VibeFM: Visual Exploration of FM Synthesis

Simeon Rau
University of Stuttgart
Stuttgart, Germany
simeon.rau@visus.uni-stuttgart.de

Finn Tobien
AudioLabs, FAU Nürnberg-Erlangen
Erlangen, Germany

Michael Sedlmair
University of Stuttgart
Stuttgart, Germany
michael.sedlmair@visus.uni-stuttgart.de

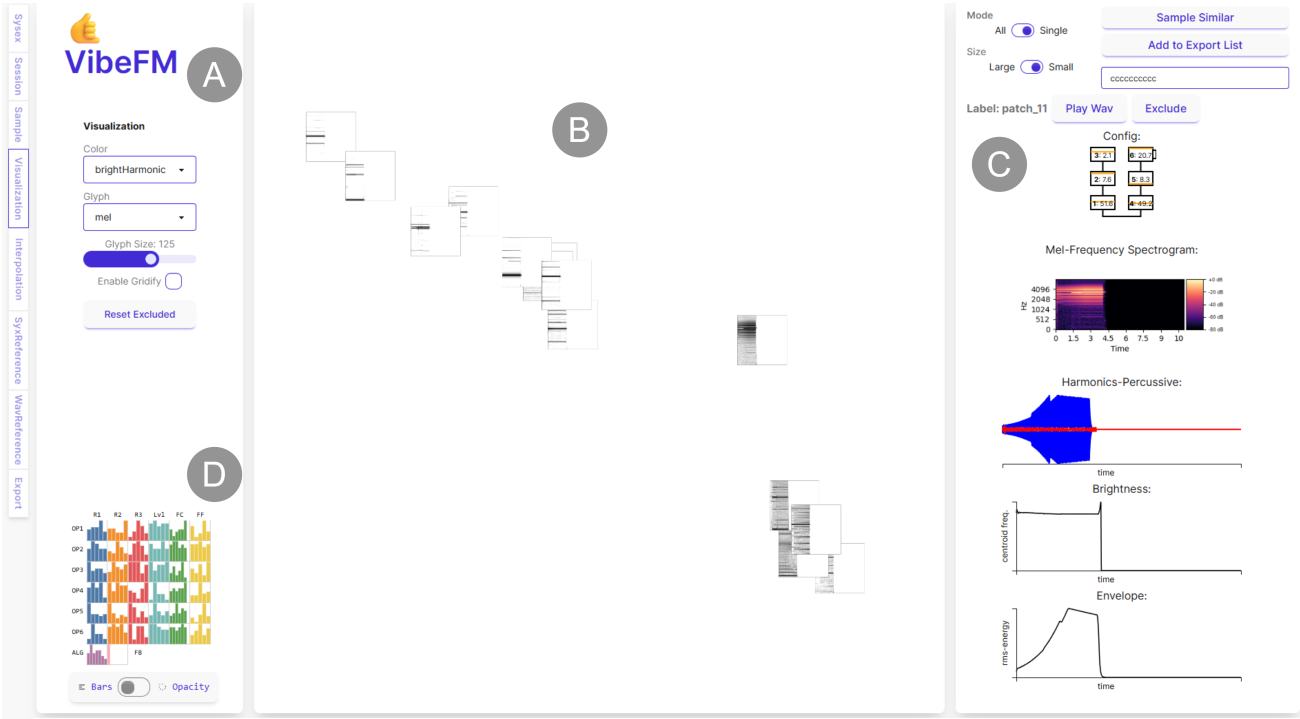


Figure 1: VibeFM's interface to explore FM synthesis configurations: A) Menu for sampling, exports, and layout & encoding options. B) Overview visualization with glyphs placed by similarity; here, the mel-spectrogram glyph. C) Provides interaction with the selected sample and details on meta information (e.g., algorithm) and sound characteristics (e.g., envelope shape). D) Overview of parameter variety.

Abstract

We propose VibeFM, an interactive visual approach for sampling, exploring, and identifying novel synthesizer sounds using frequency modulation (FM). Our approach is based on two steps: First, we systematically sample parameter combinations to create a set of sound samples. We then provide a visual overview that shows each sample as a small visualization and orders them spatially by similarity. Our premise is that, by visualizing key audio features, one can quickly parse a large collection of samples without listening to all of them. Furthermore, the approach supports serendipitous discovery of new sounds during the creative composition process. Using VibeFM, we created a soundtrack faster than our usual estimated time and also discovered new and interesting sounds. In semi-structured interviews, one professional composer and two experienced musicians found our approach to be a convenient complement to synthesizer programming.

