Spectrogram Denoising by Filtering Max-Trees

Gonzalo Romero-García, Alberto Martín-Izquierdo, Edwin Carlinet

Spectrogram denoising by filtering max-trees

Gonzalo Romero-García[1][0000−0003−1971−6204], Alberto Martín-Izquierdo[2], and Edwin Carlinet[1][0000−0001−5737−5266]

[1] Laboratoire de Recherche de l'EPITA, Le Kremlin-Bicêtre, France
{gonzalo.romero-garcia,edwin.carlinet}@epita.fr
[2] Universidad de Nebrija, Madrid, Spain
amartiiz@nebrija.es

Abstract. Spectrograms offer a time-frequency representation well-suited for audio analysis, where structured signals manifest as horizontal or vertical patterns and noise appears as irregular textures. In this paper, we propose a spectrogram denoising method based on mathematical morphology, and more precisely on max-trees. We treat the spectrogram as a grayscale image and apply a sequence of filters on its max-tree representation. We propose a method to isolate signal regions by filtering components according to contrast and shape. A binary mask is constructed from the remaining components, which is used to reconstruct the denoised signal by inverting the masked STFT. The method is highly interpretable and preserves the signal structure while effectively removing noise. We demonstrate its effectiveness on synthetic and real audio signals.

Keywords: Spectrograms · Max-tree · Mathematical morphology · Audio denoising

1 Introduction

Spectrograms are widely used in signal processing, with applications in fields such as audio analysis, biomedical signal processing, radar, and telecommunications. As a time-frequency representation, a spectrogram maps each point (t, f) in the time-frequency plane to a value $S(t, f)$, often interpreted as the local energy of the signal.

In this representation, stationary sinusoidal components manifest as horizontal lines, while transient events with brief durations appear as vertical lines. Stochastic components, such as noise, typically generate irregular patterns resembling a landscape of hills and valleys.

Denoising in the spectrogram domain consists in attenuating or removing the noisy part of the representation while preserving the relevant signal structures. In this work, we address the problem of spectrogram denoising by using image processing techniques.

In particular, we use *max-trees*, a hierarchical image representation that encodes the inclusion structure of connected components in the upper level sets of a grayscale image. This approach enables us to design simple, interpretable filters

that target signal structures in the spectrogram, while discarding noise-induced patterns. Our method constructs a binary mask from selected components, applies it to the STFT coefficients, and performs inverse STFT to recover the denoised signal. It combines time-frequency analysis with morphological image processing and offers a parametric and interpretable alternative to learning-based approaches.

The remainder of the paper is organized as follows. In Section 2, we review prior work on spectrogram denoising, specially those that uses morphological filtering. Section 3 presents the fundamental structures used throughout this work, namely the spectrogram representation and the max-tree. Section 4 details our proposed denoising pipeline. Experimental results are reported in Section 5, and we conclude in Section 6.

2 Related Work

Spectrogram-based noise suppression techniques have a long history, starting with Boll's seminal work on spectral subtraction [2]. This method estimates the noise spectrum during non-speech activity and subtracts it from the noisy signal. It remains influential, and several variants have since been proposed to improve performance under nonstationary noise conditions [15,13].

More recently, morphological operations have been proposed as nonlinear alternatives to traditional filtering methods. In [5], morphological erosion and dilation were applied to spectrograms preprocessed via spectral subtraction, resulting in enhanced perceptual quality. This approach was later refined in [6] with auditory-inspired structuring elements, further improving both enhancement and automatic speech recognition tasks.

The relevance of morphological processing extends beyond speech. In bioacoustics, Hussein [10] employed edge detection and crest factor computation for spectrogram enhancement, showing promising results on bat and bird calls. De Oliveira et al. [14] applied morphological opening to detect bird acoustic activity with reduced computational cost, validating their approach through comparative evaluations.

