LOOKUP TABLE BASED AUDIO SPECTRAL TRANSFORMATION

Ryoho Kobayashi

College of Performing and Visual Arts
J.F. Oberlin University
Tokyo, Japan
kobayashi_r@obirin.ac.jp

ABSTRACT

We present a unified visual interface for flexible spectral audio manipulation based on editable lookup tables (LUTs). In the proposed approach, the audio spectrum is visualized as a two-dimensional color map of frequency versus amplitude, serving as an editable lookup table for modifying the sound. This single tool can replicate common audio effects such as equalization, pitch shifting, and spectral compression, while also enabling novel sound transformations through creative combinations of adjustments. By consolidating these capabilities into one visual platform, the system has the potential to streamline audio-editing workflows and encourage creative experimentation. The approach also supports real-time processing, providing immediate auditory feedback in an interactive graphical environment. Overall, this LUT-based method offers an accessible yet powerful framework for designing and applying a broad range of spectral audio effects through intuitive visual manipulation.

1. INTRODUCTION

Audio processing techniques have evolved significantly over the years, with frequency-domain methods playing a crucial role in applications such as music production, speech processing, and sound design. These methods enable manipulations of audio signals that are difficult or impossible to achieve in the time domain, offering greater flexibility and control over the resulting sound [1]. One prominent example is the phase vocoder [2], which analyzes an input audio signal using the shorttime Fourier transform (STFT), modifies the resulting timefrequency representation, and then resynthesizes the signal via the inverse STFT. The phase vocoder has been widely used for time-stretching and pitch-shifting and has been further refined over the years (for instance, by the Analysis/Synthesis team at IRCAM [3]). Despite such advancements, existing tools often rely on separate, dedicated interfaces for different effects, which limits flexibility and makes complex effect chains cumbersome to manage. Moreover, the interactions between multiple effects can be difficult to control and predict, requiring extensive experimentation and expertise.

In this paper, we propose a novel approach to audio processing that leverages lookup tables (LUTs)—a technique commonly used in image processing for color grading and enhancement [4]—to create a unified spectral processing framework. By mapping the frequency and amplitude of audio signals onto a 2D image and applying image-based LUT transformations, a wide range of audio effects can be achieved through a single, intuitive interface. This approach simplifies the workflow for many common audio processing tasks and enables complex, creative manipulations that would be difficult to achieve using conventional tools. The main contributions of this paper are as follows:

1. Unified LUT-Based Audio Effect Framework: We introduce a new method for audio processing that applies LUTs in the frequency domain, unifying various audio effects (equalization, dynamic range compression, pitch/frequency shifting, etc.) under a single interface.
2. Methodology and Implementation Details: We present a detailed description of the proposed method, including the frequency-domain signal representation, LUT design strategy, interpolation techniques for applying the LUT, and phase reconstruction processes.
3. Demonstration-Based Validation: We illustrate the method's versatility through qualitative demonstrations on diverse audio examples, showcasing its ability to achieve standard audio effects as well as unconventional spectral manipulations.
4. Discussion of Applications and Future Work: We discuss the potential of the LUT-based approach for creative sound design and outline directions for future development, such as improving the tool and exploring it as a shared platform for LUT-based audio processing.

The proposed LUT-based audio processing technique has broad potential applications, including music production, sound design, audio restoration, and interactive media. For example, in music production and sound design, our method can be used to create unique effects that are difficult to realize with traditional plug-ins. In audio restoration, carefully designed LUTs could repair or enhance damaged recordings by selectively boosting or attenuating certain spectral components. In interactive contexts like video games or virtual reality, the method can generate dynamic audio effects that adapt in real time to user input or game state.

2. RELATED WORK

Frequency-Domain Audio Processing: Frequency-domain approaches to audio processing have been actively researched for decades. One of the foundational techniques is the phase vocoder [2], introduced in the 1960s for time-scale modification and pitch-shifting of audio. Over the years, the phase vocoder has been improved in various ways to address its limitations. For instance, Röbel and Rodet [3] proposed a method for efficient spectral envelope estimation to improve pitch-shifting quality. Röbel [5] introduced a technique to better handle transients within the phase vocoder framework, reducing artifacts for percussive sounds. Laroche and Dolson [6] presented new phase-locking techniques to preserve phase coherence between related frequency components, yielding more natural-sounding transformations. These and other extensions have significantly enhanced the quality and flexibility of frequency-domain audio manipulation.

Despite these advances, frequency-domain tools often remain specialized and complex. Traditional digital audio workstations and effect plugins typically treat different effects (EQ, compressors, shifters, etc.) as separate modules with distinct control interfaces. Crafting a complex sound usually involves chaining multiple such modules, and the combined parameter space can be challenging to navigate—especially for novice users—due to complex interactions between effects.

