlap-ms	False	20.0			Chunk overlap in ms	src/pvx/cli/pvxre-tune.py
--resample-mode	False	auto	auto, fft, linear			src/pvx/cli/pvxre-tune.py
--root	False	C			Scale root note (C,C#,D,...,B)	src/pvx/cli/pvxre-tune.py
--scale	False	chromatic			Named scale for 12-TET quantization	src/pvx/cli/pvxre-tune.py
--scale-cents	False				Optional comma-separated microtonal scale degrees in cents within one octave, relative to --root (example: 0,90,204,294,408,498,612,702,816,906,1020,1110)	src/pvx/cli/pvxre-tune.py

Flag	Required	Default	Choices	Action	Description	Source
--strength	False	0.85			Correction strength 0..1	src/pvx/cli/pvxre-tune.py

pvxtransient.py

Flag	Required	Default	Choices	Action	Description	Source
--pitch-shift-cents	False				Optional microtonal pitch shift in cents (added to --pitch-shift-semitones)	src/pvx/cli/pvxtransient.py
--pitch-shift-ratio	False				Pitch ratio override. Accepts decimals (1.5), integer ratios (3/2), and expressions ($2^{(1/12)}$).	src/pvx/cli/pvxtransient.py
--pitch-shift-semitones	False	0.0				src/pvx/cli/pvxtransient.py
--resample-mode	False	auto	auto, fft, linear			src/pvx/cli/pvxtransient.py
--target-duration	False				Target duration in seconds	src/pvx/cli/pvxtransient.py
--time-stretch	False	1.0				src/pvx/cli/pvxtransient.py
--transient-threshold	False	1.6				src/pvx/cli/pvxtransient.py

pvxunison.py

Flag	Required	Default	Choices	Action	Description	Source
--detune-cents	False	14.0			Total detune span in cents	src/pvx/cli/pvxunison.py
--dry-mix	False	0.2			Dry signal mix amount	src/pvx/cli/pvxunison.py
--resample-mode	False	auto	auto, fft, linear			src/pvx/cli/pvxunison.py
--voices	False	5			Number of unison voices	src/pvx/cli/pvxunison.py
--width	False	1.0			Stereo width multiplier 0..2	src/pvx/cli/pvxunison.py

pvxvoc.py

Flag	Required	Default	Choices	Action	Description	Source
--ambient-phase-mix	False	0.5			Random-phase blend when --phase-engine hybrid (0.0=propagated only, 1.0=random only; default: 0.5).	src/pvx/core/voc.py
--ambient-preset	False			store_true	Convenience preset for ambient extreme stretch (random phase engine, onset-time-credit, transient preserve, conservative staging).	src/pvx/core/voc.py
--analysis-channel	False	mix	first, mix		Channel strategy for F0 estimation with --target-f0 (default: mix)	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--auto-profile	False			store_true	Analyze input and choose a profile automatically (speech/music/percussion/ambient/extreme).	src/pvx/core/voc.py
--auto-profile-lookup-head-seconds	False	6.0			Seconds of audio used when estimating --auto-profile (default: 6.0).	src/pvx/core/voc.py
--auto-segment-seconds	False	0.0			Optional segment size in seconds for long jobs. When >0, processing runs per segment with crossfade assembly.	src/pvx/core/voc.py
--auto-transform	False			store_true	Allow automatic transform selection when --transform is not explicitly set.	src/pvx/core/voc.py
--bit-depth	False	inherit			Output bit-depth policy (default: inherit). Ignored when --subtype is set.	src/pvx/core/voc.py
--cents	False				Pitch shift in cents (+1200 is one octave up)	src/pvx/core/voc.py
--checkpoint-dir	False				Directory used to cache per-segment checkpoint chunks for resume workflows.	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--checkpoint-id	False				Optional checkpoint run identifier (default: hash of input/settings).	src/pvx/core/voc.py
--clip	False			store_true	Legacy alias: hard clip at +/-1.0 when set	src/pvx/core/voc.py
--coherence-strength	False	0.0			Coherence lock strength in [0,1] (0=off, 1=full lock).	src/pvx/core/voc.py
--compander-attack-ms	False	8.0			Compander attack time in ms	src/pvx/core/voc.py
--compander-compress-ratio	False	3.0			Compander compression ratio (>=1)	src/pvx/core/voc.py
--compander-expand-ratio	False	1.8			Compander expansion ratio (>=1)	src/pvx/core/voc.py
--compander-makeup-db	False	0.0			Compander makeup gain in dB	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--com-pander-release-ms	False	120.0			Compander release time in ms	src/pvx/core/voc.py
--com-pander-threshold-db	False				Enable compander threshold in dBFS	src/pvx/core/voc.py
--compressor-attack-ms	False	10.0			Compressor attack time in ms	src/pvx/core/voc.py
--compressor-makeup-db	False	0.0			Compressor makeup gain in dB	src/pvx/core/voc.py
--compressor-ratio	False	4.0			Compressor ratio (≥ 1)	src/pvx/core/voc.py
--compressor-release-ms	False	120.0			Compressor release time in ms	src/pvx/core/voc.py
--compressor-threshold-db	False				Enable compressor above threshold dBFS	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--cpu	False			store_true	Alias for --device cpu.	src/pvx/core/voc.py
--cuda-device	False	0			CUDA device index used when --device is auto/cuda (default: 0)	src/pvx/core/voc.py
--device	False	auto	auto, cpu, cuda		Compute device: auto (prefer CUDA), cpu, or cuda	src/pvx/core/voc.py
--dither	False	none			Dither policy before quantized writes (default: none)	src/pvx/core/voc.py
--dither-seed	False				Deterministic RNG seed for dithering (default: random seed)	src/pvx/core/voc.py
--dry-run	False			store_true	Resolve settings without writing files	src/pvx/core/voc.py
--example	False				Print copy-paste example command(s) and exit.	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--expander-attack-ms	False	5.0			Expander attack time in ms	src/pvx/core/voc.py
--expander-ratio	False	2.0			Expander ratio (≥ 1)	src/pvx/core/voc.py
--expander-release-ms	False	120.0			Expander release time in ms	src/pvx/core/voc.py
--expander-threshold-db	False				Enable downward expander below threshold dBFS	src/pvx/core/voc.py
--explain-plan	False			store_true	Print resolved processing plan JSON and exit without rendering audio.	src/pvx/core/voc.py
--extreme-stretch-threshold	False	2.0			Auto-mode threshold for multistage activation (default: 2.0).	src/pvx/core/voc.py
--extreme-time-stretch	False			store_true	Force multistage strategy even when ratio is moderate.	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--f0-max	False	1000.0			Maximum F0 search bound in Hz (default: 1000)	src/pvx/core/voc.py
--f0-min	False	50.0			Minimum F0 search bound in Hz (default: 50)	src/pvx/core/voc.py
--formant-lifter	False	32			Cepstral lifter cutoff for formant envelope extraction (default: 32)	src/pvx/core/voc.py
--formant-max-gain-db	False	12.0			Max per-bin formant correction gain in dB (default: 12)	src/pvx/core/voc.py
--formant-strength	False	1.0			Formant correction blend 0..1 when pitch mode is formant-preserving (default: 1.0)	src/pvx/core/voc.py
--fourier-sync	False			store_true	Enable fundamental frame locking. Uses generic short-time Fourier transforms with per-frame FFT sizes locked to detected F0.	src/pvx/core/voc.py
--fourier-sync-max-fft	False	8192			Maximum frame FFT size for --fourier-sync (default: 8192)	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--fourier-sync-min-fft	False	256			Minimum frame FFT size for --fourier-sync (default: 256)	src/pvx/core/voc.py
--fourier-sync-smooth	False	5			Smoothing span (frames) for pre-canned F0 track in --fourier-sync (default: 5)	src/pvx/core/voc.py
--gpu	False			store_true	Alias for --device cuda.	src/pvx/core/voc.py
--guided	False			store_true	Interactive guided mode for first-time users.	src/pvx/core/voc.py
--hard-clip-level	False				Hard clip level in linear full-scale	src/pvx/core/voc.py
--hop-size	False	512			Hop size in samples (default: 512)	src/pvx/core/voc.py
--kaiser-beta	False	14.0			Kaiser window beta parameter used when --window kaiser (default: 14.0)	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--limiter-threshold	False				Peak limiter threshold in linear full-scale	src/pvx/core/voc.py
--manifest-append	False			store_true	Append entries to an existing --manifest-json file instead of replacing it.	src/pvx/core/voc.py
--manifest-json	False				Write processing manifest JSON with per-file settings and outcomes.	src/pvx/core/voc.py
--max-stage-stretch	False	1.8			Maximum per-stage ratio used in multistage mode (default: 1.8).	src/pvx/core/voc.py
--metadata-policy	False	none			Output metadata policy: none, sidecar, or copy (sidecar implementation)	src/pvx/core/voc.py
--multires-ffts	False	1024,2048,4096			Comma-separated FFT sizes for --multires-fusion (default: 1024,2048,4096)	src/pvx/core/voc.py
--multires-fusion	False			store_true	Blend multiple FFT resolutions for each channel before pitch resampling.	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--multires-weights	False				Comma-separated fusion weights for --multires-fusion (defaults to equal weights).	src/pvx/core/voc.py
--n-fft	False	2048			FFT size (default: 2048)	src/pvx/core/voc.py
--no-center	False			store_true	Disable center padding in STFT/ISTFT	src/pvx/core/voc.py
--no-onset-realign	False			store_true	Disable fractional read-position re-alignment on onsets when --onset-time-credit is enabled.	src/pvx/core/voc.py
--no-progress	False			store_true		src/pvx/core/voc.py
--normalize	False	none	none, peak, rms		Output normalization mode	src/pvx/core/voc.py
--onset-credit-max	False	8.0			Maximum accumulated onset time credit in analysis-frame units (default: 8.0).	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--on-set-credit-pull	False	0.5			Fraction of per-frame read advance removable while onset credit exists (0.0..1.0, default: 0.5).	src/pvx/core/voc.py
--on-set-time-credit	False			store_true	Enable onset-triggered time-credit scheduling to reduce transient smear during extreme stretching.	src/pvx/core/voc.py
--out	False				Explicit output file path (single-input mode only). Alias: --out	src/pvx/core/voc.py
--output	False				Explicit output file path (single-input mode only). Alias: --out	src/pvx/core/voc.py
--out-put-dir	False				Directory for output files (default: same directory as each input)	src/pvx/core/voc.py
--out-put-format	False				Output format/extension (e.g. wav, flac, aiff). Default: keep input extension.	src/pvx/core/voc.py
--over-write	False			store_true	Overwrite existing outputs	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--peak-dBFS	False				Target peak dBFS when --normalize-peak	src/pvx/core/voc.py
--phase-engine	False	propagate			Phase synthesis engine: propagate (classic phase vocoder), hybrid (propagated + stochastic blend), random (ambient stochastic phase).	src/pvx/core/voc.py
--phase-locking	False	identity	off, identity		Inter-bin phase locking mode for transient fidelity (default: identity)	src/pvx/core/voc.py
--phase-random-seed	False				Optional deterministic seed for random/hybrid phase generation.	src/pvx/core/voc.py
--pitch	False				Pitch shift in semitones (+12 is one octave up)	src/pvx/core/voc.py
--pitch-conf-min	False	0.0			Minimum accepted map confidence (default: 0 disables gating).	src/pvx/core/voc.py
--pitch-follow-stdin	False			store_true	Shortcut for --pitch-map-stdin (sidechain pitch-follow workflows).	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
<code>--pitch-low-conf-mode</code>	False	<code>hold</code>	<code>hold, unity, interp</code>		Low-confidence map handling mode (default: hold).	<code>src/pvx/core/voc.py</code>
<code>--pitch-map</code>	False				CSV control map for time-varying stretch/pitch. Columns: start_sec,end_sec plus optional stretch,pitch_ratio/pitch_cents/pitch_semitones,confidence. Use '-' to read from stdin.	<code>src/pvx/core/voc.py</code>
<code>--pitch-map-cross-fade-ms</code>	False	<code>8.0</code>			Crossfade between processed map segments in milliseconds (default: 8.0).	<code>src/pvx/core/voc.py</code>
<code>--pitch-map-smooth-ms</code>	False	<code>0.0</code>			Moving-average smoothing over map pitch ratios in milliseconds.	<code>src/pvx/core/voc.py</code>
<code>--pitch-map-stdin</code>	False			<code>store_true</code>	Read control-map CSV from stdin.	<code>src/pvx/core/voc.py</code>
<code>--pitch-mode</code>	False	<code>standard</code>	<code>standard, formant-preserving</code>		Pitch mode: standard shift or formant-preserving correction (default: standard)	<code>src/pvx/core/voc.py</code>
<code>--pitch-shift-cents</code>	False				Pitch shift in cents (+1200 is one octave up)	<code>src/pvx/core/voc.py</code>