Additionally, morphological strategies have proven useful in surveillance and impact detection scenarios. Uchiyama and Tanabe [18] combined harmonic/percussive source separation with real-time morphological filtering to isolate hammering test signatures. Sharan and Moir [17] leveraged spectrogram-derived features for robust audio classification in noisy conditions, using a reduced spectrogram image feature coupled with SVM classifiers.

3 Fundamental structures

3.1 Spectrograms

We adopt a continuous framework for defining spectrograms, following [8], in order to maintain mathematical clarity. In practice, however, the resulting time-

frequency representations are sampled on a uniform grid, with a fixed time step and frequency resolution.

Let $x \in \mathcal{L}_\infty(\mathbb{R};\mathbb{C})$ denote the input signal and $g \in \mathcal{L}_1(\mathbb{R};\mathbb{C})$ a window function. The Short-Time Fourier Transform (STFT) of x with respect to g is defined as:

$$\begin{array}{rl} V_g[x] : \mathbb{R}\times\mathbb{R} \to \mathbb{C} & \\ (t,f) \mapsto \int_{\mathbb{R}} x(\tau)\,\overline{g(\tau-t)}\,e^{-2\pi i \tau f}\,d\tau & . \end{array} \tag{1}$$

The STFT is a time-frequency transform whose output is a complex-valued function with domain $\mathbb{R}\times\mathbb{R}$. In analytical applications, the phase is discarded, and only the magnitude is considered. The corresponding *spectrogram* is the squared modulus of the STFT:

$$S_g[x](t,f) = |V_g[x](t,f)|^2\,. \tag{2}$$

For interpretability, the spectrogram is often expressed in decibels (dB):

$$S_g^{\mathrm{dB}}[x](t,f) = 10\log_{10}\left(|S_g[x](t,f)|\right)\,. \tag{3}$$

Since the STFT can be written as $V_g[x](t,f) = e^{-2\pi i t f}(x * M_f g^*)(t)$ (where $M_f g(t) = e^{2\pi i t f} g(t)$ and $g^*(t) = \overline{g(-t)}$), we have the norm inequality

$$||V_g[x](t,f)||_\infty = ||x * M_f g^*||_\infty \le ||x||_\infty ||M_f g^*||_1 = ||x||_\infty ||g||_1\,. \tag{4}$$

In real world applications, our input signals x will be bounded by 1 (or can be rescaled to be so), so the normalization $||g||_1 = 1$ ensures that the spectrogram will also be bounded by 1, that is

$$0 \le S_g[x] \le 1 \qquad \text{and} \qquad -\infty \le S_g^{\mathrm{dB}}[x] \le 0\,. \tag{5}$$

In order to obtain a discretized grayscale image from a spectrogram, we quantize time and frequency with time resolution $\Delta t = 0.01\,\mathrm{s}$ and frequency resolution $\Delta f = 10\,\mathrm{Hz}$. Bounds for the grayscale range are necessary, so we define a lower bound of $-120\,\mathrm{dB}$ and the upper bound of $0\,\mathrm{dB}$ is given by the equation (5). Next, we quantize the intensity values using $1\,\mathrm{dB}$ intervals, obtaining a range $Q = \{-120, -119, \ldots, 0\}$. The desired grayscale image is then

$$(\mathbf{S}_{kn})_{n=0,\ldots,N}^{k=0,\ldots,K} \in Q^{K\times N}\,, \tag{6}$$

with $f_k = k\cdot\Delta f$ and $t_n = n\cdot\Delta t$.

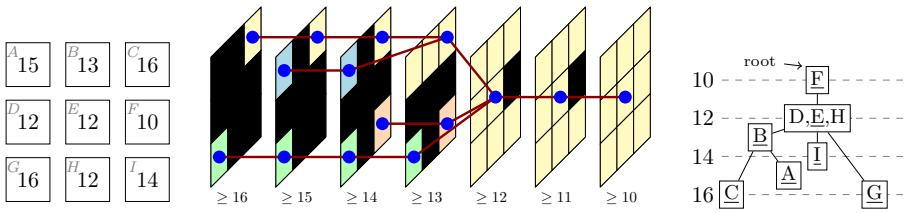
This representation enables us to apply image processing techniques, and in particular mathematical morphological ones. The geometry of a spectrogram is constrained by time-frequency properties and implies that the resulting image only contains specific shapes: horizontal structures correspond to sinusoids, vertical ones to transients, and noisy components appear as irregular patterns where valleys and hills alternate. This is illustrated in Figure 6a.