NIME '26, June 23–26, 2026, London, UK

Keywords

Visual Interface, Sound Design, FM-Synthesis, Visual Analytics, Exploration

1 Introduction

During the instrumentation phase of a music composer's workflow, they seek suitable instruments and like to experiment with new sounds [53]. A modern approach widely used by musicians and composers of all skill levels is using synthesizers and artificial sounds. One way to synthesize sounds [47] is frequency modulation (FM) synthesis [7], a modulation technique with multiple oscillators to shape sound, iconic for its futuristic, electronic sounds. A famous FM synthesizer is the Yamaha DX7, which consists of six operators structured by an algorithm, for a total of 144 parameters. For effective use, composers must find a suitable configuration from a vast space of possible configurations [10].

Currently, composers use fixed presets or manually tune hundreds of synthesizer parameters. Presets are usable but offer limited flexibility due to subjective needs [53]. In Contrast, manual tuning is flexible but cumbersome [24] even for experts and necessitates considerable expertise [21]. Our goal is to provide a new approach that balances this tradeoff.

We propose a user-centered, visual approach called VibeFM (Visualization and Interactive Bandwidth Exploration for FM synthesis) to search for sounds and support creativity [52]. For this, we systematically sample from the large space of configurations[1]. To process them, we provide a visual overview of the samples, similar to visual analytics [22], specifically, visual parameter space analysis [48]. Visual overviews support systematic exploration [10] of sampled spaces, and we allow interpolation between samples to improve navigation compared to previous trial-and-error methods. At first, composers might not know where to start exploring, so we provide visual representations to gain insights into sound characteristics. By using the concept of auditory imaging [20], i.e., imagining sound from visual representations, similar to notational audiation for notation [4], we visualize key sound characteristics, such as the envelope or brightness. Although visualization cannot replace listening to sounds, showing sound features [42] allows composers to spot interesting regions to investigate.

We conducted a case study, in which we created a soundtrack to illustrate a creative workflow and serendipity in exploring sounds. Compared to our usual workflow, we felt overall faster and were inspired to use sound variations for stereo effects. In semi-structured interviews, we received feedback from experts, who generally liked our approach and saw potential for a useful complement to programming synthesizers.

In summary, we contribute 1) the design of interactive visualizations to explore and select FM synthesizer configurations, 2) an evaluation through a case study and expert interviews, and 3) the web-based prototype source code (VibeFM[2]).

2 Background & Related Work

We cover related work on sound synthesis, synthesizer interfaces, and visual analytics in related fields.

Frequency Modulation Synthesis: In frequency modulation (FM) synthesis [7], the frequency of a carrier signal is varied by another frequency or frequency rate (modulator), as shown in Figure 2. Depending on different modulation levels (modulation index), this process generates additional spectral components (sidebands) around the carrier frequency. By adjusting these parameters, the spectral distribution and thus perceived timbre of a sound can be controlled [38]. Combining multiple modulation operations (algorithm) creates more complex sounds, where modulators can themselves act as carrier signals. As this topic extends beyond this work's scope, we refer to existing literature for further information [7, 38].

Sound Synthesis: For sound synthesis [47], in particular, FM synthesis, many commercial emulators (NI FM8, Aparillo, Bleass Omega) for real synthesizers like Yamaha GS-1, DX21, and Modx+ exist. For our work, we chose Dexed[3], which emulates the Yamaha DX7 and provides all basic functionalities. To convey parameter complexity, others use sound matching [43] to recreate existing sounds or utilize deep learning techniques [14, 26] to create sound directly. Because these are hard to control and focus on recreation, composers have difficulties exploring these models without appropriate overviews. Instead of directly creating a sound, other works aim to reconstruct configurations for a given audio to solve parameter complexity [33, 57, 58]. But to find sounds that meet a composer's needs, one has to provide

[1] 144 parameters, each around 100 values, equals around 10^{200} possible combinations
[2] https://github.com/snrau/VibeFM
[3] asb2m10.github.io/dexed/

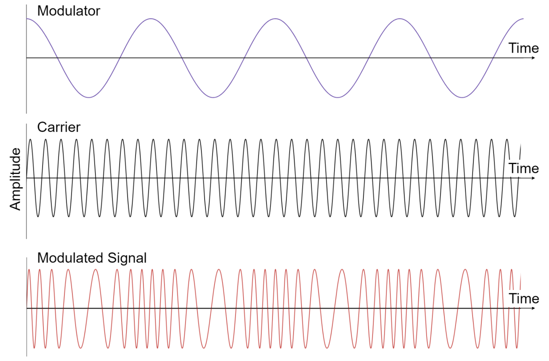


Figure 2: A basic example of frequency modulation.

a similar audio file already. Still, this might not fully satisfy the composer, whereas we support creativity by focusing on providing overviews of multiple similar configurations to explore. Caspe et al. [6] propose a differentiable deep learning approach that learns to use a portion of parameters, ultimately reducing the number of controllable parameters. While this supports finding common configurations, it introduces learned biases and hinders experts in exploring sounds freely, trading flexibility for usability. Conversely, we allow full control and provide a visual overview of alternative configurations to reduce biases.