Flag	Required	Default	Choices	Action	Description	Source
--pitch-shift-ratio	False				Pitch ratio (>1 up, <1 down). Accepts decimals (1.5), integer ratios (3/2), and expressions ($2^{(1/12)}$).	src/pvx/core/voc.py
--pitch-shift-semi-tones	False				Pitch shift in semitones (+12 is one octave up)	src/pvx/core/voc.py
--preset	False	none			High-level intent preset. Legacy: none/vocal/ambient/extreme. New: default/vocal_studio/drums_safe/extreme_ambient/stereo_coherent.	src/pvx/core/voc.py
--quality-profile	False	neutral			Named tuning profile for vocoder defaults (default: neutral)	src/pvx/core/voc.py
--quiet	False			store_true	Reduce output and hide status bars	src/pvx/core/voc.py
--ratio	False				Pitch ratio (>1 up, <1 down). Accepts decimals (1.5), integer ratios (3/2), and expressions ($2^{(1/12)}$).	src/pvx/core/voc.py
--ref-channel	False	0			Reference channel index used by --stereo-mode ref_channel_lock (default: 0).	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--resample-mode	False	auto	auto, fft, linear		Resampling engine (auto=fft if scipy available, else linear)	src/pvx/core/voc.py
--resume	False			store_true	Reuse existing checkpoint chunks from --checkpoint-dir when available.	src/pvx/core/voc.py
--rms-dBFS	False				Target RMS dBFS when --normalize rms	src/pvx/core/voc.py
--semi-tones	False				Pitch shift in semitones (+12 is one octave up)	src/pvx/core/voc.py
--silent	False			store_true	Suppress all console output	src/pvx/core/voc.py
--soft-clip-drive	False	1.0			Soft clip drive amount (>0)	src/pvx/core/voc.py
--soft-clip-level	False				Soft clip output ceiling in linear full-scale	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--soft-clip-type	False	tanh	tanh, arctan, cubic		Soft clip transfer type	src/pvx/core/voc.py
--stdout	False			store_true	Write processed audio to stdout stream (for piping); requires exactly one input	src/pvx/core/voc.py
--stereo-mode	False	independent	independent, mid_side_lock, ref_channel_lock		Channel coherence strategy: independent (legacy), mid_side_lock (M/S-coupled), ref_channel_lock (phase-lock to reference channel).	src/pvx/core/voc.py
--stretch	False				Alias for --time-stretch.	src/pvx/core/voc.py
--stretch-mode	False	auto	auto, standard, multistage		Stretch strategy: standard (single pass), multistage (chained moderate passes), or auto (multistage only for extreme ratios; default: auto).	src/pvx/core/voc.py
--subtype	False				Explicit libsndfile output subtype override (e.g., PCM_16, PCM_24, FLOAT)	src/pvx/core/voc.py
--suffix	False	_pv			Suffix appended to output filename stem (default: _pv)	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--target-duration	False	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Absolute target duration in seconds (overrides --time-stretch)	src/pvx/core/voc.py
--target-f0	False	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Target fundamental frequency in Hz. Auto-estimates source F0 per file.	src/pvx/core/voc.py
--target-lufs	False	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Integrated loudness target in LUFS	src/pvx/core/voc.py
--target-pitch-shift-semitones	False	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Pitch shift in semitones (+12 is one octave up)	src/pvx/core/voc.py
--target-sample-rate	False	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Output sample rate in Hz (default: keep input rate)	src/pvx/core/voc.py
--time-stretch	False	<input type="checkbox"/> 1.0	<input type="checkbox"/>	<input type="checkbox"/>	Final duration multiplier (1.0=unchanged, 2.0=2x longer)	src/pvx/core/voc.py
--time-stretch-factor	False	<input type="checkbox"/> 1.0	<input type="checkbox"/>	<input type="checkbox"/>	Final duration multiplier (1.0=unchanged, 2.0=2x longer)	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--transform	False	fft			Per-frame transform backend for STFT/ISTFT paths (default: fft; options: fft, dft, czt, dct, dst, hartley)	src/pvx/core/voc.py
--transient-cross-fade-ms	False	10.0			Crossfade duration for transient/steady stitching (default: 10 ms).	src/pvx/core/voc.py
--transient-mode	False	off	off, reset, hybrid, wsola		Transient handling mode: off (none), reset (phase reset), hybrid (PV steady + WSOLA transients), or wsola (time-domain transient-safe path).	src/pvx/core/voc.py
--transient-preserve	False			store_true	Enable transient phase resets based on spectral flux	src/pvx/core/voc.py
--transient-protect-ms	False	30.0			Transient protection width in milliseconds (default: 30).	src/pvx/core/voc.py
--transient-sensitivity	False	0.5			Transient detector sensitivity in [0,1] (higher catches more onsets).	src/pvx/core/voc.py
--transient-threshold	False	2.0			Spectral-flux multiplier for transient detection (default: 2.0)	src/pvx/core/voc.py

Flag	Required	Default	Choices	Action	Description	Source
--true-peak-max-dbtp	False				Apply output gain trim to enforce max true-peak in dBTP	src/pvx/core/voc.py
--verbose	False	0		count	Increase verbosity (repeat for extra detail)	src/pvx/core/voc.py
--verbosity	False	normal			Console verbosity level	src/pvx/core/voc.py
--win-length	False	2048			Window length in samples (default: 2048)	src/pvx/core/voc.py
--window	False	hann			Window type (default: hann)	src/pvx/core/voc.py

pvxwarp.py

Flag	Required	Default	Choices	Action	Description	Source
--cross-fade-ms	False	8.0				src/pvx/cli/pvxwarp.py
--map	True				CSV map with start_sec,end_sec,stretch	src/pvx/cli/pvxwarp.py

Flag	Required	Default	Choices	Action	Description	Source
--resample-mode	False	auto	auto, fft, linear			src/pvx/cli/ pvxwarp.py

pvx Citation Quality

Source: citations.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Citation quality report and BibTeX export.

[docs/references.bib](#)

Link-Type Summary

Link type	Count
arxiv	7

Link type	Count
doi	33
publisher_or_standard	8
scholar	89
web	6

Scholar-Link Entries (upgrade candidates)

Year	Authors	Title	URL
1976	M. R. Portnoff	Implementation of the Digital Phase Vocoder Using the Fast Fourier Transform	link
1980	M. R. Portnoff	Time-Scale Modification of Speech Based on Short-Time Fourier Analysis	link
1986	R. J. McAulay; T. F. Quatieri	Speech Analysis/Synthesis Based on a Sinusoidal Representation	link
2002	C. Duxbury; M. Davies; M. Sandler	Improved Time-Scaling of Musical Audio Using Phase Locking at Transients	link
2000	J. Bonada	Automatic Technique in Frequency Domain for Near-Lossless Time-Scale Modification of Audio	link
1985	N. Roucos; A. Wilgus	High Quality Time-Scale Modification for Speech	link
1993	W. Verhelst; M. Roelands	An Overlap-Add Technique Based on Waveform Similarity (WSOLA) for High Quality Time-Scale Modification of Speech	link
2002	S. Arfib; D. Keiler; U. Zölzer	DAFX: Digital Audio Effects (chapter references on pitch/time processing)	link
1967	A. M. Noll	Cepstrum Pitch Determination	link
1976	L. R. Rabiner; M. J. Cheng; A. E. Rosenberg; C. A. McGonegal	A Comparative Performance Study of Several Pitch Detection Algorithms	link