Since the image can be interpreted as a topographic map where the signals appear as mountains, it makes sense to apply morphology techniques based on level sets and hierarchical structures. We opted for the max-tree representation and we aim to apply successive filters in a pipeline to separate the mountains from the irregular landscape.

3.2 Max-Tree

Component trees are data structures that organize the objects in an image and represent their inclusion relationships. They are useful for image segmentation and analysis, as they allow a hierarchical representation of objects. These trees have received particular attention regarding their robustness to noise. For example, in [11], the authors studied the robustness of attribute profiles computed from these trees for the classification of hyperspectral image pixels. In [3], the stability of the tree of shapes is investigated for region of interest detection, which improves by 20 to 30% when using this structure. This robustness has even been studied more formally in [4], and some even use these trees to estimate the noise level in an image [7].

The max-tree, introduced by [16], is based on the notion of upper level sets of an image. Let u be a grayscale image; the upper level sets of u form the family $\mathcal{U} = [u \geq \lambda]_{\lambda \in \mathbb{R}}$, where $[u \geq \lambda]$ denotes the set of pixels with an intensity greater than or equal to λ. The max-tree is simply the set of connected components of the elements of $\mathcal{U}$. Since any two components are either nested or disjoint, this set of components forms a tree structure, as illustrated in Figure 1.



(a) Original image (b) Inclusion of upper-level-set decomposition (c) Max-tree

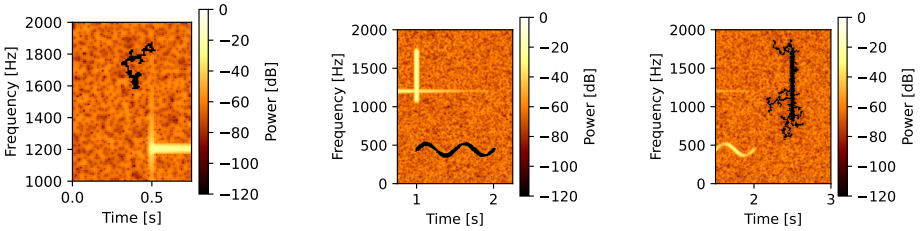
Fig. 1: Illustration of max-tree construction: (a) original image, (b) inclusion of upper-level-set decomposition, and (c) corresponding max-tree. Each node in the tree represents a connected component of the level-sets.

The max-tree applied on a spectrogram can contain different kinds of components: some components only contain noise, others only contain signal, and some (usually closer to the tree root) contain both noise and signal. Some examples of these components are illustrated in Figure 2.

Our goal now is to select the components that only contain signal and filter out the ones that contain noise. Once this is achieved, we can obtain a mask by taking the union of the selected components. This will allow us to produce a new spectrogram by masking the original STFT and applying the inverse STFT [1].

4 Processing Pipeline

We propose a three-stage pipeline to denoise an audio signal by applying morphological filters to the max-tree of its spectrogram:



(a) Component containing only noise

(b) Component containing only signal

(c) Mixed noise and signal component

Fig. 2: Examples of max-tree components: (a) noise-only, (b) signal-only, and (c) mixed noise and signal.

1. **Pre-processing:** We compute the STFT $V_g[x]$ with resolutions Δt, Δf, take its squared modulus, convert to dB, and quantize into Q levels to obtain the input spectrogram image $\mathbf{S}_{kn}$.
2. **Morphological filtering:** We build the max-tree $\mathcal{T}$ of the quantized spectrogram and apply, in any other, the following filters:
   - *Area filter*: remove components C_i with area $A_i > A_{\max}$,
   - *Height filter*: remove components C_i with height $H_i < H_{\min}$,
   - *Shape filter*: remove components C_i with average 8-neighbor count per pixel $R_i > R_{\min}$.