Lookup Tables in Image and Audio Processing: In the field of image processing, LUTs are widely used for tasks like color grading and tone mapping [4]. A LUT provides a precomputed mapping from an input value (or set of values) to an output value, enabling efficient application of complex transformations. LUTs offer an intuitive way to modify images: for example, photographers can apply a color LUT to consistently adjust the colors and contrast of an image with a single operation. LUTs can be easily visualized, shared, and edited, which has contributed to their popularity in visual media.

By contrast, the potential of LUTs for audio processing has been relatively unexplored. There have been some efforts to simplify audio effect control through learned mappings or intelligent systems. For example, Rafii and Pardo [7] developed a system to control a reverberation effect using high-level perceptual descriptors, and Sheng and Fazekas [8] used a learned model to intuitively control a dynamic range compressor. These approaches use machine learning to map user-friendly parameters to effect settings, rather than explicit LUTs, and they are generally limited to specific effects. Other recent advances in audio processing have taken different directions: generative models and deep learning techniques have been applied to audio synthesis and transformation tasks (e.g., using adversarial networks to synthesize drum sounds with specified timbral features [9]) and differentiable signal processing frameworks have been created to integrate machine learning with traditional audio effects (e.g., the DDSP library for trainable audio effect modules [10]). While these methods greatly expand the possibilities for sound manipulation, they are fundamentally different from our LUT-based approach.

Motivation for a LUT-Based Approach: The above techniques highlight a trend toward more intuitive and powerful audio processing tools. However, there remains a gap for a unified approach that can handle many types of spectral transformations within one interactive interface. Inspired by the use of LUTs in image editing, our work aims to fill this gap by bringing LUT-based manipulation to the audio spectrogram domain. In contrast to previous work, our method applies LUTs directly to the frequency–amplitude representation of audio, providing a general framework rather than targeting a single effect or using black-box learning. This allows one visual paradigm to encompass a variety of effects. In the following sections, we describe the proposed LUT-based method in detail and demonstrate how it addresses the need for a flexible, user-friendly spectral editing tool.

3. PROPOSED METHOD

The proposed method consists of three main stages: (1) frequency-domain representation of the audio signal, (2) LUT design (creating the transformation as an image), and (3) LUT application (applying the LUT to modify the spectrogram and resynthesizing audio). Additionally, we have developed a user interface to facilitate real-time use of the method. In this section, we detail each stage.

3.1. Frequency-Domain Representation

The first stage involves converting the time-domain audio signal into a time-frequency representation suitable for image-based processing. We perform an STFT analysis of the input audio by dividing the signal into overlapping frames (using a window function) and computing the discrete Fourier transform for each frame. This produces a complex spectrogram $X(f, t)$, where f indexes frequency bin and t indexes time frame.

For use with LUTs, we focus on the magnitude of the spectrogram. We compute the log-magnitude spectrogram, $S(f, t) = \log |X(f, t)|$, which provides a perceptually relevant representation of amplitude across frequencies. This log-magnitude spectrogram is then interpreted as a grayscale image: the horizontal axis corresponds to time t, the vertical axis corresponds to frequency f, and the intensity of each pixel represents the amplitude (in log scale) at that time-frequency point. By using a log scale for amplitude, we ensure that a wide range of signal strengths is represented in a way that aligns better with human loudness perception.

3.2. LUT Design

The next stage is to design a lookup table that will transform the spectrogram image to achieve the desired audio effect. We represent the LUT as a 2D image of the same dimensionality as the spectrogram (frequency vs. amplitude). In this image, one axis corresponds to frequency and the other to the input amplitude (log-magnitude), and each pixel's value (brightness) encodes the new amplitude for that frequency-amplitude combination.

To make the LUT image intuitive to edit, we incorporate a color-coding for frequency. In our design, the LUT image uses hue to represent frequency: low frequencies are mapped to red hues and high frequencies to blue, with intermediate frequencies represented by colors like green and yellow. This scheme is inspired by the well-known "rainbow" colormap often used in scientific visualization [11], which effectively conveys an ordering of values through visible color transitions. The brightness (value) in the LUT image represents the output gain applied to a given input amplitude at that frequency. Figure 1 shows an example of a reference LUT image using this frequency–color mapping (with no processing applied yet, so brightness is uniform). By visualizing frequency as color, the user can easily identify and target certain frequency ranges when editing the LUT, while the brightness indicates how those frequencies' amplitudes will be scaled.