Year	Authors	Title	URL
1995	D. Talkin	A Robust Algorithm for Pitch Tracking (RAPT)	link
1993	P. Boersma	Accurate Short-Term Analysis of the Fundamental Frequency and the Harmonics-to-Noise Ratio of a Sampled Sound	link
2010	C. Schörkhuber; A. Klapuri	Constant-Q Transform Toolbox for Music Processing	link
2002	P. Flandrin; F. Auger; E. Chassande-Mottin	Time-Frequency Reassignment: From Principles to Algorithms	link
1991	S. Mann; S. Haykin	The Chirplet Transform: Physical Considerations	link
2007	T. Virtanen	Monaural Sound Source Separation by Nonnegative Matrix Factorization with Temporal Continuity and Sparseness Criteria	link
1949	N. Wiener	Extrapolation, Interpolation, and Smoothing of Stationary Time Series	link
1996	P. Scalart; J. V. Filho	Speech Enhancement Based on a Priori Signal to Noise Estimation	link
2012	Y. Yoshioka; T. Nakatani	Generalization of Multi-Channel Linear Prediction Methods for Blind MIMO Impulse Response Shortening	link
1997	S. Pulkki	Virtual Sound Source Positioning Using Vector Base Amplitude Panning	link
2019	M. H. C. de Gesmundo et al.	ViSQOL v3: An Open Source Production Ready Objective Speech and Audio Metric	link
1989	R. Bristow-Johnson; M. Bogdanowicz	Phase Vocoder Done Right	link
1998	S. Disch	A New Phase Vocoder Technique for Time-Scale Modification of Audio Signals	link
1995	M. Puckette	Phase-Locked Vocoder	link
2012	B. Zavalishin	The Art of VA Filter Design (sections on phase and spectral processing)	link

Year	Authors	Title	URL
2009	N. Moreau	Toolbox for Time-Scale Modification and Pitch-Shifting of Audio Signals	link
2013	M. Dolson	History of the Phase Vocoder	link
2010	D. S. Hamon; A. Lazarides	Improved WSOLA for Real-Time Time-Scale Modification	link
2015	M. Riess; A. R. Chhetri	Transient Preservation in Time-Scale Modification	link
2008	J. Bonada	Wide-Band Harmonic Sinusoidal Modeling for Time-Scale and Pitch-Scale Modification	link
2004	A. de Cheveigne	Pitch and Time Manipulation of Speech	link
2011	M. Le Roux; E. Vincent	Consistent Wiener Filtering for Audio Source Separation and Time-Frequency Processing	link
2003	J. Laroche	About this Phasiness Business	link
2014	J. Driedger; T. Pratzlich; M. Muller	Let It Bee - Towards NMF-Inspired Audio Mosaicing	link
2010	A. Roebel	A Shape-Invariant Phase Vocoder for Speech Transformation	link
2001	H. Kawahara; A. de Cheveigne	STRAIGHT, Exploitation of the Other Aspects of Vocal Source Information	link
2006	A. Klapuri	Multiple Fundamental Frequency Estimation by Harmonicity and Spectral Smoothness	link
2005	M. McLeod; G. Wyvill	A Smarter Way to Find Pitch	link
1999	S. Ahmadi; H. Spanias	Cepstrum-Based Pitch Detection Using FFT	link
2018	S. Bock; M. Korzeniowski	piano_transcription - F0 and onset tracking	link
2015	B. W. Schuller et al.	Paralinguistics and robust pitch tracking methods	link

Year	Authors	Title	URL
2012	K. Dressler	Sinusoidal Extraction using Frequency-Domain Matching Pursuit	link
2000	S. Kum; C. L. Nikias	Robust F0 Estimation in Noise	link
2001	K. Grochenig	Foundations of Time-Frequency Analysis	link
1998	P. Flandrin	Time-Frequency/Time-Scale Analysis	link
1995	D. L. Donoho	De-Noising by Soft-Thresholding	link
1993	S. Mallat; Z. Zhang	Matching Pursuits with Time-Frequency Dictionaries	link
1989	L. Cohen	Time-Frequency Distributions - A Review	link
2010	A. V. Oppenheim; R. W. Schafer	Discrete-Time Signal Processing (STFT and filter-bank chapters)	link
2006	A. C. Gilbert; M. J. Strauss	Approximation of signals in sparse Fourier dictionaries	link
2019	M. Doring; M. M. Kokuer	Practical NSGT audio applications	link
2012	S. Essid; G. Richard	Musical signal analysis with reassignment methods	link
2007	P. Smaragdis	Convulsive Speech Bases and Their Application to Supervised Speech Separation	link
2009	C. Févotte; N. Bertin; J.-L. Durrieu	Nonnegative Matrix Factorization with the Itakura-Saito Divergence	link
2000	A. Hyvarinen; E. Oja	Independent Component Analysis: Algorithms and Applications	link
2011	E. J. Candès; X. Li; Y. Ma; J. Wright	Robust Principal Component Analysis?	link
2011	N. Ono	Stable and fast update rules for independent low-rank matrix analysis based on auxiliary function technique	link

Year	Authors	Title	URL
1995	A. J. Bell; T. J. Sejnowski	An Information-Maximization Approach to Blind Separation and Blind Deconvolution	link
2019	J. Le Roux; S. Wisdom; H. Erdogan; J. R. Hershey	SDR half-baked or well done?	link
2019	A. Defossez et al.	Music Source Separation in the Waveform Domain	link
2018	N. Takahashi et al.	MMDenseLSTM for source separation	link
2019	F.-R. Stoter; S. Uhlich; A. Liutkus; Y. Mitsufuji	Open-Unmix	link
2001	I. Cohen; B. Berdugo	Speech Enhancement for Non-Stationary Noise Environments	link
1979	S. Boll	Suppression of Acoustic Noise in Speech Using Spectral Subtraction	link
2012	T. Yoshioka; T. Nakatani	Generalization of Multi-Channel Linear Prediction Methods for Blind MIMO Impulse Response Shortening	link
2012	N. D. Gaubitch; P. A. Naylor	Speech Dereverberation	link
2016	K. Kinoshita et al.	A summary of the REVERB challenge	link
2018	N. W. D. Evans et al.	The voicehome and CHiME challenges	link
2018	J.-M. Valin	A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement	link
2019	A. Pandey; D. Wang	A New Framework for CNN-Based Speech Enhancement in the Time Domain	link
2001	A. Rix et al.	Perceptual Evaluation of Speech Quality (PESQ)	link
2011	C. Taal et al.	An Algorithm for Intelligibility Prediction of Time-Frequency Weighted Noisy Speech	link
2012	A. Hines et al.	VISQOL: An Objective Speech Quality Model	link

Year	Authors	Title	URL
2011	K. K. Paliwal	Importance of phase in speech enhancement	link
2013	P. C. Loizou	Speech Enhancement: Theory and Practice	link
2007	T. Lund	Loudness and True-Peak in Digital Audio	link
2020	S. Stables et al.	A Study of Loudness Normalization in Streaming Services	link
2018	M. Fenton	Practical Dynamics Processing for Modern Music Production	link
2011	J. Reiss	Under the Hood of a Dynamic Range Compressor	link
2012	D. Giannoulis; M. Massberg; J. Reiss	Digital Dynamic Range Compressor Design	link
2015	M. Ballou	Handbook for Sound Engineers (Loudness and Dynamics chapters)	link
2017	A. Gibiansky et al.	Deep Voice 2: Multi-Speaker Neural Text-to-Speech	link
2018	N. Kalchbrenner et al.	Efficient Neural Audio Synthesis	link
2019	R. Prenger et al.	WaveGlow: A Flow-based Generative Network for Speech Synthesis	link
2019	K. Kumar et al.	MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis	link
2020	J. Kong et al.	HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis	link
2017	Y. Wang et al.	Tacotron: Towards End-to-End Speech Synthesis	link
2021	S. Wisdom et al.	Differentiable Consistency Constraints for Improved Deep Speech Enhancement	link
2018	C. Donahue; J. McAuley; M. Puckette	Adversarial Audio Synthesis	link

pvx Group: Analysis, QA, and Automation

Source: groups/analysis_qa_and_automation.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Theme: Analysis, QA, and Automation

Folder: `analysis_qa_and_automation` | **Algorithms:** 8

Subgroups: `core (8)`

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
<code>analysis_qa_and_automation.onset_beat_downbeat_tracking</code>	Onset/beat/downbeat tracking	core	<code>src/pvx/algorithms/analysis_qa_and_automation/onset_beat_downbeat_tracking.py</code>	None (generic/default path)	None
<code>analysis_qa_and_automation.key_chord_detection</code>	Key/chord detection	core	<code>src/pvx/algorithms/analysis_qa_and_automation/key_chord_detection.py</code>	None (generic/default path)	None
<code>analysis_qa_and_automation.structure_segmentation_verse_chorus_sections</code>	Structure segmentation (verse/chorus/sections)	core	<code>src/pvx/algorithms/analysis_qa_and_automation/structure_segmentation_verse_chorus_sections.py</code>	None (generic/default path)	Structure segmentation
<code>analysis_qa_and_automation.silence_speech_music_classifiers</code>	Silence/speech/music classifiers	core	<code>src/pvx/algorithms/analysis_qa_and_automation/silence_speech_music_classifiers.py</code>	None (generic/default path)	None
<code>analysis_qa_and_automation.-clip_hum_buzz_artifact_detection</code>	Clip/hum/buzz artifact detection	core	<code>src/pvx/algorithms/analysis_qa_and_automation/clip_hum_buzz_artifact_detection.py</code>	None (generic/default path)	None
<code>analysis_qa_and_automation.pesq_stoi_visqol_quality_metrics</code>	PESQ/STOI/VISQOL quality metrics	core	<code>src/pvx/algorithms/analysis_qa_and_automation/pesq_stoi_visqol_quality_metrics.py</code>	None (generic/default path)	PESQ, STOI, ViSQOL

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
<code>analysis_qa_and_automation.auto_parameter_tuning_bayesian_optimization</code>	Auto-parameter tuning (Bayesian optimization)	core	<code>src/pvx/algorithms/analysis_qa_and_automation/auto_parameter_tuning_bayesian_optimization.py</code>	<code>target_centered</code>	Bayesian optimization
<code>analysis_qa_and_automation.batch_preset_recommendation_based_on_source_features</code>	Batch preset recommendation based on source features	core	<code>src/pvx/algorithms/analysis_qa_and_automation/batch_preset_recommendation_based_on_source_features.py</code>	None (generic/default path)	None

pvx Group: Creative Spectral Effects

Source: groups/creative_spectral_effects.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Theme: Creative Spectral Effects

Folder: creative_spectral_effects | **Algorithms:** 8

Subgroups: core (8)