   Then, we form the binary mask M as the union of all remaining components.
3. **Reconstruction:** We mask the STFT coefficients, i.e. compute $V_g[x] \odot M$ and then perform the inverse STFT to obtain the denoised audio.

This pipeline is illustrated in Figure 3. Now, let us discuss in detail each step and expose the motivations and effects they have.

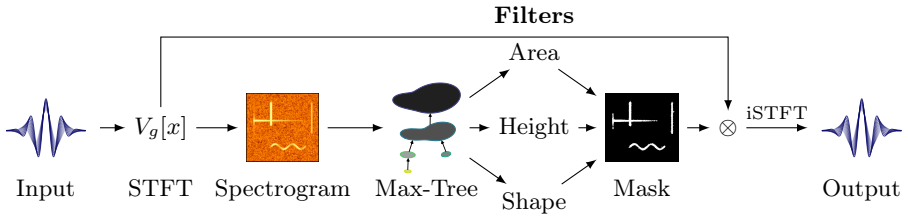


Fig. 3: Spectrogram processing pipeline.

4.1 Max-tree Construction

We compute the max-tree $\mathcal{T}$ of the image $\mathbf{S} \in Q^{K \times N}$, seen as a function from the discrete time-frequency domain $\Omega = \{0, \ldots, K-1\} \times \{0, \ldots, N-1\}$ to the set of quantized values Q. The max-tree is defined as a tree where the nodes (indexed by I) are the connected components $C_i \subseteq \Omega$ and the edges are given by $C_i \to C_j \Leftrightarrow C_i \subset C_j$ and $\{i' \in I : C_i \subset C_{i'} \subset C_j\} = \emptyset$. We can then encode the max-tree as the triplet

$$\mathcal{T} = (\mathcal{C}, p, \Lambda) \tag{7}$$

where $\mathcal{C} := \{C_i\}_{i \in I}$ is the set of components, $p : I \to I \cup \{\emptyset\}$ is the function that associates to each component index the index of its parent (and $\emptyset$ for the root), and $\Lambda := \{\lambda_i\}_{i \in I} \subseteq Q$ is the set of values.

4.2 Filtering by area

First, we apply a filter by area as a heuristic to remove components that are too large to only contain signal. The area of a component C_i is defined as $A_i = |C_i|$, i.e. the number of elements of the component.

This filter has additional benefits: it eliminates edge cases and reduces computation time for the following filters. We choose a value of $A_{\max} = 75\%$ of the image size. Our experiments show this value produces a very conservative filter, ensuring that the probability of losing signal is negligible.

4.3 Filtering by height

This is arguably the most critical filter. When applying the max-tree to noised signals, the noise is split into several components. Several of these display line-like morphological attributes locally (horizontal, vertical, or curved), although they do not correspond to signal. This phenomenon happens because their values are only slightly higher than those of the adjacent components.

Although these components do not contain signal and do not appear clearly as mountains, they are hard to distinguish from the rest of the noise: they are surrounded by a shallow ditch and thus remain disconnected from neighboring structures.

To eliminate them we introduce a new attribute, the *height*. For a component C_i, its height H_i is the maximum elevation of any child component relative to C_i. Formally.

$$H_i = \max_{j \in I,\, C_j \subseteq C_i} \{\lambda_j - \lambda_i\} \geq 0\,. \tag{8}$$

Figure 4 illustrates this attribute.

In a sense, the height encodes the local signal-to-noise ratio. Thus, we discard all components whose height falls below a threshold $H_{\min}$. In our experiments, we set $H_{\min} = 6\,\mathrm{dB}$ which is a value that preserves most signal components. This parameter should be tuned to the specific application.