When creating a LUT for a particular effect, the user effectively "draws" the desired spectral mapping into this image. For instance, a LUT that darkens (lowers brightness of) a certain color band will attenuate those frequencies, whereas a LUT that brightens specific regions will boost those frequencies or amplitudes. Because the LUT is two-dimensional, it can capture context-dependent transformations—for example, making the gain

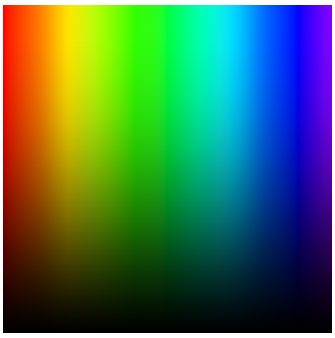

Figure 1: *Reference lookup table (unprocessed)*

at a frequency depend on the current amplitude at that frequency (allowing dynamic, amplitude-dependent effects).

3.3. LUT Application

Once the LUT image is designed, it is applied to the spectrogram of the input signal to produce a modified spectrogram, which is then converted back to audio. The LUT application process works as follows. First, we normalize the frequency and amplitude coordinates of the spectrogram to a 0–1 range to correspond to the normalized axes of the LUT image. Then, for each time-frequency bin (f, t) in the spectrogram, we use the LUT image as a lookup table to find a scaling factor for that bin's amplitude.

In practice, the frequency f (normalized between 0 and 1 across the audio band) corresponds to a specific horizontal position (hue) in the LUT image, and the original amplitude (in log scale, also normalized 0–1 over the amplitude range) corresponds to a vertical position in the LUT image. We retrieve the LUT image's brightness at that (frequency, amplitude) coordinate to get the new amplitude. Since the coordinate may not fall exactly on a pixel center, we use bilinear interpolation between the four nearest pixels in the LUT image to compute an accurate output value. This yields the LUT-transformed magnitude for each time-frequency point.

After applying the LUT mapping to every pixel of the input spectrogram image, we obtain a modified magnitude spectrogram $S'(f, t)$. To resynthesize the audio, we combine the modified magnitudes with the original phase information from the input spectrogram $X(f, t)$. That is, we form a new complex spectrogram $X'(f, t)$ where $|X'(f, t)| = \exp(S'(f, t))$ (inverting the log) and $\angle X'(f, t) = \angle X(f, t)$ (phase unchanged). Finally, we perform the inverse STFT on $X'(f, t)$ to reconstruct the time-domain audio signal corresponding to the processed spectrogram. In summary, the frequency content of the audio is unchanged in position, but the amplitude of each frequency-time component is altered according to the LUT's prescribed mapping.

3.4. User Interface

To make the LUT-based processing approach user-friendly, we developed a simple, interactive graphical user interface. The interface has two main components: a LUT editor and a spectrogram display.

3.5. LUT Editor

The LUT editor allows users to create and modify the 2D LUT image directly. It provides basic image editing tools such as brushes (to paint regions of the LUT with a chosen brightness or color), gradient fill tools (to smoothly interpolate brightness across a region), and a color picker for selecting hue ranges. The editor also supports loading pre-designed LUT images from file and saving the user's current LUT design for later use or sharing. These features enable users to experiment with different LUT shapes and to reuse effective LUTs for similar tasks or creative outcomes.

3.6. Spectrogram Display

The spectrogram display shows two images side by side in real time: the original log-magnitude spectrogram of the input audio, and the spectrogram after applying the current LUT transformation. As the user edits the LUT, the modified spectrogram updates accordingly, giving immediate visual feedback on how the LUT affects the frequency content of the audio. Users can zoom in and pan across the spectrogram to inspect specific time-frequency regions in detail. The interface also allows adjustment of the colormap or dynamic range of the display (for example, to highlight lower-level details or limit the range of interest), so that the user can better see how subtle changes in the LUT impact the spectrogram. Audio playback controls are integrated to allow the user to audition the original and processed audio easily.

Through this interface, the proposed method becomes an interactive spectral processing tool: the user can intuitively "draw" on the spectrogram via the LUT and immediately hear and see the results, enabling an iterative and creative workflow.

4. DEMONSTRATIONS

To evaluate the proposed LUT-based audio processing method, we conducted a series of experiments using a variety of audio signals and LUT designs. The goals of these experiments were to demonstrate that our method can reproduce common audio effects within the LUT framework and also enable novel manipulations that are difficult to achieve with conventional tools.

4.1. Demonstration Setup

We implemented the method in C++ using the JUCE framework for real-time audio processing and the GUI. The STFT analysis and inverse STFT synthesis used a typical overlap-add method with a Hanning window. All experiments were run on mono audio signals sampled at 44.1 kHz. We tested the system on diverse audio input examples, including spoken voice, instrumental music clips, and environmental sounds, spanning a range of durations.

Unless otherwise noted, the LUT images in our tests were designed with a resolution of 256x256 pixels. This resolution provides a balance between fine-grained control (being able to distinguish subtle differences in frequency and amplitude) and computational efficiency. At 256x256, the LUT application via interpolation is fast enough for real-time use on a modern CPU while still capturing detailed transformation curves. We note that higher resolutions could improve precision at the cost of performance, as discussed later.