Algorithm ID	Name	Sub group	Module Path	Parameter keys	Concept links
<code>creative_spectral_effect-s.cross_synthesis_vocoder</code>	Cross-synthesis vocoder	core	<code>src/pvx/algorithms/creative_spectral_effects/cross_synthesis_vocoder.py</code>	None (generic/default path)	Cross synthesis
<code>creative_spectral_effect-s.spectral_convolution_effects</code>	Spectral convolution effects	core	<code>src/pvx/algorithms/creative_spectral_effects/spectral_convolution_effects.py</code>	kernel_size	Convolution
<code>creative_spectral_effect-s.spec-tral_freeze_banks</code>	Spectral freeze banks	core	<code>src/pvx/algorithms/creative_spectral_effects/spec-tral_freeze_banks.py</code>	frame_ratio	None
<code>creative_spectral_effect-s.spec-tral.blur_smear</code>	Spectral blur/smear	core	<code>src/pvx/algorithms/creative_spectral_effects/spectral.blur_smear.py</code>	None (generic/default path)	None
<code>creative_spectral_effect-s.phase_randomization_textures</code>	Phase randomization textures	core	<code>src/pvx/algorithms/creative_spectral_effects/phase_randomization_textures.py</code>	strength	None
<code>creative_spectral_effects.formant_painting_warping</code>	Formant painting/warping	core	<code>src/pvx/algorithms/creative_spectral_effects/formant_painting_warping.py</code>	ratio	Formant
<code>creative_spectral_effects.resonator_filterbank_morphing</code>	Resonator/filterbank morphing	core	<code>src/pvx/algorithms/creative_spectral_effects/resonator_filterbank_morphing.py</code>	None (generic/default path)	Resonator
<code>creative_spectral_effect-s.spectral_contrast_exaggeration</code>	Spectral contrast exaggeration	core	<code>src/pvx/algorithms/creative_spectral_effects/spectral_contrast_exaggeration.py</code>	amount	None

pvx Group: Denoise and Restoration

Source: groups/denoise_and_restoration.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Theme: Denoise and Restoration

Folder: `denoise_and_restoration` | **Algorithms:** 8

Subgroups: `core (8)`

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
de-noise_and_restoration.wiener_denoising	Wiener denoising	core	src/pvx/algorithms/de-noise_and_restoration/wiener_denoising.py	None (generic/default path)	None
de-noise_and_restoration.mmse_stsa	MMSE-STSA	core	src/pvx/algorithms/de-noise_and_restoration/mmse_stsa.py	None (generic/default path)	MMS E-STS A
de-noise_and_restoration.log_mmse	Log-MMSE	core	src/pvx/algorithms/de-noise_and_restoration/log_mmse.py	None (generic/default path)	Log-MMSE
de-noise_and_restoration.minimum_statistics_noise_tracking	Minimum-statistics noise tracking	core	src/pvx/algorithms/de-noise_and_restoration/minimum_statistics_noise_tracking.py	None (generic/default path)	Minimum statistics
de-noise_and_restoration.rnnoise_style_denoiser	RNNoise-style denoiser	core	src/pvx/algorithms/de-noise_and_restoration/rnnoise_style_denoiser.py	None (generic/default path)	RNN noise
de-noise_and_restoration.diffusion_based_speech_audio_denoise	Diffusion-based speech/audio denoise	core	src/pvx/algorithms/de-noise_and_restoration/diffusion_based_speech_audio_denoise.py	None (generic/default path)	None
de-noise_and_restoration.declip_via_sparse_reconstruction	Declip via sparse reconstruction	core	src/pvx/algorithms/de-noise_and_restoration/declip_via_sparse_reconstruction.py	clip_threshold	Declip

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
de-noise_and_restoration.declick_decrackle_median_wavelet_interpolation	Declick/decrackle (median/wavelet + interpolation)	core	src/pvx/algorithms/de-noise_and_restoration/declick_decrackle_median_wavelet_interpolation.py	spike_threshold	Declik

pvx Group: Dereverb and Room Correction

Source: groups/dereverb_and_room_correction.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Theme: Dereverb and Room Correction

Folder: [dereverb_and_room_correction](#) | **Algorithms:** 8

Subgroups: [core \(8\)](#)

Algorithm ID	Name	Sub group	Module Path	Parameter keys	Concept links
dereverb_and_-room_correction.wpe_dereverberation	WPE dereverberation	core	src/pvx/algorithms/dereverb_and_room_correction/wpe_dereverberation.py	taps, delay	Dereverberation
dereverb_and_-room_correction.spectral_decay_subtraction	Spectral decay subtraction	core	src/pvx/algorithms/dereverb_and_room_correction/spectral_decay_subtraction.py	None (generic/default path)	None
dereverb_and_-room_correction.late_reverb_suppression_via_coherence	Late reverb suppression via coherence	core	src/pvx/algorithms/dereverb_and_room_correction/late_reverb_suppression_via_coherence.py	None (generic/default path)	None
dereverb_and_-room_correction.-room_impulse_inverse_filtering	Room impulse inverse filtering	core	src/pvx/algorithms/dereverb_and_room_correction/room_impulse_inverse_filtering.py	None (generic/default path)	None
dereverb_and_-room_correction.-multi_band_adaptive_deverb	Multi-band adaptive deverb	core	src/pvx/algorithms/dereverb_and_room_correction/multi_band_adaptive_deverb.py	None (generic/default path)	None
dereverb_and_-room_correction.-drr_guided_dereverb	DRR-guided dereverb	core	src/pvx/algorithms/dereverb_and_room_correction/drr_guided_dereverb.py	None (generic/default path)	None
dereverb_and_-room_correction.blind_deconvolution_dereverb	Blind deconvolution dereverb	core	src/pvx/algorithms/dereverb_and_room_correction/blind_deconvolution_dereverb.py	None (generic/default path)	Blind deconvolution
dereverb_and_-room_correction.neural_dereverb_module	Neural dereverb module	core	src/pvx/algorithms/dereverb_and_room_correction/neural_dereverb_module.py	None (generic/default path)	None

pvx Group: Dynamics and Loudness

Source: groups/dynamics_and_loudness.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Theme: Dynamics and Loudness

Folder: `dynamics_and_loudness` | **Algorithms:** 8

Subgroups: `core (8)`

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
dynamic_s_and_loudness.ebu_r128_normalization	EBU R128 normalization	core	src/pvx/algorithms/dynamics_and_loudness/ebu_r128_normalization.py	target_lufs	EBU R128
dynamic_s_and_loudness.itu_bs_1770_loudness_measurement_gating	ITU BS.1770 loudness measurement/gating	core	src/pvx/algorithms/dynamics_and_loudness/itu_bs_1770_loudness_measurement_gating.py	gate_lufs	None
dynamic_s_and_loudness.-multi_band_compression	Multi-band compression	core	src/pvx/algorithms/dynamics_and_loudness/multi_band_compression.py	None (generic/default path)	None
dynamic_s_and_loudness.upward_compression	Upward compression	core	src/pvx/algorithms/dynamics_and_loudness/upward_compression.py	threshold_db, ratio	None
dynamic_s_and_loudness.-transient_shaping	Transient shaping	core	src/pvx/algorithms/dynamics_and_loudness/transient_shaping.py	attack_boost, sustain	None
dynamic_s_and_loudness.spectral_dynamics_bin_wise_compressor_expander	Spectral dynamics (bin-wise compressor/expander)	core	src/pvx/algorithms/dynamics_and_loudness/spectral_dynamics_bin_wise_compressor_expander.py	threshold_db, ratio	Compressor, Expander
dynamic_s_and_loudness.true_peak_limiting	True-peak limiting	core	src/pvx/algorithms/dynamics_and_loudness/true_peak_limiting.py	threshold	True peak

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
dynamic-s_and_loudness.lufts_target_mastering_chain	LUFS-target mastering chain	core	src/pvx/algorithms/dynamics_and_loudness/lufts_target_mastering_chain.py	target_lufs	LUFS

pvx Group: Granular and Modulation

Source: groups/granular_and_modulation.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Theme: Granular and Modulation

Folder: [granular_and_modulation](#) | **Algorithms:** 8

Subgroups: [core \(8\)](#)

Algorithm ID	Name	Sub group	Module Path	Parameter keys	Concept links
<code>granular_and_modulation.granular_time_stretch_engine</code>	Granular time-stretch engine	core	<code>src/pvx/algorithms/granular_and_modulation/granular_time_stretch_engine.py</code>	<code>stretch</code> , <code>grain</code> , <code>hop</code>	None
<code>granular_and_modulation.-grain-cloud_pitch_textures</code>	Grain-cloud pitch textures	core	<code>src/pvx/algorithms/granular_and_modulation/grain_cloud_pitch_textures.py</code>	<code>seed</code> , <code>grain</code> , <code>count</code> , <code>stretch</code>	None
<code>granular_and_modulation.freeze_grain_morphing</code>	Freeze-grain morphing	core	<code>src/pvx/algorithms/granular_and_modulation/freeze_grain_morphing.py</code>	<code>grain</code> , <code>start</code>	None
<code>granular_and_modulation.am_fm_ring_modulation_blocks</code>	AM/FM/ring modulation blocks	core	<code>src/pvx/algorithms/granular_and_modulation/am_fm_ring_modulation_blocks.py</code>	<code>freq_hz</code> , <code>fm_depth</code>	Ring modulation
<code>granular_and_modulation.spectral_tremolo</code>	Spectral tremolo	core	<code>src/pvx/algorithms/granular_and_modulation/spectral_tremolo.py</code>	<code>lfo_hz</code>	None
<code>granular_and_modulation.formant_lfo_modulation</code>	Formant LFO modulation	core	<code>src/pvx/algorithms/granular_and_modulation/formant_lfo_modulation.py</code>	<code>lfo_hz</code>	Formant, LFO
<code>granular_and_modulation.rhythmic_gate_stutter_quantizer</code>	Rhythmic gate/stutter quantizer	core	<code>src/pvx/algorithms/granular_and_modulation/rhythmic_gate_stutter_quantizer.py</code>	<code>rate_hz</code> , <code>duty</code>	None
<code>granular_and_modulation.envelope_followed_modulation_routing</code>	Envelope-followed modulation routing	core	<code>src/pvx/algorithms/granular_and_modulation/envelope_followed_modulation_routing.py</code>	<code>depth</code>	None

pvx Group: Pitch Detection and Tracking

Source: groups/pitch_detection_and_tracking.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Theme: Pitch Detection and Tracking