Interfaces: Instead of automatic ways, research proposed user interfaces to make sound design more controllable. They often use an iterative approach, where users sample sounds, select suitable ones, and continue from there [9, 21]. Their focus lies on sampling, through chats [3] and machine learning [51], which might introduce bias and miss the desired output. Instead, Luke et al. [29] let users vote for 16 candidates, which are used to sample the next round. Due to missing visual overviews, users have limited options to choose from and might miss out on other options, which our work addresses. Easthope's work [12] uses a lightweight synthesizer to interpolate between sounds with a web-based interface, but focuses on direct sound interpolation rather than configurations. Further, Vaillant et al. [28] use visualization to display preset and found that interpolation between configurations provides better results than changing parameters. Similarly, we allow composers to interpolate between samples for interim sounds, ultimately providing more control and flexibility. AudioStellar [17] and Cosmos[4] use visualization for a corpus of sound presets but lack sampling and exploring novel sounds. While selecting samples based on their visual representations is hard, we focus on providing information about sound characteristics directly in the overview.

Visual Analytics: To the best of our knowledge, no prior research addressed visual analytics for FM synthesis, but used these concepts in other music domains. For example, Visualization is used to analyze song structures [19] or harmonic contexts in composition corpora [34], often including dimensionality reduction [44]. Visual analytics is used for exploring song collections [18, 36], AI-generated sample collection [41], or organizing music collections with audio feature extraction [35, 36]. Rau et al. [40] interactively sample melodies using AI and provide a visual overview for exploration and selection, a highly subjective task [5]. Our approach has similar goals for a different scenario, but focuses on different sampling strategies, audio feature extraction, and representation.

[4] www.waves.com/plugins/cosmos-sample-finder

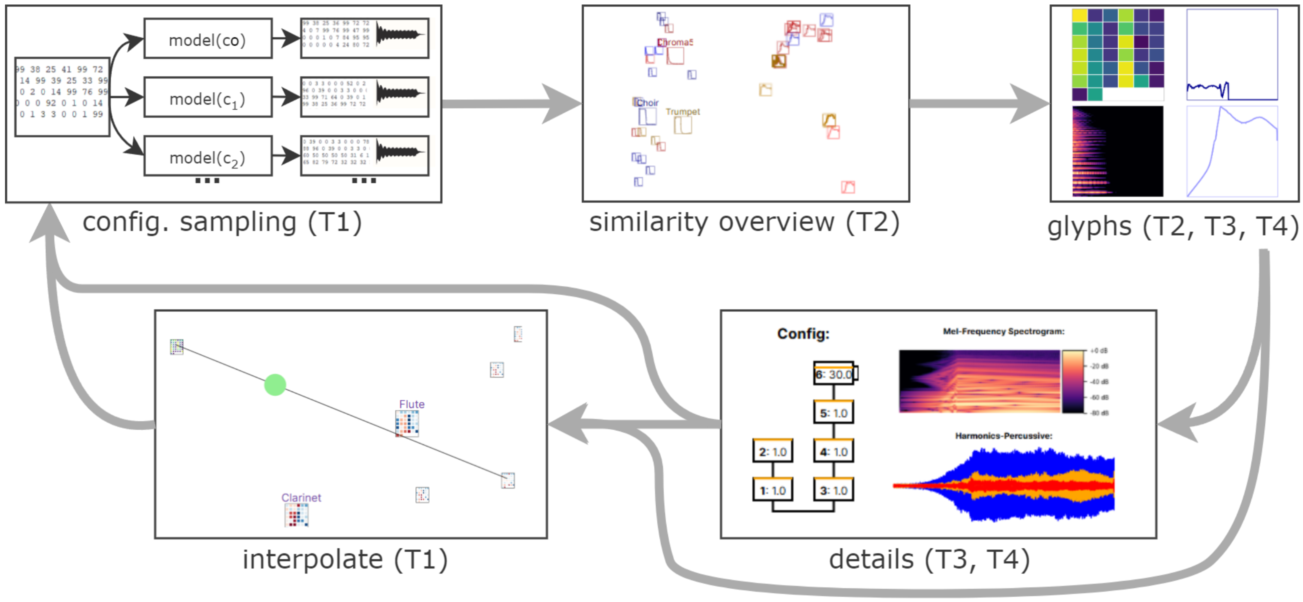


Figure 3: Our intended workflow starts with sampling configurations (top left), then exploring the visual overview using similarities and glyphs, picking interesting sounds and analyzing their features in detail, using interpolation, and iteratively refining sampling again.

3 Design

For VibeFM's development, we closely followed the design study methodology [50], in which we worked with one expert, our second author. He has six years of experience with FM synthesis and has used and programmed various FM synthesizers, including Dexed, in two movies and a few smaller projects. In bi-weekly meetings, we extracted and evaluated user needs, tasks, metrics, and visual design, and continually refined them over time.