4.2. Reconstructing Common Effects

The first set of experiments illustrates how the LUT-based method can achieve common audio effects such as volume control, compression, equalization, filtering, frequency shifting, and pitch shifting, all through appropriate design of the LUT image. For each effect, we designed a LUT and applied it to various test signals, then compared the result qualitatively to the expected outcome of the corresponding conventional audio effect.

4.2.1. Volume Adjustment

The simplest effect is a uniform gain change. To apply, for example, +6 dB gain to the entire signal, the user increases the brightness of the whole LUT image uniformly. Conversely, for a cut in gain, the LUT is made darker overall. The result is that all frequencies at all amplitudes are scaled equally. This corresponds exactly to a master volume fader. Figure 1 shows the LUT for a volume reduction (overall darker image). As expected, the output audio is simply quieter (or louder) with no change in tone.

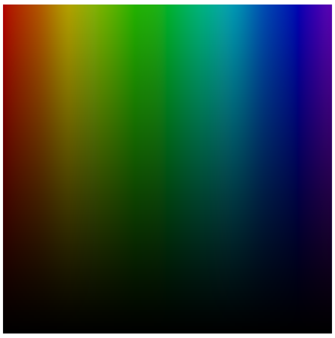

Figure 2: *LUT representation of a volume adjustment (overall brightness uniformly decreased).*

4.2.2. Dynamic Range Compression

A static compressor effect can be achieved by using the LUT to reduce gain at high input amplitudes. In the LUT image, this means the upper portion (representing loud input magnitudes) is made darker relative to the lower portion. We designed a curve in the LUT brightness that starts at unity gain for low-level inputs and gradually reduces gain for high-level inputs (above a "threshold"). Visually, the LUT's top region is dimmed, implementing a soft knee compressor transfer function across all frequencies. Figure 2 illustrates a LUT configured as a compressor: the brightest areas (top of the image) have been attenuated. The processed audio's dynamic range is reduced – loud parts are brought down in level – mimicking a broad-band compressor. (If the opposite adjustment is drawn – boosting the top region – the LUT would act as an expander.)

4.2.3. Equalization (EQ)

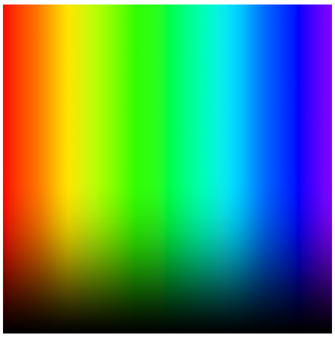

Figure 3: *LUT for dynamic range compression (upper-amplitude regions are darkened to attenuate loud inputs).*

To boost or cut specific frequency bands, the user modifies brightness in localized vertical strips of the LUT image. For example, to create a bass boost, one can brighten the lower-frequency (left side) region of the LUT across all amplitudes. To cut a narrow band (notch filter), one draws a dark line vertically at the corresponding frequency (hue) position. We verified that painting these patterns yields the expected filtering effect on the audio. Figure 3 shows an example where a low-frequency band in the LUT is brightened, resulting in an EQ that raises the low frequencies. This approach is analogous to drawing an EQ curve: the benefit of the LUT method is that the shape can be arbitrary or even amplitude-dependent if desired (unlike a traditional EQ which is static irrespective of input level).

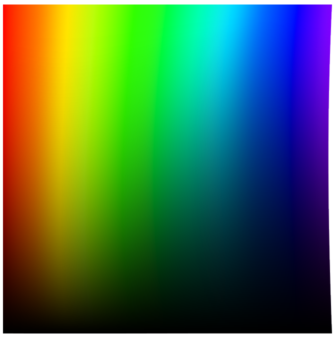

Figure 4: *LUT implementing a band-specific gain (boosting the brightness of a low-frequency region, left side of image).*

4.2.4. Filtering

A variety of filter shapes (low-pass, high-pass, shelves, etc.) can be implemented by painting regions of the LUT fully dark or bright. For instance, a simple low-pass filter is achieved by darkening the LUT entirely for frequencies above a chosen cutoff frequency (i.e., making the right side of the image black, so high frequencies always get zero gain). Figure 4 shows a LUT configured as a low-pass filter: the high-frequency region (right side) is darkened. The processed audio in this case loses all content above the cutoff, exactly as expected from a low-pass filter. More complex filters, like a band-stop, would involve darkening a band in the middle while leaving low and high frequencies bright. The LUT approach essentially lets the user "draw" any frequency response shape.