Folder: `pitch_detection_and_tracking` | **Algorithms:** 8

Subgroups: `core (8)`

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
<code>pitch_detection_and_tracking.yin</code>	YIN	core	<code>src/pvx/algorithms/pitch_detection_and_tracking/yin.py</code>	None (generic/default path)	YIN
<code>pitch_detection_and_tracking.pyin</code>	pYIN	core	<code>src/pvx/algorithms/pitch_detection_and_tracking/pyin.py</code>	None (generic/default path)	pYIN
<code>pitch_detection_and_tracking.rapt</code>	RAPT	core	<code>src/pvx/algorithms/pitch_detection_and_tracking/rapt.py</code>	None (generic/default path)	RAPT
<code>pitch_detection_and_tracking.swipe</code>	SWIPE	core	<code>src/pvx/algorithms/pitch_detection_and_tracking/swipe.py</code>	None (generic/default path)	SWIPE
<code>pitch_detection_and_tracking.harmonic_product_spectrum_hps</code>	Harmonic Product Spectrum (HPS)	core	<code>src/pvx/algorithms/pitch_detection_and_tracking/harmonic_product_spectrum_hps.py</code>	None (generic/default path)	Harmonic product spectrum
<code>pitch_detection_and_tracking.subharmonic_summation</code>	Subharmonic summation	core	<code>src/pvx/algorithms/pitch_detection_and_tracking/subharmonic_summation.py</code>	None (generic/default path)	Subharmonic summation
<code>pitch_detection_and_tracking.crepe_style_neural_f0</code>	CREPE-style neural F0	core	<code>src/pvx/algorithms/pitch_detection_and_tracking/crepe_style_neural_f0.py</code>	None (generic/default path)	None

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
<code>pitch_detection_and_tracking.viterbi_smoothed_pitch_contour_tracking</code>	Viterbi-smoothed pitch contour tracking	core	<code>src/pvx/algorithms/pitch_detection_and_tracking/viterbi_smoothed_pitch_contour_tracking.py</code>	<code>root_midi</code> , <code>scale_cents</code> , <code>scale</code> , <code>fmin</code> , <code>fmax</code>	None

pvx Group: Retune and Intonation

Source: groups/retune_and_intonation.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Theme: Retune and Intonation

Folder: `retune_and_intonation` | **Algorithms:** 8

Subgroups: `core (8)`

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
<code>retune_and_intonation.-chord_aware_re-tuning</code>	Chord-aware retuning	core	<code>src/pvx/algorithms/re-tune_and_intonation/chord_aware_retuning.py</code>	strength	None
<code>retune_and_intonation.key_aware_re-tuning_with_confidence_weight-ing</code>	Key-aware retuning with confidence weighting	core	<code>src/pvx/algorithms/re-tune_and_intonation/key_aware_retuning_with-confidence_weight-ing.py</code>	None (generic/default path)	None
<code>retune_and_intonation.just_intonation_map-ping_per_key_center</code>	Just intonation mapping per key center	core	<code>src/pvx/algorithms/re-tune_and_intonation/just_intonation_mapping_per_key_center.py</code>	None (generic/default path)	Just intonation
<code>retune_and_intonation.adap-tive_intonation_-context_sensi-tive_intervals</code>	Adaptive intonation (context-sensitive intervals)	core	<code>src/pvx/algorithms/re-tune_and_intonation/adaptive_intonation_context_sensitive_intervals.py</code>	None (generic/default path)	None
<code>retune_and_intonation.scala_mt-s_scale_im-port_and_quantifi-zation</code>	Scala/MTS scale import and quantization	core	<code>src/pvx/algorithms/re-tune_and_intonation/scala_mts_scale_im-port_and_quantification.py</code>	None (generic/default path)	None
<code>retune_and_intonation.time_vary-ing_cents_maps</code>	Time-varying cents maps	core	<code>src/pvx/algorithms/re-tune_and_intonation/time_varying_cents_maps.py</code>	cents_curve	Cents
<code>retune_and_intonation.vibrato_p-reserving_correc-tion</code>	Vibrato-preserving correction	core	<code>src/pvx/algorithms/re-tune_and_intonation/vibrato_preserving_correction.py</code>	strength	None

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
<code>retune_and_intonation.portamento_to_aware_re-tune_curves</code>	Portamento-aware re-tune curves	core	<code>src/pvx/algorithms/re-tune_and_intonation/portamento_aware_re-tune_curves.py</code>	<code>max_semitone_step, compression</code>	Portamento

pvx Group: Separation and Decomposition

Source: groups/separation_and_decomposition.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Theme: Separation and Decomposition

Folder: `separation_and_decomposition` | **Algorithms:** 8

Subgroups: `core (8)`

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
<code>separation_and_decomposition.rpc_hpss</code>	RPCA HPSS	core	<code>src/pvx/algorithms/separation_and_decomposition/rpc_hpss.py</code>	None (generic/default path)	HPSS , RPCA
<code>separation_and_decomposition.nmf_decomposition</code>	NMF decomposition	core	<code>src/pvx/algorithms/separation_and_decomposition/nmf_decomposition.py</code>	components	NMF
<code>separation_and_decomposition.ica_bss_for_multichannel_stems</code>	ICA/BSS for multi-channel stems	core	<code>src/pvx/algorithms/separation_and_decomposition/ica_bss_for_multichannel_stems.py</code>	None (generic/default path)	ICA
<code>separation_and_decomposition.sinusoidal_residual_transient_decomposition</code>	Sinusoidal +residual+ transient decomposition	core	<code>src/pvx/algorithms/separation_and_decomposition/sinusoidal_residual_transient_decomposition.py</code>	None (generic/default path)	None
<code>separation_and_decomposition.demucs_style_stem_separation_backend</code>	Demucs-style stem separation backend	core	<code>src/pvx/algorithms/separation_and_decomposition/demucs_style_stem_separation_backend.py</code>	None (generic/default path)	Demucs
<code>separation_and_decomposition.u_net_vocal_accompaniment_split</code>	U-Net vocal/accompaniment split	core	<code>src/pvx/algorithms/separation_and_decomposition/u_net_vocal_accompaniment_split.py</code>	None (generic/default path)	U-Net
<code>separation_and_decomposition.tensor_decomposition_cp_tucker</code>	Tensor decomposition (CP/Tucker)	core	<code>src/pvx/algorithms/separation_and_decomposition/tensor_decomposition_cp_tucker.py</code>	rank	Tensor decomposition

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
separation_and_decomposition.probabilistic_latent_component_separation	Probabilistic latent component separation	core	src/pvx/algorithms/separation_and_decomposition/probabilistic_latent_component_separation.py	None (generic/default path)	None

pvx Group: Spatial and Multichannel

Source: groups/spatial_and_multichannel.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Theme: Spatial and Multichannel

Folder: [spatial_and_multichannel](#) | **Algorithms:** 24

Subgroups: [imaging_and_panning \(6\)](#) [phase_vocoder_spatial \(6\)](#)
[multichannel_restoration \(6\)](#) [creative_spatial_fx \(6\)](#)

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
spatial_and_multichannel.vbap_adaptive_panning	VBAP adaptive panning	imag-ing_and_panning	src/pvx/algorithms/spatial_and_multichannel/imag-ing_and_panning/vbap_adaptive_panning.py	output_channels , azimuth_deg , width	VBAP
spatial_and_multichannel.dbap_distance_based_amplitude_panning	DBAP (distance-based amplitude panning)	imag-ing_and_panning	src/pvx/algorithms/spatial_and_multichannel/imag-ing_and_panning/dbap_distance_based_amplitude_panning.py	output_channels , source_x , source_y , rolloff	DBAP
spatial_and_multichannel.binaural_itd_ild_synthesis	Binaural ITD/ILD synthesis	imag-ing_and_panning	src/pvx/algorithms/spatial_and_multichannel/imag-ing_and_panning/binaural_itd_ild_synthesis.py	azimuth_deg , itd_max_ms , ild_db	None
spatial_and_multichannel.-transaural_crosstalk_-cancellation	Transaural crosstalk cancellation	imag-ing_and_panning	src/pvx/algorithms/spatial_and_multichannel/imag-ing_and_panning/transaural_crosstalk_-cancellation.py	cancellation , delay_ms	Crossstalk cancellation
spatial_and_multichannel.stereo_width_frequency_dependent_control	Stereo width (frequency-dependent control)	imag-ing_and_panning	src/pvx/algorithms/spatial_and_multichannel/imag-ing_and_panning/stereo_width_frequency_dependent_control.py	width_low , width_high , crossover_hz	None

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
spatial_and_multichannel.phase_aligned_mid_side_field_rotation	Phase-aligned mid/side field rotation	imagining_and_panning	src/pvx/algorithms/spatial_and_multichannel/imaging_and_panning/phase_aligned_mid_side_field_rotation.py	rotation_deg	None
spatial_and_multichannel.pvx_interchannel_phase_locking	pvx interchannel phase locking	phase_vocoder_spatial	src/pvx/algorithms/spatial_and_multichannel/phase_vocoder_spatial/pvx_interchannel_phase_locking.py	lock_strength	Phase vocoder, Phase locking
spatial_and_multichannel.pvx_spatial_transient_preservation	pvx spatial transient preservation	phase_vocoder_spatial	src/pvx/algorithms/spatial_and_multichannel/phase_vocoder_spatial/pvx_spatial_transient_preservation.py	transient_threshold, phase_smooth, preserve_amount	Phase vocoder, Transient preservation
spatial_and_multichannel.pvx_interaural_coherence_shaping	pvx interaural coherence shaping	phase_vocoder_spatial	src/pvx/algorithms/spatial_and_multichannel/phase_vocoder_spatial/pvx_interaural_coherence_shaping.py	coherence_target	Phase vocoder
spatial_and_multichannel.pvx_directional_spectral_warp	pvx directional spectral warp	phase_vocoder_spatial	src/pvx/algorithms/spatial_and_multichannel/phase_vocoder_spatial/pvx_directional_spectral_warp.py	warp_amount, azimuth_deg	Phase vocoder