Our target audience is professional composers, sound designers, and musicians who search for instrumentation and experiment with out-of-the-ordinary sounds using FM synthesis in their artistic works.

We expect the following workflow from our users (Figure 3): They start with sampling from presets or randomly with varying degrees of change, depending on their intention. Afterward, they explore the space using a visual overview with glyphs (small visualizations), analyze sound characteristics, and select promising sounds. They might listen to a few sounds, mark their favorites for later, or use interpolation for novel interim results. Users can fine-tune favorites by sampling similar ones, repeating previous steps in the process. In the end, users export their marked configurations for external use.

With our second author's expertise, we identify tasks for sampling and discover suitable sounds:

- T1) Iteratively sample meaningful configurations,
- T2) explore the space of configurations systematically and select favorites,
- T3) discover sound characteristics in sets of configurations,
- T4) analyze the influence of parameters on the sound and find appropriate values for sampling.

3.1 Data & Sampling

Our approach works with DX7 configurations emulated in Dexed. These configurations consist of six different operators, each combined by the algorithm for a total of 144 parameters. Sampling all combinations of parameter values from the Cartesian product is not feasible, so we reduce the number of parameters during sampling. With our expert (second author), we iteratively removed parameters used only for minor changes, aiming to obtain a small subset that spans the soundscape well. Finally, we selected the remaining 38 parameters (see Figure 4c), six for each operator, including frequencies and rates, and two global ones. Based on our experience, these have the largest impact on sound and form and can therefore produce a wide range of sounds.

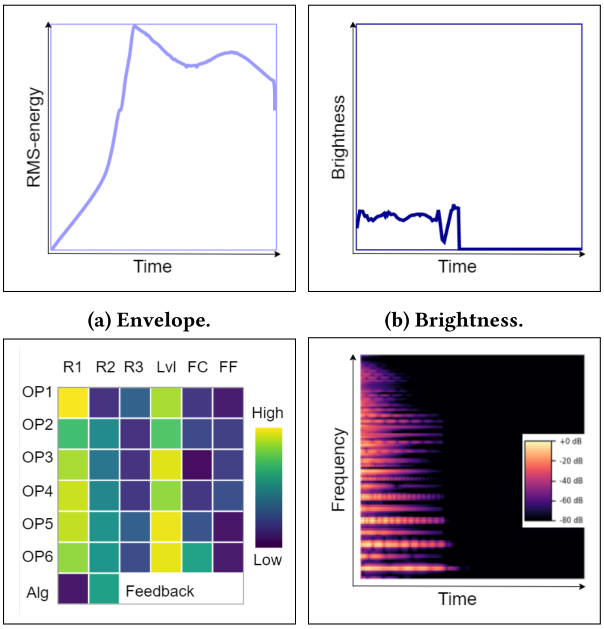
Similar to user behaviour in other works [40], we expect users to explore either large subspaces for inspiration or to inspect similar variations around a sound for fine-tuning. Without a concrete idea, composers can sample a wide range of configurations to find variety and inspiration with reduced bias. For fine-tuning, they would rather sample a subspace of similar sounds around the seed by changing single parameters, operators, or all parameters in different portions (5%, 10%, 25% ...). In our prototype, we render the audio by triggering a script on a local server that uses Dexed to emulate the configuration and Reaper's[5] built-in recording function to store an audio file and calculate metrics for the visualizations later.

3.2 Metrics

For composers to interpret our visual representations, we need metrics that describe the nature of the sound for auditive imaging. To find interesting regions, we need a way to measure the distance between sounds and visually group similar samples.

Composers are highly interested in the timbre and envelope of a sound [2, 31]. Based on previous work on timbre descriptors for sounds [39], we chose three characteristics as a starting point. First, composers must differentiate between harmonic and percussive sounds based on their needs. To do so, we extract the harmonic, percussive, and in-between features using harmonic-percussive source separation [11]. Second, to learn more about

[5] reaper.fm



(a) Envelope. **(b) Brightness.**

(c) Configuration. **(d) Mel spectrogram.**

Figure 4: Our different graphs/glyphs represent configuration and sound characteristics: (a) Envelope glyph shows its shape via rms-energy, (b) brightness glyph shows the centroid frequency over time, (c) configuration glyph displays values of chosen parameters (rates 1-3, levels, and frequencies per operator, and algorithm and feedback) as color, and (d) shows sound complexity in a mel-spectrogram.

a sound's timbre, the perceived timbral brightness is a common descriptor. We calculate the spectral centroid, which strongly correlates with this brightness [46]. For later use, we use values over time or calculate the mean. For composers to identify the proper use of a sound, the attack (rise) and decay (fall-off) phases, known as the envelope, are important characteristics. The rms-energy (root mean square) reveals how a sound rises and decays and strongly correlates with the envelope's phases [15].