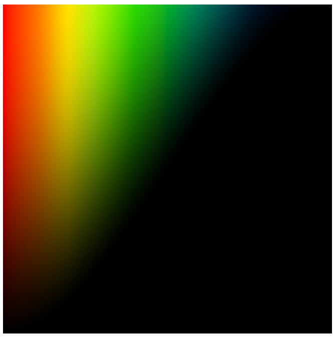

Figure 5: *LUT implementing a low-pass filter (high-frequency region on the right is darkened, cutting those frequencies).*

4.2.5. Frequency Shifting

Frequency shifting (as opposed to pitch shifting) means adding a constant offset to all frequency components (e.g., shifting the spectrum down by 500 Hz). With an LUT, we accomplish this by remapping hue positions horizontally. In practice, we prepared an alternate version of the LUT where the hue-to-frequency mapping is shifted: for example, each input frequency bin f is mapped to a brightness taken from a slightly lower frequency region of the image. This can be done by “dragging” the LUT image content to the left or right. Figure 5 illustrates a LUT that has been shifted to realize a downward frequency shift (the entire pattern is shifted leftward). Applied to audio, this results in an effect akin to single-sideband modulation – all spectral components move down in frequency by the same amount, producing an inharmonic but interestingly warped version of the sound. Unlike pitch shifting, frequency shifting does not preserve harmonic ratios, so the effect can sound like a bizarre detuning; our LUT implementation provides a straightforward way to experiment with such transformations.

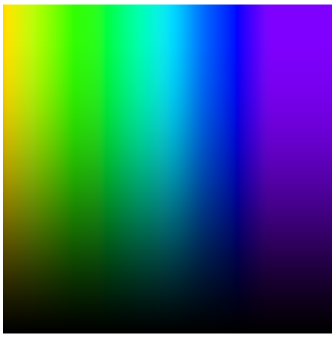

Figure 6: *LUT implementing a downward frequency shift (the hue pattern is shifted to the left, so each input frequency maps to a lower output frequency).*

4.2.6. Pitch Shifting

Pitch shifting by a ratio (e.g., up one octave) can be achieved by stretching or contracting the LUT horizontally. We treat the left edge of the image (0 Hz) as an anchor and stretch the hue mapping such that a given input frequency corresponds to a different (scaled) output frequency. For example, for an upward pitch shift, the LUT image is non-linearly stretched so that high-frequency content is compressed towards the right edge. Figure 6 shows a LUT that implements a pitch increase: the hues are stretched such that each input frequency bin now maps to a higher frequency region (except 0 Hz which stays in place). This yields a classic pitch shift effect (using the phase vocoder backbone for time scaling). In our tests, the LUT-based pitch shifting produced the expected change in perceived pitch. It behaves similarly to a standard phase vocoder pitch shifter, with slight phase smearing on transients if the shift is large, as expected.

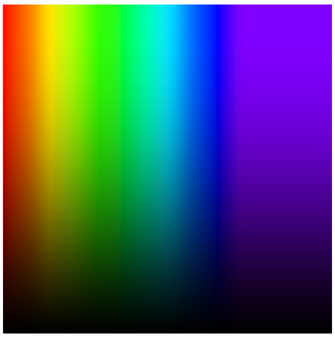

Figure 7: *LUT implementing a pitch raise (the LUT image is stretched horizontally, so frequencies are expanded towards the high end, effectively doubling frequency values for an octave-up shift).*

4.3. Creative and Complex Manipulations

Beyond standard effects, the LUT approach allows combinations and novel mappings that can be cumbersome to realise with conventional tools, although a formal comparison remains future work. We present a few creative examples that exploit the 2D nature of the LUT.

4.3.1. Amplitude-Dependent Frequency Shifting

One interesting possibility is to make a frequency shift that happens only at certain amplitudes. For instance, we designed a LUT that performs an upward pitch shift only when the input is loud. This was done by stretching the hue mapping (like pitch shifting) only in the upper part of the image (above a certain brightness threshold), while leaving the lower part unaltered. The effect on a sustained synth chord was that when the chord was played softly, it remained at the original pitch, but when played with force, the notes shifted upward in pitch. Essentially, the loud portions of the sound were mapped to higher frequencies, creating an amplitude-driven pitch bend. Figure 7 shows such an amplitude-dependent pitch shift LUT: the horizontal stretching is prominent in the bright upper regions of the image. Sonically, this produced a dynamic effect where the sound’s pitch warbled in response to amplitude changes – e.g., a piano note would start in tune when soft and then suddenly jump up an overtone when hit hard. This kind of effect has no direct equivalent in traditional processors, illustrating the unique mappings possible with our approach.