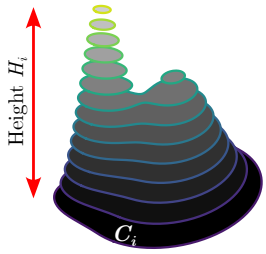
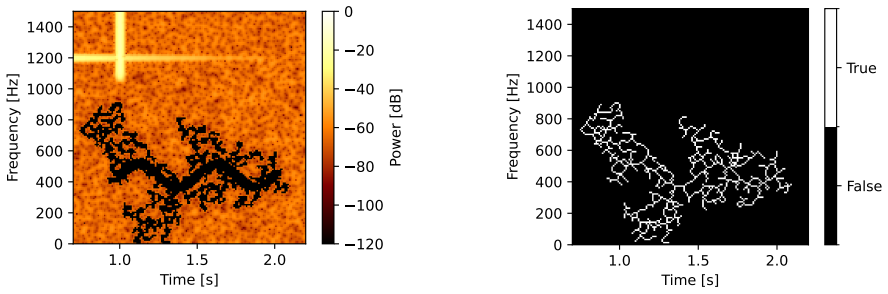


Fig. 4: Height parameter H_i of the component C_i

4.4 Filtering by shape

The last stage decides whether a surviving component has an acceptable *shape*. After the height filter, most components containing noise are removed. A few persist though, typically containing a mixture of signal with noise (see Figure 5).



(a) Component to thin

(b) Thinning of the component

Fig. 5: Thinning of a component that contains signal and noise.

We perform the following computations to remove the noisy components based on their shape:

Thinning: Each candidate component is thinned with the parallel algorithm of Guo and Hall [9], which iteratively erodes contour pixels while preserving eight-connectedness and every 2×2 block [12]. Let T denote the binary image of the resulting skeleton.

Neighbor count: For every pixel $p \in T$ we count its eight neighbors inside T:

$$N(p) = \sum_{q \in \mathcal{N}_8(p)} T(q) \quad \in \{0, \ldots, 8\}. \tag{9}$$

In practice, this is a convolution of T with a 3×3 kernel whose entries are all equal to 1 except for the central one which is equal to 0, followed by a point-wise multiplication with T. Pixels at line ends have $N(p) = 1$, interior line pixels give $N(p) = 2$, and bifurcations yield $N(p) \geq 3$.

Neighbor ratio: To characterize the global shape, we average the neighbor count *only over pixels with* $N(p) \geq 2$, avoiding the bias introduced by the endpoints of a line:

$$R_i = \frac{\sum_{p \in C_i^2} N(p)}{\left|C_i^2\right|} \; > \; 2 \quad \text{with} \quad C_i^2 = \{p \in C_i : N(p) \geq 2\}. \tag{10}$$

For a perfectly thinned line $R_i = 2$. Components whose skeleton is not a single line give a larger value (for Figure 5, $R_i \approx 2.52$).

We filter by keeping the components whose neighbor ratio is below a certain threshold $R_{\max}$. In our experiments, we choose a value of $R_{\max} = 2.1$ that allows for a small tolerance to the components that are not exactly lines.

4.5 Mask Construction

With the component-wise filters in place, we can now build the binary mask. A component is accepted only if it passes *all* the tests. Formally,

$$\mathcal{F}_i = (A_i \leq A_{\max}) \wedge (H_i \geq H_{\min}) \wedge (R_i \leq R_{\max}) \in \{\texttt{true}, \texttt{false}\}. \tag{11}$$

Let $I^\star = \{i \in I : \mathcal{F}_i = \texttt{true}\}$ be the indices of the surviving components. The final mask is the union,

$$M = \bigcup_{i \in I^\star} C_i. \tag{12}$$

5 Experimental results

We evaluate the proposed method on both synthetic and real audio signals. The synthetic data are built from a mixture of sinusoids, transients, and additive white noise, which is the signal already used for illustration in previous sections. For real data, we select several isolated musical-instrument notes and contaminate them with artificial noise.