Composers are mainly interested in a sample's sound, which should be reflected in our distance measure. Instead of distances between configurations, we use distances directly on rendered audio with commonly used Mel-Frequency Cepstrum Coefficients (MFCC) and the Euclidean distance [23, 30] This metric captures timbre [54] and shows characteristic similarities to the composer. In our prototype, we calculate all metrics using librosa [32] within a Python script, that runs automatically on our local server.

3.3 Visualization Design

For composers to sample and visually explore a space, VibeFM's interface consists of three main parts: A menu to control sampling (T1), visual representations, and exports, and a statistical overview to display configuration diversity (Figure 1a). Composers can interactively restrict parameters for sampling using the statistical overview (Figure 1d). The main view in the middle provides an overview (T2) using a dimensionality-reduced map (Figure 1b) displaying all configurations using small glyphs (T3, T4) [56]; small visual representations (Figure 4). Here, composers browse through space, interpolate configurations, and inspect single points (T3, T4) for additional details (Figure 1c). Since composers need flexible options for listening to and playing with sounds at any time, we connect configurations to an external synthesizer through MIDI messages.

Get an overview: Composers want to systematically explore the space and find interesting samples or whole regions (T2). A commonly used layout in visual analytics is a two-dimensional map that groups similar sounds close together and others farther apart [44], providing information about sound similarity. Using our distance metric, we create this similarity-based layout by using dimensionality reduction like t-SNE [55] or MDS [25], implemented with DruidJS [8]. Because composers focus on regions of similar sounds, we use t-SNE by default, exploiting its clustering strength. To help users navigate the space, they add presets or audio files as visual references for orientation (Figure 6).

Additionally, composers want to identify interesting regions and hide others efficiently. By encoding our metrics with color, composers can get information about the algorithm, sound clusters, the presence of harmonic/percussive parts, and brightness.

Find samples: Although colors indicate sound characteristics (T3), more fine-grained information is needed to imagine audio. Therefore, we visualize each sample with glyphs (Figure 4), small visualizations showing complex data [1].

Composers are interested in a sound's envelope, so we present a curve of the rms-energy over time. Figure 4a, for example, shows a slow attack phase and holds the sound without much fall off. Similarly, we display brightness over time, where Figure 4b indicates a mostly steady dark sound. Because only hearable elements are interesting, we smooth the curve with the rms-energy. To some users, the configurations themselves (T4) are interesting to learn controlling parameters, so we display the selected 38 parameter values with color in a matrix. For example, Figure 4c reveals mostly low-frequency settings (frequency coarse (FC) and fine (FF)) for all operators (R1). For comparing two configurations, this glyph changes colors and shows value differences, making large differences easy to spot. Lastly, composers want to get a sense of a sound's complexity, as revealed in a mel-frequency spectrogram, and let experienced users imagine it. Figure 4d shows a complex sound with many occurring frequencies.

After identifying interesting sounds, composers might want to refine them by sampling variations or by finding sounds between them. For the second, they use linear interpolation to morph favorites and search for a sweet spot (Figure 5). Users can listen to interpolated configurations live and add them as new configurations/references to the space.

Investigate details: After finding interesting samples in the overview, composers want to inspect these further. Similar to our glyphs, we show the envelope, brightness, and mel spectrogram in visualizations simultaneously (Figure 1c).

The algorithm heavily influences the sound's character, so composers are interested in connections between the operators and their parameters. Therefore, we intuitively display the algorithm's structure with its frequency ratios and output levels. For example, Figure 1c shows a low output level of operator five, indicating a limited impact on the output. In the end, composers export favorite samples for later use or sampling variations.

4 Evaluation

As there is no simple quantitative measure of success due to subjectivity and ill-defined tasks, which are common in visual analytics [45], we use qualitative feedback and anecdotal evidence from experts [37]. Therefore, we conducted an exploratory case

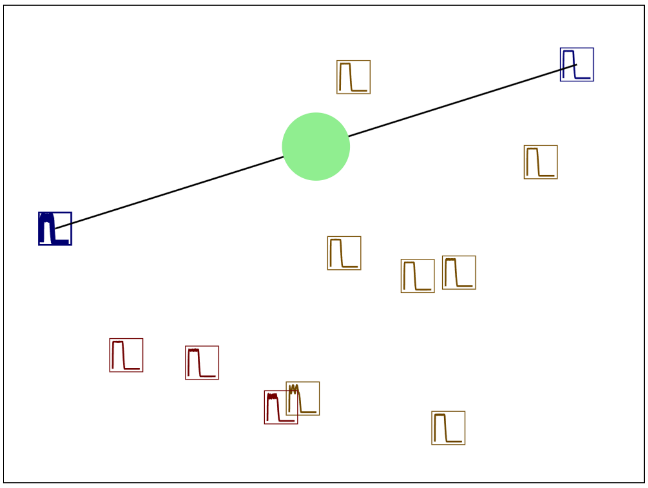


Figure 5: Similarity-based layout using envelope glyphs and active interpolation. For the latter, two configurations are linked so the user can drag along to find a sweet spot. Most samples have similar fast-attack envelopes, while two at the bottom show wobbly behavior.

study [16, 59], in which we composed a soundtrack to illustrate VibeFM in practice and highlight interesting observations during the process. Additionally, we present the feedback from semi-structured interviews with three experienced artists.