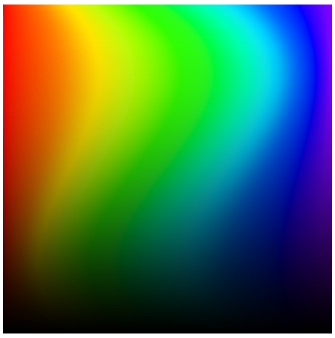

Figure 8: *LUT implementing an amplitude-dependent pitch shift (the amount of horizontal stretch increases for brighter parts of the image, causing loud inputs to be pitch-shifted).*

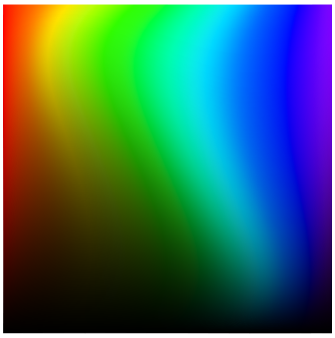

Figure 9: *complex LUT combining multiple effects in one image (e.g., simultaneous bass cut, treble boost, and mid compression as described in the text).*

4.3.2. Frequency-Dependent Dynamic Range

As a more selective form of compression, we used the LUT to compress only a specific frequency band when it gets loud. We painted a dark horizontal blob in the LUT at a particular frequency (around 5 kHz) in the upper amplitude region. The result is a frequency-selective de-esser: loud content around 5 kHz is attenuated, but the rest of the spectrum is unaffected unless it also becomes loud. We tested this on a music mix where the vocal sibilance was strong around 5 kHz. With our LUT, whenever that band's amplitude crossed the threshold (became bright in the LUT), its gain was reduced significantly, which in turn slightly ducked the overall signal energy (since that band carried a lot of the mix's power). The behavior was akin to a multi-band compressor set to only trigger on that band. Achieving this with traditional tools would require a tuned sidechain detector or a multiband compressor; in our system, it was created simply by "drawing" a dark shape at the desired frequency and amplitude region. Outside the targeted band and amplitude, the LUT remained at unity (no change), so the dynamics and frequency content were preserved elsewhere. This example demonstrated how one can intuitively design custom dynamic EQ or de-essing behavior visually.

4.3.3. Custom Spectral Morphing

We constructed a more complex LUT that combines multiple effects simultaneously, to demonstrate the all-in-one capability of the approach. One such LUT included a low-cut filter for bass (dark at low-frequency, low-amplitude region), a high-frequency shelf boost (bright region at high-frequency, mid-amp), and a gentle compression in the midrange (slightly dimmer in the top part of mid-frequency region). Applying this single LUT to a drum loop yielded a combined effect: it tightened the sub-bass (removing rumble), added brightness to the cymbals, and tamed the snare transients – all in one processing step. Designing this by ear took only a few iterations using the GUI, and it was intuitive: we could literally see which parts of the spectrum were active for each drum hit and adjust the LUT image accordingly. The final LUT image looked complex (various colored regions and gradients), embodying a unique combination of EQ and dynamics that would normally require a chain of several plugins. This showcases how a sound designer can craft a signature spectral effect in the LUT domain that is hard to replicate otherwise. Figure 8 depicts an example of such a complex LUT combining multiple audio effects.

4.3.4. Generative/Procedural Effects

Since a LUT is essentially an image, it can be created or modified algorithmically to explore unusual sound transformations. We experimented with populating the LUT with procedural patterns (random noise, geometric shapes) to see their effect on audio. In one trial, we filled the LUT with a pseudo-random checkerboard pattern of bright and dark squares. When applied to a sustained ambient sound, this LUT imparted a granular, shimmering texture: certain frequency bins would randomly drop in and out as the sound decayed, due to the alternating gain pattern. In another trial, we drew a diagonal stripe across the LUT (from top-left to bottom-right). This caused a frequency-dependent gating effect: as a note's amplitude fell below the diagonal stripe (threshold), frequencies above a certain proportion were cut off. The result was a kind of frequency-dynamic tremolo, where different frequency bands of a decaying sound were being gated at different times, creating an evolving timbre. These are exotic effects that go beyond typical studio needs, but they hint at the exploratory creative space the LUT method opens up. Sound designers could use algorithmic images (fractals, noise, etc.) to "texture" a sound in ways that have few analogues in traditional processing.