It is difficult to use a quantitative measure for evaluating the quality of the denoised signal. In order to evaluate it visually, we show the original spectrograms and the mask we retrieve. Auditory perception was also used as a qualitative method in the experimental setup to assess the quality of the denoising, but it is not included in this work due to the limitations of the written medium.

All experiments use a Gaussian analysis window $g(t) = \exp\left(-\frac{1}{2}\frac{t^2}{\sigma^2}\right)$, $t \in [-T/2, T/2]$ with duration T and relative width $\sigma = \sqrt{\frac{\Delta t}{2\pi \Delta f}} \approx 0.0126\,\text{s}$; this value produces an isometric the time-frequency plane [8], which in particular means that the horizontal and vertical lines of same power have the same width. The window is normalized so that the upper bound of Equation (5) holds, i.e. divided by $\|g\|_1$. Table 1 lists all analysis and filter parameters.

Parameter	Symbol	Value	Unit
Sampling rate	f_s	20 000	Hz
Time resolution	Δt	0.01	s
Frequency resolution	Δf	10	Hz
Window duration	T	0.1	s
Gaussian width	σ	0.0126	s

(a) Spectrogram parameters.

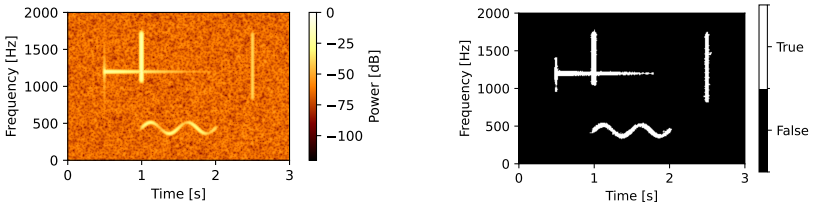
Filter	Symbol	Value	Unit
Height	$H_{\min}$	6	dB
Neighbor ratio	$R_{\max}$	2.1	n.u.
Area	$A_{\max}$	75	%

(b) Filter parameters.

Table 1: Parameters used in the pipeline.

5.1 Synthetic example

Figure 6 shows a synthetic signal composed of an AM/FM sinusoid, a decaying sinusoid, and two transients (one intersecting the decaying sinusoid) mixed with white noise at SNR = 3 dB. The recovered mask (Figure 6b) closely matches the ground-truth signal components.



(a) Input spectrogram

(b) Resulting mask

Fig. 6: Spectrogram computed from a synthetic audio signal and the resulting binary mask M.

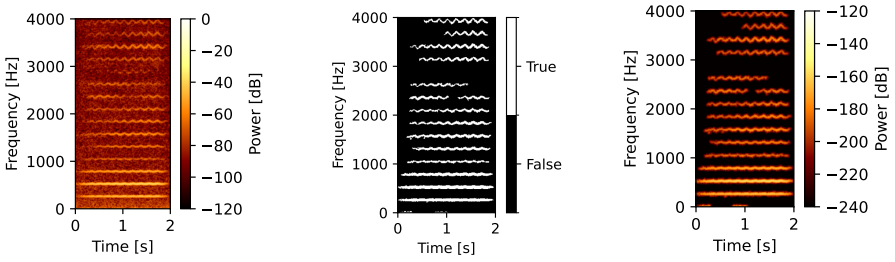
5.2 Violin with vibrato

To assess robustness on non-stationary harmonics, we process a violin note with natural vibrato, down-sampled to 20 kHz and mixed with pink noise at −6 dB SNR.[3] Figure 7 confirms that the pipeline preserves the undulating harmonics while largely removing the broadband noise.

5.3 Parameter-sensitivity analysis.

Finally, we investigate the effect of the two main thresholds, height $H_{\min}$ and neighbor ratio $R_{\max}$. Figure 8 sweeps each parameter independently while hold-

[3] Pink noise exhibits a $1/f$ power spectrum.