4.1 Case Study

Our second author, who has six years of experience with synthesizers, composed a soundtrack[6] using VibeFM to explore different sounds. With the help of VibeFM, he worked for about 180 minutes on the soundtrack, created all six sounds and variations except for drums, and got inspired to use variations for a stereo effect. Based on previous experience, he felt faster than experimenting with sounds the usual way, but also slower than taking presets without exploration.

Because this case study is based on a single user's experience, who was involved in the design, it introduces potential bias through prior knowledge and personal investment. Therefore, findings should not be interpreted as an objective evaluation or validation but rather showcase possible applications and inform future, more rigorous studies.

At the start, he used a random sampling strategy to get ideas and foster inspiration. He looked through the samples and found an interesting sound, but was not fully satisfied. Therefore, he sampled small adjustments around this as a second sampling strategy. This allowed him to find the characteristics he imagined by looking at the color encoding brightness. Throughout the session, he also found, for example, a bell-like sound he did not expect and used it at the start of the soundtrack. Figure 6 shows a screenshot from his session, which shows a reference and some randomly sampled sounds.

A different approach he used included more manual agency, by providing a manually created starting point or making adjustments at any time. During visual exploration in VibeFM, he got inspired to edit a sound and panned generated variations to the sides for a stereo effect, which was an unexpected use of VibeFM.

[6] github.com/snrau/VibeFM/blob/master/VibeFM_soundtrack.mp3

In this session, the interpolation and configuration glyphs were not as beneficial for him.

Concluding, he used VibeFM to create a soundtrack and found inspiration for a stereo effect and novel sounds that rarely occur in his usual work style. The visual representations helped him explore the space of generated variations to refine sounds.

4.2 Interviews

We conducted semi-structured interviews with one professional composer and two creative artists/musicians, averaging 65 minutes. Participants were recruited through convenience sampling from the researchers' personal networks to capture diverse backgrounds. The composer has extensive experience and knowledge in sound design. The two artists have considerable expertise as musicians with several instruments, and one of them has an extensive background in sound engineering.

During the interviews, we showed our prototype, explained its functionality, and let experts test it for about 45 minutes, and recorded the screen. We gathered feedback from the observations, asked questions during the testing stage and asked for think aloud insights by the participants. We especially focused on the usefulness of our approach, functions, visual representations, and potential future improvements, and were interested in VibeFM's potential for an artistic workflow and surprising findings. Because musicians are often unfamiliar with complex visualizations, we expected an initial learning curve.

Participants generally liked our approach, especially to explore (P1) and find new sounds (P2) in such *"a complex structure like FM synthesis"* (P3), but also mentioned the expected initial learning curve (P2) associated with learning to interpret the visualizations and map them to a sound, similar to auditory imaging [20]. With our global overview, which is an *"improvement over the previous editors"* (P1) and trial-and-error adjustments (P3), our approach *"complements the problem of programming a synthesizer in a convenient way, which has been difficult for decades"* (P1). Especially from an artist's perspective, it is important for P3 that the "machine" amplifies creativity and does not take over the process through ratings. According to P3, our approach supports user creativity and provides full control. On the other hand, P2 had trouble learning the projection's semantics to make sense of the overview, due to insufficient sample grouping and felt limited control over sampling, which resulted in envelopes being too extreme. Still, participants found several unexpected sounds they were *"not looking for"* (P2), and P2 realized that using visual overviews implies more possibilities than expected before, underlining the serendipity our visualizations provide.

Especially, our envelope glyph helped most in excluding unwanted sounds to organize the space (P2, P3), though for P2, the map did not cluster similar envelopes well. The other glyphs helped some participants more than others in different situations. For example, P2 had experience reading spectrograms and could imagine their sound (auditive imaging), whereas P1 was not familiar with it.

P1 was highly interested in the *"world of interim results"* between good samples using interpolation, rather than looking for perfect samples. Most participants found our interpolation feature useful, despite some unsatisfying results due to discrete steps, which led to non-smooth interpolation along the line (P2).

In general, all participants mentioned they would like to test VibeFM further and explore the possibilities, and especially P1 would use it for a composition at a suitable occasion.