4.4. Sound Quality and Artifacts

In our tests, the audio quality of the processed results was generally high, and the limitations were mostly those typical of STFT-based processing. When making extreme spectral modifications (such as very sharp filters or large pitch shifts), slight smearing or phasiness can occur – as is well-known in phase vocoder techniques. Incorporating advanced phase-processing techniques (e.g., transient preservation algorithms, phase consistency constraints, or simply higher FFT overlap) can mitigate these issues; these are matters of engineering refinement rather than fundamental limitations of the LUT concept. One artifact specific to our LUT approach arises with heavy amplitude-dependent processing: because the STFT operates on finite frames, a sudden gain reduction in the LUT for loud content can slightly dull the onset of transients (the reduction is applied uniformly over a frame, e.g., 20–40 ms, which may smooth a sharp attack). Using shorter hops (more overlap) or windowing can alleviate this by responding faster to changes. Another consideration is phase for frequencies that were initially silent: if the LUT introduces energy at a frequency bin that had near-zero input energy (for example, an extreme frequency warp painting energy into an empty band), the output phase for that bin is essentially arbitrary (often residual noise), which can cause a hollow or phasy sound if overdone. In practice, this was rarely an issue unless making radical spectral warping. Solutions could include injecting a very small noise floor to provide stable phase in “empty” bins or applying phase synthesis techniques, but we did not find it necessary for moderate use. In summary, the creative potential of the LUT approach is vast, but, as with any STFT-based process, care must be taken for extreme transformations to avoid known artifacts. Reasonable settings (adequate overlap, not pushing filters beyond spectral resolution limits) yielded clean results in our experiments.

5. DISCUSSION

The demonstrations above suggest that the proposed LUT-based audio processing method is both flexible and creative, capable of reproducing standard effects and enabling unconventional sound manipulations. By representing an audio spectrogram as an image and applying visually designed transformations, our approach offers a novel, intuitive way to manipulate sound.

5.1. Advantages

One key advantage of the method is its ability to achieve complex and expressive sound effects using a single unified framework. Through appropriate LUT designs, a virtually unlimited variety of spectral transformations can be realized—from very subtle tonal adjustments to radical modifications that fundamentally alter the sound. This flexibility is especially valuable in creative applications like music production and sound design, where unique and personalized effects are often sought. Moreover, the framework encourages experimentation: since all effects are controlled via the LUT image, users can try unusual combinations or patterns without needing a deep technical understanding of the underlying DSP for each effect.

Another advantage is the method’s computational efficiency and potential for real-time performance. The core processing steps (STFT, LUT application, and inverse STFT) can be implemented with fast FFT algorithms and pixel-wise operations, which modern processors (and even GPUs) handle very well. In our implementation, we achieved real-time processing for moderate frame sizes and a 256x256 LUT on a standard PC. This low latency capability means the technique can be used in live settings or interactive systems where immediate audio feedback is required, such as live electronic music performances or adaptive audio systems in games.

5.2. Limitations

Despite its strengths, the LUT-based approach has some limitations. One limitation is related to resolution and precision. The LUT image imposes a fixed resolution in frequency and amplitude mapping. If the resolution is too low, the transformations might be coarse, missing fine spectral details or introducing quantization artifacts (e.g., audible stepping when a narrow band is boosted). Increasing the LUT resolution improves precision but consumes more memory and processing power. In practice, one must balance the need for detail with the available computational resources. For example, a 1024x1024 LUT might capture very fine spectral nuances but could be slow to apply in real time. Determining the optimal LUT size for a given application (or implementing multi-resolution strategies) is an area for further optimization.

Another challenge is the possibility of artifacts. Because our method manipulates spectral magnitudes independently and then reuses the original phases, certain transformations can cause inconsistencies between magnitude and phase. This may result in slight distortions or a “phasiness” in the output audio. For instance, if the LUT introduces very sharp changes in amplitude for adjacent frequency bins, the phase relationships might no longer be ideal, potentially causing transient smearing or interference patterns. We mitigate some issues by using overlap-add in the STFT, but artifacts can still occur for extreme LUT patterns or very fast-changing audio. Developing phase-aware processing (where the phase is also adjusted in tandem with magnitude changes) or applying post-processing (like mild reverb or smoothing) could alleviate these artifacts and is worth exploring in future work.

The interpolation method used during LUT application is another factor. We chose bilinear interpolation for its simplicity and efficiency. While it provides a smooth mapping in most cases, bilinear interpolation might not capture very sharp transitions accurately, and it could introduce minor blurring of the LUT pattern. More advanced interpolation schemes (e.g., bicubic or Lanczos/sinc interpolation) could yield more accurate adherence to the desired LUT shape, at the cost of additional computation. In scenarios where the LUT contains critical fine features (like a very narrow notch), a higher-order interpolation might be justified to preserve that detail in the audio output.

Finally, the LUT design process itself can be demanding for users when targeting complex outcomes. Designing a LUT by hand to do exactly what you want (especially for a complex multi-effect) may involve some trial and error. While the visual nature of the tool is a big improvement in intuitiveness, it still requires understanding how image manipulations translate to spectral changes. Some users might find it challenging to draw the right curve for a desired effect. This suggests an opportunity to incorporate assistive tools: for example, automatically generating a LUT from example “before and after” audio, or providing template LUTs for common effects that users can then tweak. Integrating such intelligent LUT design aids (potentially using machine learning to suggest LUT modifications) could greatly enhance the usability of the system for those who are less technically inclined or who want to achieve a specific target sound more directly.