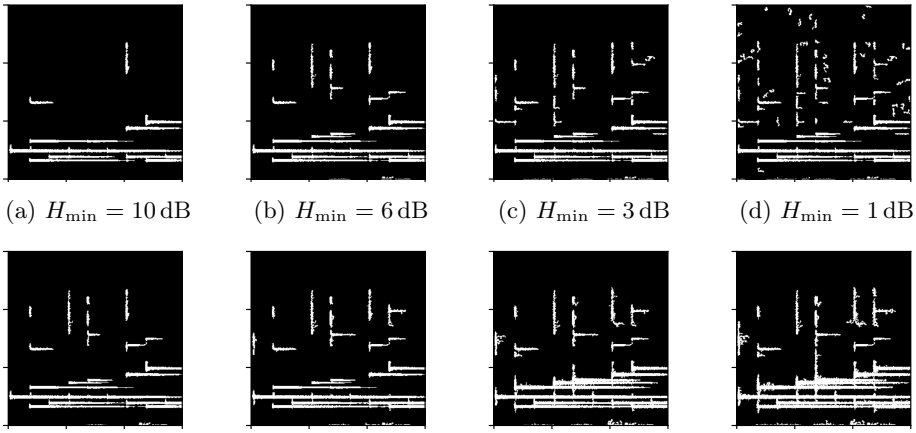


(a) Spectrogram of a violin signal with vibrato

(b) Mask obtained after filtering

(c) Result after applying the denoising pipeline

Fig. 7: Example of a violin signal with vibrato with added pink noise.



(a) $H_{\min} = 10\,\mathrm{dB}$ (b) $H_{\min} = 6\,\mathrm{dB}$ (c) $H_{\min} = 3\,\mathrm{dB}$ (d) $H_{\min} = 1\,\mathrm{dB}$

(e) $R_{\max} = 2.01$ (f) $R_{\max} = 2.2$ (g) $R_{\max} = 2.3$ (h) $R_{\max} = 2.4$

Fig. 8: Influence of the two main threshold parameters on the binary mask.
Top row: height threshold $H_{\min}$ is swept from 10 to 1 dB while the neighbor-ratio threshold is fixed at $R_{\max} = 2.1$.
Bottom row: $R_{\max}$ is swept from 2.01 to 2.4 with $H_{\min} = 6\,\mathrm{dB}$ kept constant.

ing the other constant, demonstrating that the method remains effective over a broad range and can be fine-tuned for specific applications.

6 Conclusions and Future Work

We have presented a morphological spectrogram–denoising pipeline that treats the magnitude spectrogram as a grayscale image and exploits the max-tree representation to isolate signal components. By combining simple, interpretable

attributes (area, height, and a shape-based neighbor ratio) our method removes broadband noise while preserving the harmonic and transient structures corresponding to signal. Our approach relies solely on morphological operations, so it is lightweight, does not require training data, and runs in a reasonable amount of time on a CPU for small audio samples.

Our technique is explainable and interpretable, as it is based on the geometry of spectrograms and the properties of the max-tree.

One of the main limitations of our approach is that it relies on hand-tuned thresholds that were calibrated on a limited set of signals; their transferability to very different recording conditions (*e.g.* low-bit-rate speech, ultrasound) remains to be established. Moreover, our evaluation is mostly qualitative; the absence of perceptual metrics (*e.g.* PESQ, STOI) and large-scale statistical tests prevents a rigorous comparison with recent deep-learning baselines.

The future work will focus on benchmarking the method against classical and modern denoising techniques: it remains to be determined the soundness of our approach in comparison to the state of the art. One of the main goals will be to determine the robustness of the method to different types of noise and to the choice of parameters.

Beyond denoising, we believe that the proposed morphological framework can serve as a generic *pre-processing front-end* for a variety of downstream tasks—including onset detection, harmonic/percussive source separation, and acoustic-event classification. We therefore plan to release an open-source implementation together with a small benchmark corpus and reference masks, so as to foster reproducible research and encourage the integration of hierarchical morphology with data-driven (e.g. neural) approaches in the broader audio-signal-processing community.