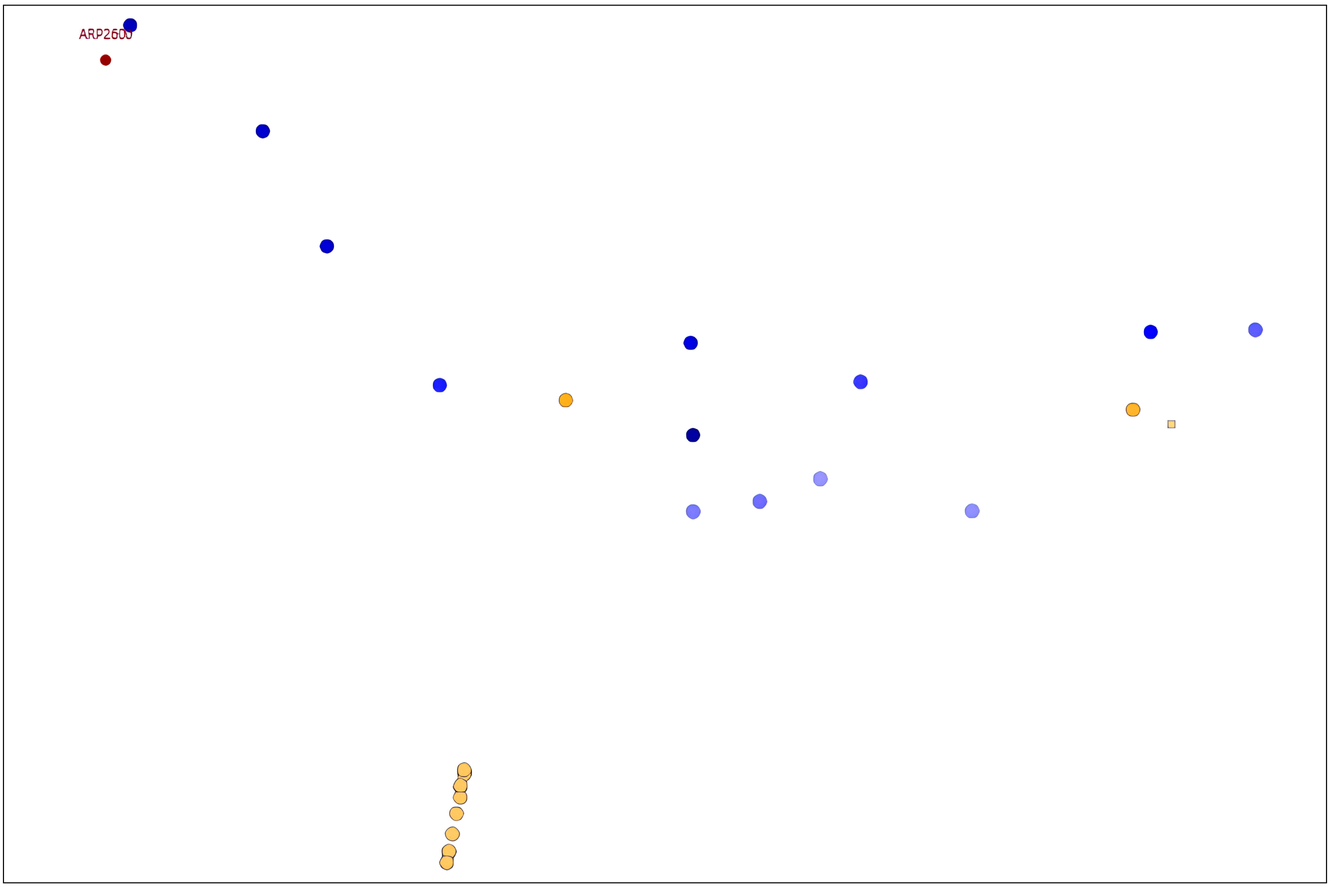


Figure 6: An upscale screenshot taken during our case study. This overview shows a reference preset named "Arp2600" in the top-left corner, which seems to have more percussive structure indicated by the red color. A sample with similar brightness is near the preset but shows more harmonic than percussive presence indicated by the blue color. At the bottom, a cluster of very similar sounds from the same starting point with a bright timbre can be spotted. Therefore, these sounds are further away from all other randomly sampled sounds.

In conclusion, all participants found our approach and visualizations helpful for exploring and discovering interesting and out-of-the-ordinary sounds. To improve VibeFM's effectiveness, participants suggested a closer connection to digital audio workstations, more control during sampling, and direct interaction with parameters. Additionally, exchanging the synthesizer, extensions to other devices, and a more polished version would benefit the approach as well (P1).

5 Discussion

During our evaluation, we found increasing evidence that our visual approach helps explore sound options and serves as a useful tool in programming synthesizers. Given the task's ill-defined and subjective nature, we found that users want full control over the creative process. Instead of optimized rankings, users require flexibility to interactively explore large sets of samples, which is also one of the goals and strengths of visual analytics [13].