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
spatial_and_multichannel.pvx_multichannel_time_alignment	pvx multichannel time alignment	phase_vocoder_spatial	src/pvx/algorithms/spatial_and_multichannel/phase_vocoder_spatial/pvx_multichannel_time_alignment.py	max_lag	Phase vocoder
spatial_and_multichannel.pvx_spatial_freeze_and_trajectory	pvx spatial freeze and trajectory	phase_vocoder_spatial	src/pvx/algorithms/spatial_and_multichannel/phase_vocoder_spatial/pvx_spatial_freeze_and_trajectory.py	frame_ratio, orbit_hz	Phase vocoder
spatial_and_multichannel.multichannel_wiener_postfilter	Multichannel Wiener postfilter	multichannel_restoration	src/pvx/algorithms/spatial_and_multichannel/multichannel_restoration/multichannel_wiener_postfilter.py	noise_floor	None
spatial_and_multichannel.coherence_based_dereverb_multichannel	Coherence-based dereverb (multichannel)	multichannel_restoration	src/pvx/algorithms/spatial_and_multichannel/multichannel_restoration/coherence_based_dereverb_multichannel.py	coherence_threshold, decay	None
spatial_and_multichannel.multichannel_noise_psd_tracking	Multichannel noise PSD tracking	multichannel_restoration	src/pvx/algorithms/spatial_and_multichannel/multichannel_restoration/multichannel_noise_psd_tracking.py	alpha, floor	None

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
spatial_and_multichannel.phase_consistent_multichannel_denoise	Phase-consistent multichannel denoise	multichannel_restoration	src/pvx/algorithms/spatial_and_multichannel/multichannel_restoration/phase_consistent_multichannel_denoise.py	reduction_db, floor	None
spatial_and_multichannel.microphone_array_calibration_tones	Microphone-array calibration tones	multichannel_restoration	src/pvx/algorithms/spatial_and_multichannel/multichannel_restoration/microphone_array_calibration_tones.py	tone_hz, apply_correction	None
spatial_and_multichannel.cross_channel_click_pop_repair	Cross-channel click/pop repair	multichannel_restoration	src/pvx/algorithms/spatial_and_multichannel/multichannel_restoration/cross_channel_click_pop_repair.py	spike_threshold	None
spatial_and_multichannel.rotating_speaker_doppler_field	Rotating-speaker Doppler field	creative_spatial_fx	src/pvx/algorithms/spatial_and_multichannel/creative_spatial_fx/rotating_speaker_doppler_field.py	output_channels, rotation_hz, depth_ms	None
spatial_and_multichannel.binaural_motion_trajectory_designer	Binaural motion trajectory designer	creative_spatial_fx	src/pvx/algorithms/spatial_and_multichannel/creative_spatial_fx/binaural_motion_trajectory_designer.py	trajectory, trajectory_hz, width, itd_ms	None

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
spatial_and_multichannel.stochastic_spatial_diffusion_cloud	Stochastic spatial diffusion cloud	creative_spatial_fx	src/pvx/algorithms/spatial_and_multichannel/creative_spatial_fx/stochastic_spatial_diffusion_cloud.py	output_channels, diffusion, max_delay_ms, seed	None
spatial_and_multichannel.decorrelated_reverb_upmix	Decorrelated reverb upmix	creative_spatial_fx	src/pvx/algorithms/spatial_and_multichannel/creative_spatial_fx/decorrelated_reverb_upmix.py	output_channels, decay_s, rir_length, mix, seed	None
spatial_and_multichannel.spectral_spatial_granulator	Spectral spatial granulator	creative_spatial_fx	src/pvx/algorithms/spatial_and_multichannel/creative_spatial_fx/spectral_spatial_granulator.py	output_channels, grain, spread_semitones, density, seed	None
spatial_and_multichannel.spatial_freeze_resynthesis	Spatial freeze resynthesis	creative_spatial_fx	src/pvx/algorithms/spatial_and_multichannel/creative_spatial_fx/spatial_freeze_resynthesis.py	output_channels, frame_ratio, phase_drift, transform	None

pvx Group: Spectral and Time-Frequency Transforms

Source: groups/spectral_time_frequency_transforms.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Theme: Spectral and Time-Frequency Transforms

Folder: [spectral_time_frequency_transforms](#) | **Algorithms:** 8

Subgroups: [core \(8\)](#)

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
<code>spectral_time_frequency_transforms.constant_q_transform_m_cqt_processing</code>	Constant-Q Transform (CQT) processing	core	<code>src/pvx/algorithms/spectral_time_frequency_transforms/constant_q_transform_cqt_processing.py</code>	None (generic/default path)	Constant-Q transform (CQT)
<code>spectral_time_frequency_transforms.variable_q_transform_m_vqt</code>	Variable-Q Transform (VQT)	core	<code>src/pvx/algorithms/spectral_time_frequency_transforms/variable_q_transform_m_vqt.py</code>	None (generic/default path)	Variable-Q transform (VQT)
<code>spectral_time_frequency_transforms.ngt_based_processing</code>	NSGT-based processing	core	<code>src/pvx/algorithms/spectral_time_frequency_transforms/ngt_based_processing.py</code>	None (generic/default path)	NSGT
<code>spectral_time_frequency_transforms.reassigned_spectrogram_methods</code>	Reassigned spectrogram methods	core	<code>src/pvx/algorithms/spectral_time_frequency_transforms/reassigned_spectrogram_methods.py</code>	None (generic/default path)	Reassigned spectrogram
<code>spectral_time_frequency_transforms.synchrosqueezed_stft</code>	Synchrosqueezed STFT	core	<code>src/pvx/algorithms/spectral_time_frequency_transforms/synchrosqueezed_stft.py</code>	None (generic/default path)	None
<code>spectral_time_frequency_transforms.chirplet_transform_analysis</code>	Chirplet transform analysis	core	<code>src/pvx/algorithms/spectral_time_frequency_transforms/chirplet_transform_analysis.py</code>	None (generic/default path)	Chirplet transform
<code>spectral_time_frequency_transforms.wavelet_packet_processing</code>	Wavelet packet processing	core	<code>src/pvx/algorithms/spectral_time_frequency_transforms/wavelet_packet_processing.py</code>	None (generic/default path)	Wavelet packet

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
<code>spectral_time_frequency_transforms.multi_window_stft_fusion</code>	Multi-window STFT fusion	core	<code>src/pvx/algorithms/spectral_time_frequency_transforms/multi_window_stft_fusion.py</code>	None (generic/default path)	None

pvx Group: Time-Scale and Pitch Core

Source: groups/time_scale_and_pitch_core.html

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

Theme: Time-Scale and Pitch Core

Folder: `time_scale_and_pitch_core` | **Algorithms:** 7

Subgroups: `core (7)`

Algorithm ID	Name	Subgroup	Module Path	Parameter keys	Concept links
<code>time_s-scale_and_pitch_core.wsolala_waveform_similarity_overlap_add</code>	WSOLA (Waveform Similarity Overlap-Add)	core	<code>src/pvx/algorithms/time_s-scale_and_pitch_core/wsola_waveform_similarity_overlap_add.py</code>	<code>stretch</code> , <code>grain_size</code> , <code>hop</code>	WSOLA
<code>time_s-scale_and_pitch_core.td_psola</code>	TD-PSOLA	core	<code>src/pvx/algorithms/time_s-scale_and_pitch_core/td_psola.py</code>	<code>semitones</code> , <code>stretch</code>	TD-PSOLA
<code>time_s-scale_and_pitch_core.lp_psola</code>	LP-PSOLA	core	<code>src/pvx/algorithms/time_s-scale_and_pitch_core/lp_psola.py</code>	<code>semitones</code>	LP-PSOLA
<code>time_s-scale_and_pitch_core.multi_resolution_phase_vocoder</code>	Multi-resolution phase vocoder	core	<code>src/pvx/algorithms/time_s-scale_and_pitch_core/multi_resolution_phase_vocoder.py</code>	<code>stretch</code>	Phase vocoder
<code>time_s-scale_and_pitch_core.harmonic_percussive_split_tsm</code>	Harmonic/percussive split TSM	core	<code>src/pvx/algorithms/time_s-scale_and_pitch_core/harmonic_percussive_split_tsm.py</code>	<code>harmonic_stretch</code> , <code>percussive_stretch</code>	None
<code>time_s-scale_and_pitch_core.beat_synchronous_time_warping</code>	Beat-synchronous time warping	core	<code>src/pvx/algorithms/time_s-scale_and_pitch_core/beat_synchronous_time_warping.py</code>	<code>stretch</code>	None
<code>time_s-scale_and_pitch_core.nonlinear_time_maps</code>	Nonlinear time maps (curves, anchors, spline timing)	core	<code>src/pvx/algorithms/time_s-scale_and_pitch_core/nonlinear_time_maps.py</code>	<code>curve</code> , <code>stretch</code> , <code>fmin</code> , <code>fmax</code>	None

References (143 papers)

Source: [papers.html](#)

Acronym primer

- application programming interface (API)
- command-line interface (CLI)
- path environment variable (PATH)
- digital signal processing (DSP)
- short-time Fourier transform (STFT)
- inverse short-time Fourier transform (ISTFT)
- fast Fourier transform (FFT)
- discrete Fourier transform (DFT)
- central processing unit (CPU)
- graphics processing unit (GPU)
- Compute Unified Device Architecture (CUDA)
- comma-separated values (CSV)
- JavaScript Object Notation (JSON)
- HyperText Markup Language (HTML)
- Portable Document Format (PDF)
- continuous integration (CI)
- fundamental frequency (F0)
- waveform similarity overlap-add (WSOLA)
- input/output (I/O)
- root-mean-square (RMS)
- loudness units relative to full scale (LUFS)
- signal-to-noise ratio (SNR)

This bibliography collects foundational and directly related literature that informed pvx's phase-vocoder-centric architecture and the broader DSP algorithm roadmap.

Total references: **143** across **8** categories.

Links point to DOI pages, publisher archives, arXiv, standards documents, project docs, or Google Scholar queries where an official landing page can vary by publisher access.