References

1. Allen, J.: Short term spectral analysis, synthesis, and modification by discrete Fourier transform. IEEE Transactions on Acoustics, Speech, and Signal Processing **25**(3), 235–238 (1977)
2. Boll, S.: Suppression of acoustic noise in speech using spectral subtraction. IEEE Transactions on Acoustics, Speech, and Signal Processing **27**(2), 113–120 (1979)
3. Bosilj, P., Kijak, E., Lefèvre, S.: Beyond mser: Maximally stable regions using tree of shapes. In: British Machine Vision Conference (2015)
4. Boutry, N., Tochon, G.: Stability of the tree of shapes to additive noise. In: International Conference on Discrete Geometry and Mathematical Morphology. pp. 365–377. Springer (2021)
5. Cadore, J., Gallardo-Antolín, A., Peláez-Moreno, C.: Morphological Processing of Spectrograms for Speech Enhancement. In: Advances in Nonlinear Speech Processing. pp. 224–231. Springer, Berlin, Heidelberg (2011)
6. Cadore, J., Valverde-Albacete, F.J., Gallardo-Antolín, A., Peláez-Moreno, C.: Auditory-Inspired Morphological Processing of Speech Spectrograms: Applications in Automatic Speech Recognition and Speech Enhancement. Cognitive Computation **5**(4), 426–441 (2013)

7. Esteban, B., Tochon, G., Géraud, T.: Estimating the Noise Level Function with the Tree of Shapes and Non-parametric Statistics. In: Computer Analysis of Images and Patterns. pp. 377–388. Springer International Publishing (2019)
8. Gröchenig, K.: Foundations of Time-Frequency Analysis. Birkhäuser, Boston (2001)
9. Guo, Z., Hall, R.W.: Parallel thinning with two-subiteration algorithms. Communications of the ACM **32**(3), 359–373 (1989)
10. Hussein, W.: Spectrogram Enhancement By Edge Detection Approach Applied To Bioacoustics Calls Classification. Signal & Image Processing : An International Journal **3**(2), 1–20 (2012)
11. Koç, S.G., Aptoula, E., Bosilj, P., Damodaran, B.B., Dalla Mura, M., Lefevre, S.: A comparative noise robustness study of tree representations for attribute profile construction. In: 2017 25th Signal Processing and Communications Applications Conference (SIU). pp. 1–4. IEEE (2017)
12. Lam, L., Lee, S.W., Suen, C.: Thinning methodologies-a comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **14**(9), 869–885 (1992)
13. Lin, Z., Goubran, R.: Musical noise reduction in speech using two-dimensional spectrogram enhancement. In: The 2nd IEEE Internatioal Workshop on Haptic, Audio and Visual Environments and Their Applications, 2003. HAVE 2003. Proceedings. pp. 61–64 (2003)
14. de Oliveira, A.G., Ventura, T.M., Ganchev, T.D., de Figueiredo, J.M., Jahn, O., Marques, M.I., Schuchmann, K.L.: Bird acoustic activity detection based on morphological filtering of the spectrogram. Applied Acoustics **98**, 34–42 (2015)
15. Saldanha, J.C., R, S.O.: Reduction of noise for speech signal enhancement using Spectral Subtraction method. In: 2016 International Conference on Information Science (ICIS). pp. 44–47 (2016)
16. Salembier, P., Oliveras, A., Garrido, L.: Antiextensive connected operators for image and sequence processing. IEEE Transactions on Image Processing **7**(4), 555–570 (1998)
17. Sharan, R.V., Moir, T.J.: Noise robust audio surveillance using reduced spectrogram image feature and one-against-all SVM. Neurocomputing **158**, 90–99 (2015)
18. Uchiyama, R., Tanabe, N.: Real-Time Noise Suppression Using Harmonic/Percussive Separation with Morphological Operations for Hammering Test. In: 2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). pp. 1100–1106 (2023)