Comparison to Traditional Methods: It is informative to consider where the LUT-based approach stands relative to traditional audio effect design. Conventional plugins are usually engineered for one task (e.g., a compressor, an EQ, a shifter) and finely tuned for audio quality in that domain. Our method is more general but might not yet match the polished sound of specialized tools in every case—for example, a high-end analog-modelled compressor plugin may produce a more transparent or musically pleasing compression on a complex mix than a quickly drawn LUT compressor. However, the strength of our approach lies in its versatility and the ease of crafting new hybrid effects that have no direct counterpart among standard tools. As such, we see LUT-based processing not as a replacement for all traditional effects, but as a complement that opens new sound design avenues. In scenarios where a well-known effect is needed with optimal quality, a dedicated tool might be preferred; but when an innovative or unusual effect is desired, our framework shines by allowing freeform experimentation.

6. CONCLUSION

We have presented a novel method for audio processing based on the use of lookup tables (LUTs) applied to an audio spectrogram. By treating the spectrogram as an image and manipulating it with color-coded LUTs, our approach provides a flexible and intuitive framework for spectral sound manipulation. Through a series of experiments, we demonstrated that this method can successfully implement a wide range of audio effects—including volume control, equalization, filtering, compression, frequency shifting, and pitch shifting—using a unified representation. Moreover, we showed that the same framework facilitates complex and creative manipulations (such as adaptive spectral effects and combined multi-effects) that would be difficult to achieve with conventional audio processing chains.

The proposed LUT-based approach offers several advantages: it unifies many effects in a single paradigm, encourages creative exploration by making spectral changes visual, and runs with computational efficiency suitable for real-time use. Users ranging from professional sound designers to hobbyists could utilize this tool to explore sound transformations in a more intuitive way, "painting" their effects and immediately hearing the results. The ability to directly shape the time-frequency content of audio unlocks new possibilities for artistic expression and sound experimentation.

At the same time, we acknowledge limitations that present opportunities for future work. Ensuring high precision without sacrificing performance will require careful choice of LUT resolution or adaptive schemes. Reducing artifacts, possibly by incorporating phase processing or smarter interpolation, is important for maintaining audio fidelity under extreme transformations. Improving the user experience for designing LUTs—potentially by adding intelligent assistants that suggest LUT modifications or by using machine learning to derive LUTs from example outputs—could broaden the method's accessibility and appeal. For instance, a future system might allow a user to specify "make my audio sound like this reference track," and an algorithm could generate a LUT to approximate that spectral difference.

We also plan to explore integrating LUT-based spectral processing with other audio effect paradigms. One idea is to combine our method with time-domain effects (like convolutional reverb or time-stretching) to build an even more comprehensive audio transformation environment. Another exciting direction is the development of a community platform for sharing LUT designs. Since a LUT image can be saved and shared easily, users could swap their favorite "presets" much like they exchange impulse responses or plugin presets today. This would help accumulate a library of interesting spectral effects and spur further innovation as people build on each other's ideas.

7. REFERENCES

[1] U. Zòlzer, Ed., *DAFX: Digital Audio Effects, 2nd ed*, Wiley, 2011.

[2] J. L. Flanagan and R. M. Golden, "Phase vocoder," *Bell Syst. Tech. J.*, vol. 45, no. 9, pp. 1493–1509, 1966.

[3] A. Röbel and X. Rodet, "Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation," in *Proc. 8th Int. Conf. on Digital Audio Effects (DAFx)*, 2005, pp. 203–206.

[4] K. Myszkowski, Ed., *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting, 2nd ed.*, Morgan Kaufmann, 2010.

[5] A. Röbel, "A new approach to transient processing in the phase vocoder," in *Proc. 6th Int. Conf. on Digital Audio Effects (DAFx)*, 2003, pp. 344–349.

[6] J. Laroche and M. Dolson, "New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 1999, pp. 91–94.

[7] Z. Rafii and B. Pardo, "Learning to control a reverberator using subjective perceptual descriptors," in *Proc. 14th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2013, pp. 285–290.

[8] D. Sheng and G. Fazekas, "A feature learning siamese model for intelligent control of the dynamic range compressor," in *Proc. 20th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2019, pp. 816–823.

[9] S. Lattner, J. Nistal, and G. Richard, "Drumgan: Synthesis of drum sounds with timbral feature conditioning using generative adversarial networks," in *Proc. 20th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2020, pp. 221–228.

[10] C. Gu, J. Engel, L. H. Hantrakul, and A. Roberts, "Ddsp: Differentiable digital signal processing," in *Proc. 8th Int. Conf. Learning Representations (ICLR)*, 2020.

[11] A. Light and P. J. Bartlein, "The end of the rainbow? color schemes for improved data graphics," *Eos Trans. AGU*, vol. 85, no. 40, pp. 385–391, 2004.