Categories

- Phase Vocoder Foundations (21)
- Time-Scale and Pitch Methods (14)
- Pitch Detection and Tracking (17)
- Time-Frequency and Transform Methods (19)
- Separation and Decomposition (21)
- Denoising, Dereverberation, and Spatial Audio (31)
- Loudness, Dynamics, and Mastering (10)
- ML Audio and Neural Vocoding (10)

Phase Vocoder Foundations

Year	Authors	Title	Venue	Link type	Link
2013	M. Dolson	History of the Phase Vocoder	CCRMA Note	scholar	Link
2012	B. Zavalishin	The Art of VA Filter Design (sections on phase and spectral processing)	Book	scholar	Link
2009	N. Moreau	Toolbox for Time-Scale Modification and Pitch-Shifting of Audio Signals	DAFx	scholar	Link
2003	A. Röbel	A New Approach to Transient Processing in the Phase Vocoder	DAFx	web	Link
2002	C. Duxbury; M. Davies; M. Sandler	Improved Time-Scaling of Musical Audio Using Phase Locking at Transients	AES Convention	scholar	Link
2000	J. Bonada	Automatic Technique in Frequency Domain for Near-Lossless Time-Scale Modification of Audio	ICMC	scholar	Link
1999	S. M. Bernsee	Pitch Shifting Using the Fourier Transform	DAFX Workshop Note	web	Link
1999	J. Laroche; M. Dolson	New Phase-Vocoder Techniques for Pitch-Shifting, Harmonizing and Other Exotic Effects	WASPAA	doi	Link

Year	Authors	Title	Venue	Link type	Link
1999	J. Laroche; M. Dolson	Improved Phase Vocoder Time-Scale Modification of Audio	IEEE Trans. Speech and Audio Processing	doi	Link
1998	S. Disch	A New Phase Vocoder Technique for Time-Scale Modification of Audio Signals	DAFx	scholar	Link
1995	M. Puckette	Phase-Locked Vocoder	ICMC	scholar	Link
1990	X. Serra; J. O. Smith	Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition	Computer Music Journal	publisher_or_standard	Link
1989	R. Bristow-Johnson; M. Bogdanowicz	Phase Vocoder Done Right	AES Preprint	scholar	Link
1986	M. Dolson	The Phase Vocoder: A Tutorial	Computer Music Journal	publisher_or_standard	Link
1986	R. J. McAulay; T. F. Quatieri	Speech Analysis/Synthesis Based on a Sinusoidal Representation	IEEE Trans. ASSP	scholar	Link
1985	N. Roucos; A. Wilgus	High Quality Time-Scale Modification for Speech	ICASSP	scholar	Link
1984	D. W. Griffin; J. S. Lim	Signal Estimation from Modified Short-Time Fourier Transform	IEEE Trans. ASSP	doi	Link
1980	M. R. Portnoff	Time-Scale Modification of Speech Based on Short-Time Fourier Analysis	IEEE Trans. ASSP	scholar	Link
1977	J. B. Allen; L. R. Rabiner	A Unified Approach to Short-Time Fourier Analysis and Synthesis	Proceedings of the IEEE	doi	Link
1976	M. R. Portnoff	Implementation of the Digital Phase Vocoder Using the Fast Fourier Transform	IEEE Trans. ASSP	scholar	Link

Year	Authors	Title	Venue	Link type	Link
1966	J. L. Flanagan; R. M. Golden	Phase Vocoder	Bell System Technical Journal	doi	Link

Time-Scale and Pitch Methods

Year	Authors	Title	Venue	Link type	Link
2016	J. Driedger; M. Müller	A Review of Time-Scale Modification of Music Signals	Applied Sciences	doi	Link
2015	M. Riess; A. R. Chhetri	Transient Preservation in Time-Scale Modification	WASPAA	scholar	Link
2014	J. Driedger; T. Pratzlich; M. Muller	Let It Bee - Towards NMF-Inspired Audio Mosaicing	ISMIR	scholar	Link
2011	M. Le Roux; E. Vincent	Consistent Wiener Filtering for Audio Source Separation and Time-Frequency Processing	IEEE Signal Processing Letters	scholar	Link
2010	D. S. Hamon; A. Lazarides	Improved WSOLA for Real-Time Time-Scale Modification	AES Convention	scholar	Link
2010	A. Roebel	A Shape-Invariant Phase Vocoder for Speech Transformation	Interspeech	scholar	Link
2008	J. Bonada	Wide-Band Harmonic Sinusoidal Modeling for Time-Scale and Pitch-Scale Modification	DAFx	scholar	Link
2005	A. Röbel; X. Rodet	Efficient Spectral Envelope Estimation and Its Application to Pitch Shifting and Envelope Preservation	DAFx	web	Link
2004	A. de Cheveigne	Pitch and Time Manipulation of Speech	Tutorial	scholar	Link

Year	Authors	Title	Venue	Link type	Link
2003	S. M. Bernsee	Time Stretching and Pitch Shifting of Audio Signals - An Overview	Online article	web	Link
2003	J. Laroche	About this Phasiness Business	DAFx	schoolar	Link
2002	S. Arfib; D. Keiler; U. Zölzer	DAFx: Digital Audio Effects (chapter references on pitch/time processing)	Wiley	schoolar	Link
1993	W. Verhelst; M. Roelands	An Overlap-Add Technique Based on Waveform Similarity (WSOLA) for High Quality Time-Scale Modification of Speech	ICASSP	schoolar	Link
1990	E. Moulines; F. Charpentier	Pitch-Synchronous Waveform Processing Techniques for Text-to-Speech Synthesis Using Diphones	Speech Communication	doi	Link

Pitch Detection and Tracking

Year	Authors	Title	Venue	Link type	Link
2018	S. Bock; M. Korzeniowski	piano_transcription - F0 and onset tracking	ISMIR	schoolar	Link
2018	J. W. Kim; J. Salamon; P. Li; J. P. Bello	CREPE: A Convolutional Representation for Pitch Estimation	ICASSP	arxiv	Link
2017	R. Bittner et al.	Deep Salience Representations for F0 Estimation in Polyphonic Music	ISMIR	arxiv	Link
2015	B. W. Schuller et al.	Paralinguistics and robust pitch tracking methods	IEEE	schoolar	Link
2014	M. Mauch; S. Dixon	pYIN: A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions	ICASSP	doi	Link

Year	Authors	Title	Venue	Link type	Link
2012	K. Dressler	Sinusoidal Extraction using Frequency-Domain Matching Pursuit	DAFx	scholar	Link
2008	A. Camacho; J. G. Harris	A Sawtooth Waveform Inspired Pitch Estimator for Speech and Music (SWIPE)	JASA	doi	Link
2006	A. Klapuri	Multiple Fundamental Frequency Estimation by Harmonicity and Spectral Smoothness	IEEE TASLP	scholar	Link
2005	M. McLeod; G. Wyvill	A Smarter Way to Find Pitch	ICMC	scholar	Link
2002	A. de Cheveigné; H. Kawahara	YIN, a Fundamental Frequency Estimator for Speech and Music	JASA	doi	Link
2001	H. Kawahara; A. de Cheveigne	STRAIGHT, Exploitation of the Other Aspects of Vocal Source Information	ICASSP	scholar	Link
2000	S. Kum; C. L. Nikias	Robust F0 Estimation in Noise	IEEE	scholar	Link
1999	S. Ahmadi; H. Spanias	Cepstrum-Based Pitch Detection Using FFT	Conference	scholar	Link
1995	D. Talkin	A Robust Algorithm for Pitch Tracking (RAPT)	In Speech Coding and Synthesis	scholar	Link
1993	P. Boersma	Accurate Short-Term Analysis of the Fundamental Frequency and the Harmonics-to-Noise Ratio of a Sampled Sound	IFA Proceedings	scholar	Link
1976	L. R. Rabiner; M. J. Cheng; A. E. Rosenberg; C. A. McGonegal	A Comparative Performance Study of Several Pitch Detection Algorithms	IEEE Trans. ASSP	scholar	Link

Year	Authors	Title	Venue	Link type	Link
1967	A. M. Noll	Cepstrum Pitch Determination	JASA	scholar	Link

Time-Frequency and Transform Methods

Year	Authors	Title	Venue	Link type	Link
2019	M. Doring; M. M. Kokuer	Practical NSGT audio applications	DAFx	scholar	Link
2012	S. Essid; G. Richard	Musical signal analysis with reassignment methods	Book chapter	scholar	Link
2011	I. Daubechies; J. Lu; H.-T. Wu	Synchrosqueezed Wavelet Transforms	Applied and Computational Harmonic Analysis	doi	Link
2011	G. Velasco; N. Holighaus; M. Dörfler; T. Grill	Constructing an Invertible Constant-Q Transform with Nonstationary Gabor Frames	DAFx	web	Link
2010	A. V. Oppenheim; R. W. Schafer	Discrete-Time Signal Processing (STFT and filter-bank chapters)	Pearson	scholar	Link
2010	C. Schörkhuber; A. Klapuri	Constant-Q Transform Toolbox for Music Processing	SMC	scholar	Link
2006	A. C. Gilbert; M. J. Strauss	Approximation of signals in sparse Fourier dictionaries	IEEE	scholar	Link
2002	P. Flandrin; F. Auger; E. Chassande-Mottin	Time-Frequency Reassignment: From Principles to Algorithms	Applications in Time-Frequency Signal Processing	scholar	Link

Year	Authors	Title	Venue	Link type	Link
2001	K. Grochenig	Foundations of Time-Frequency Analysis	Birkhauser	scholar	Link
1998	P. Flandrin	Time-Frequency/Time-Scale Analysis	Academic Press	scholar	Link
1995	F. Auger; P. Flandrin	Improving the Readability of Time-Frequency and Time-Scale Representations by the Reassignment Method	IEEE Trans. SP	doi	Link
1995	D. L. Donoho	De-Noising by Soft-Thresholding	IEEE Trans. Information Theory	scholar	Link
1993	S. Mallat; Z. Zhang	Matching Pursuits with Time-Frequency Dictionaries	IEEE Trans. Signal Processing	scholar	Link
1992	R. R. Coifman; M. V. Wickerhauser	Entropy-Based Algorithms for Best Basis Selection	IEEE Trans. IT	doi	Link
1992	J. C. Brown; M. S. Puckette	An Efficient Algorithm for the Calculation of a Constant Q Transform	JASA	doi	Link
1991	S. Mann; S. Haykin	The Chirplet Transform: Physical Considerations	IEEE Trans. SP	scholar	Link
1991	J. C. Brown	Calculation of a Constant Q Spectral Transform	JASA	doi	Link
1989	L. Cohen	Time-Frequency Distributions - A Review	Proceedings of the IEEE	scholar	Link
1989	S. Mallat	A Theory for Multiresolution Signal Decomposition: The Wavelet Representation	IEEE Trans. PAMI	doi	Link

Separation and Decomposition

Year	Authors	Title	Venue	Link type	Link
2019	J. Le Roux; S. Wisdom; H. Erdogan; J. R. Hershey	SDR half-baked or well done?	ICASSP	scholar	Link
2019	F.-R. Stöter; S. Uhlich; A. Liutkus; Y. Mitsufuji	Open-Unmix - A Reference Implementation for Music Source Separation	Journal of Open Source Software	doi	Link
2019	F.-R. Stoter; S. Uhlich; A. Liutkus; Y. Mitsufuji	Open-Unmix	ISMIR	scholar	Link
2019	A. Défossez et al.	Music Source Separation in the Waveform Domain	arXiv	arxiv	Link
2019	A. Defossez et al.	Music Source Separation in the Waveform Domain	ISMIR	scholar	Link
2018	N. Takahashi et al.	MMDenseLSTM for source separation	ICASSP	scholar	Link
2017	A. Jansson et al.	Singing Voice Separation with Deep U-Net Convolutional Networks	ISMIR	arxiv	Link
2011	N. Ono	Stable and fast update rules for independent low-rank matrix analysis based on auxiliary function technique	WASPAA	scholar	Link
2011	E. J. Candès; X. Li; Y. Ma; J. Wright	Robust Principal Component Analysis?	JACM	doi	Link
2011	E. J. Candes; X. Li; Y. Ma; J. Wright	Robust Principal Component Analysis?	Journal of ACM	scholar	Link
2010	D. Fitzgerald	Harmonic/Percussive Separation Using Median Filtering	DAFx	web	Link

Year	Authors	Title	Venue	Link type	Link
2009	C. Févotte; N. Bertin; J.-L. Durrieu	Nonnegative Matrix Factorization with the Itakura-Saito Divergence	Neural Computation	scholar	Link
2007	T. Virtanen	Monaural Sound Source Separation by Nonnegative Matrix Factorization with Temporal Continuity and Sparseness Criteria	IEEE Trans. Audio, Speech, and Language Processing	scholar	Link
2007	P. Smaragdis	Convulsive Speech Bases and Their Application to Supervised Speech Separation	IEEE TASLP	scholar	Link
2001	D. D. Lee; H. S. Seung	Algorithms for Non-negative Matrix Factorization	NIPS	doi	Link
2000	A. Hyvärinen; E. Oja	Independent Component Analysis: Algorithms and Applications	Neural Networks	doi	Link
2000	A. Hyvärinen; E. Oja	Independent Component Analysis: Algorithms and Applications	Neural Networks	scholar	Link
1999	D. D. Lee; H. S. Seung	Learning the Parts of Objects by Non-negative Matrix Factorization	Nature	doi	Link
1999	A. Hyvärinen	Fast and Robust Fixed-Point Algorithms for Independent Component Analysis	IEEE Trans. Neural Networks	doi	Link
1995	A. J. Bell; T. J. Sejnowski	An Information-Maximization Approach to Blind Separation and Blind Deconvolution	Neural Computation	scholar	Link
1994	P. Comon	Independent Component Analysis, A New Concept?	Signal Processing	doi	Link

Denoising, Dereverberation, and Spatial Audio

Y e ar	Authors	Title	Venue	Link type	Li n k
2023	EBU	EBU R128: Loudness Normalisation and Permitted Maximum Level of Audio Signals	Recommendation	pub- lish- er_or_s- tandard	Li nk
2019	M. H. C. de Gesmundo et al.	ViSQOL v3: An Open Source Production Ready Objective Speech and Audio Metric	QoMEX	schol- ar	Li nk
2019	A. Pandey; D. Wang	A New Framework for CNN-Based Speech Enhancement in the Time Domain	IEEE TASLP	schol- ar	Li nk
2018	N. W. D. Evans et al.	The voicehome and CHiME challenges	IEEE	schol- ar	Li nk
2018	J.-M. Valin	A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement (RNNNoise)	MMSP / arXiv	arxiv	Li nk
2018	J.-M. Valin	A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement	MMSP	schol- ar	Li nk
2016	K. Kinoshita et al.	A summary of the REVERB challenge	EURASIP Journal	schol- ar	Li nk
2015	ITU-R	BS.1770-4: Algorithms to Measure Audio Programme Loudness and True-Peak Audio Level	Recommendation	pub- lish- er_or_s- tandard	Li nk
2013	P. C. Loizou	Speech Enhancement: Theory and Practice	CRC Press	schol- ar	Li nk

Year	Authors	Title	Venue	Link type	Link
2012	A. Hines et al.	VISQOL: An Objective Speech Quality Model	EURASIP	scholar	Link
2012	N. D. Gaubitch; P. A. Naylor	Speech Dereverberation	Springer Handbook	scholar	Link
2012	Y. Yoshioka; T. Nakatani	Generalization of Multi-Channel Linear Prediction Methods for Blind MIMO Impulse Response Shortening	IEEE TASLP	scholar	Link
2012	T. Yoshioka; T. Nakatani	Generalization of Multi-Channel Linear Prediction Methods for Blind MIMO Impulse Response Shortening	IEEE TASLP	scholar	Link
2011	K. K. Paliwal	Importance of phase in speech enhancement	Speech Communication	scholar	Link
2011	C. H. Taal; R. C. Hendriks; R. Heusdens; J. Jensen	An Algorithm for Intelligibility Prediction of Time-Frequency Weighted Noisy Speech (STOI)	IEEE TASLP	doi	Link
2011	C. Taal et al.	An Algorithm for Intelligibility Prediction of Time-Frequency Weighted Noisy Speech	IEEE TASLP	scholar	Link
2010	T. Nakatani; M. Miyoshi; K. Kinoshita	Speech Dereverberation Based on Variance-Normalized Delayed Linear Prediction	IEEE Trans. Audio, Speech, and Language Processing	doi	Link
2009	I. Cohen; B. Berdugo	Speech Enhancement for Non-Stationary Noise Environments	Signal Processing	scholar	Link

Year	Authors	Title	Venue	Link type	Link
2001	A. W. Rix; J. G. Beerends; M. P. Hollier; A. P. Hekstra	Perceptual Evaluation of Speech Quality (PESQ)	ICASSP	doi	Link
2001	A. Rix et al.	Perceptual Evaluation of Speech Quality (PESQ)	ICASSP	scholar	Link
2001	R. Martin	Noise Power Spectral Density Estimation Based on Optimal Smoothing and Minimum Statistics	IEEE Trans. Speech and Audio Processing	doi	Link
1997	S. Pulkki	Virtual Sound Source Positioning Using Vector Base Amplitude Panning	JAES	scholar	Link
1996	P. Scalart; J. V. Filho	Speech Enhancement Based on a Priori Signal to Noise Estimation	ICASSP	scholar	Link
1985	Y. Ephraim; D. Malah	Speech Enhancement Using a Minimum Mean-Square Error Log-Spectral Amplitude Estimator	IEEE Trans. ASSP	doi	Link
1984	Y. Ephraim; D. Malah	Speech Enhancement Using a Minimum Mean-Square Error Short-Time Spectral Amplitude Estimator	IEEE Trans. ASSP	doi	Link
1979	S. F. Boll	Suppression of Acoustic Noise in Speech Using Spectral Subtraction	IEEE Trans. ASSP	doi	Link
1979	S. Boll	Suppression of Acoustic Noise in Speech Using Spectral Subtraction	IEEE TASSP	scholar	Link

Year	Authors	Title	Venue	Link type	Link
1976	C. Knapp; G. Carter	The Generalized Correlation Method for Estimation of Time Delay	IEEE Trans. ASSP	doi	Link
1972	O. L. Frost III	An Algorithm for Linearly Constrained Adaptive Array Processing	Proceedings of the IEEE	doi	Link
1969	J. Capon	High-Resolution Frequency-Wavenumber Spectrum Analysis	Proceedings of the IEEE	doi	Link
1949	N. Wiener	Extrapolation, Interpolation, and Smoothing of Stationary Time Series	Book	scholar	Link

Loudness, Dynamics, and Mastering

Year	Authors	Title	Venue	Link type	Link
2023	EBU	Tech 3343: Production Guidelines	Recommendation	publisher_or_standard	Link
2023	EBU	Tech 3342: Loudness Range	Recommendation	publisher_or_standard	Link
2023	EBU	Tech 3341: Loudness Metering: EBU Mode	Recommendation	publisher_or_standard	Link
2023	ITU-R	BS.1770-5: Algorithms to Measure Audio Programme Loudness and True-Peak Audio Level	Recommendation	publisher_or_standard	Link
2020	S. Stables et al.	A Study of Loudness Normalization in Streaming Services	AES Convention	scholar	Link

Year	Authors	Title	Venue	Link type	Link
2018	M. Fenton	Practical Dynamics Processing for Modern Music Production	AES Tutorial	scholar	Link
2015	M. Ballou	Handbook for Sound Engineers (Loudness and Dynamics chapters)	CRC Press	scholar	Link
2012	D. Giannoulis; M. Massberg; J. Reiss	Digital Dynamic Range Compressor Design	JAES	scholar	Link
2011	J. Reiss	Under the Hood of a Dynamic Range Compressor	AES	scholar	Link
2007	T. Lund	Loudness and True-Peak in Digital Audio	TC Electronic Whitepaper	scholar	Link

ML Audio and Neural Vocoding

Year	Authors	Title	Venue	Link type	Link
2021	S. Wisdom et al.	Differentiable Consistency Constraints for Improved Deep Speech Enhancement	ICASSP	scho lar	Link
2020	J. Kong et al.	HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis	NeurIPS	scho lar	Link
2020	J. Engel et al.	DDSP: Differentiable Digital Signal Processing	ICLR	arx- iv	Link
2019	R. Prenger et al.	WaveGlow: A Flow-based Generative Network for Speech Synthesis	ICASSP	scho lar	Link
2019	K. Kumar et al.	MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis	NeurIPS	scho lar	Link
2018	N. Kalchbrenner et al.	Efficient Neural Audio Synthesis	ICML	scho lar	Link
2018	C. Donahue; J. McAuley; M. Puckette	Adversarial Audio Synthesis	ICLR Workshop	scho lar	Link

Year	Authors	Title	Venue	Link type	Link
2017	Y. Wang et al.	Tacotron: Towards End-to-End Speech Synthesis	Interspeech	scho lar	Link
2017	A. Gibiansky et al.	Deep Voice 2: Multi-Speaker Neural Text-to-Speech	NeurIPS Workshop	scho lar	Link
2016	A. van den Oord et al.	WaveNet: A Generative Model for Raw Audio	ArXiv	arx- iv	Link