Our evaluation showed that audio features and visual representations help users to varying degrees, depending on their needs and expertise. We learned that glyphs for direct sound or characteristic display, like the spectrogram and envelope glyph, perform better (auditive imaging) than statistical representations, like the configuration glyph. Although we think our metrics and glyphs cover interesting information for most composers, other users might need different sound characteristics and visual representations. Therefore, there will not be a one-size-fits-all set of metrics and glyphs.

Although our approach works with the Yamaha DX7 synthesizer, composers might also use other synthesizers. Most of our metrics and visualizations are based on audio features and, therefore, synthesizer-agnostic, working for any sound-generating model. However, the visualized algorithm, configuration structure, and connection to external programs need adjustments for different structured synthesizers.

Limitations: When using external emulators, their limitations transfer over. For example, missing changes during interpolation reduce creativity with this functionality. Adding an internal synthesizer would enable smoother interaction and novel effects such as a low-frequency oscillator (LFO) for interpolation. A further limitation is the trade-off between offering many options and providing an easy start with mostly usable sounds through fewer parameters. Re-balancing this requires adjustments to our visual representations. A key limitation is the small number of experts involved in analyzing requirements, verifying the process, and evaluating our approach, which may led to overly tailored designs and limited generalizability. Working with experienced experts comes with several challenges [49]. However, we believe that collaborating with experts like ours is necessary to extract characteristics and beneficial design concepts. Although we reflect on subjective experiences in our case study, which may differ for other users, this showcases VibeFM's strengths, as confirmed by the interviewees.

6 Conclusion

We propose VibeFM, a user-centered, interactive, and visual approach to explore frequency modulation (FM) synthesis to find novel sounds. Our design is based on systematically sampling parameter combinations and providing a visual overview with glyphs. In our evaluation, we found that a visual approach provides high flexibility and supports creativity through serendipitous findings. We received overall positive feedback from experts that our approach supports programming synthesizers, a generally hard problem.

In the future, we want to conduct a follow-up, longitudinal study with more experts to examine their workflow and flexibility using VibeFM. With the results, VibeFM can be extended to improve support for auditory imaging. Adding generative artificial intelligence [27] could benefit sampling strategies, but should not replace manual sampling to reduce biases. Furthermore, VibeFM's approach could be transferred to other sound-generating techniques in a composer's instrumentation phase.

7 Ethical Standards

The authors declare that no conflicts of interest are associated with this research. All participants gave consent to record their data for anonymised use in this work. No living subject was harmed during studies. The authors used generative language models (ChatGPT and Grammarly) to assist with improving the grammar, clarity, and readability of text written by the authors. Common prompts the authors used for all sections were: "Improve the following text on writing and grammar: sentence or paragraph". To improve the quality of one image, the authors used an AI-powered tool (iloveimg) to upscale the resolution, ensuring no additional content was generated during the process. All scientific content, ideas, technical descriptions, and conclusions are the authors' own. The tool was not used to generate research ideas, methods, results, or analyses, and all revisions were carefully reviewed and approved by the authors.

Acknowledgments

Funded by the Ministry of Science, Research and the Arts Baden-Wuerttemberg in the Artificial Intelligence Software Academy (AISA) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 251654672 – TRR 161.

References

[1] Rita Borgo, Johannes Kehrer, David H. S. Chung, Eamonn Maguire, Laramee Robert S., Hauser Helwig, Ward Matthew, and Chen Min. 2013. Glyph-based Visualization: Foundations, Design Guidelines, Techniques and Applications. In *Eurographics (State of the Art Reports)*. 39–63. https://doi.org/10.2312/conf/EG2013/stars/039-063

[2] Pierre Boulez. 1987. Timbre and composition - timbre and language. *Contemporary Music Review* (1987), 161–171. https://doi.org/10.1080/07494468708567057

[3] Stephen Brade, Bryan Wang, Mauricio Sousa, Gregory Lee Newsome, Sageev Oore, and Tovi Grossman. 2024. SynthScribe: Deep Multimodal Tools for Synthesizer Sound Retrieval and Exploration. In *Proc. Int. Conf. Intelligent User Interfaces (IUI)*. 51–65. https://doi.org/10.1145/3640543.3645158

[4] Warren Brodsky, Yoav Kessler, Bat-Sheva Rubinstein, Jane Ginsborg, and Avishai Henik. 2008. The mental representation of music notation: notational audiation. *Journal of Experimental Psychology: Human Perception and Performance* (2008), 427–445. https://doi.org/10.1037/0096-1523.34.2.427

[5] Dylan Cashman, Shah Rukh Humayoun, Florian Heimerl, Kendall Park, Subhajit Das, John Thompson, Bahador Saket, Abigail Mosca, John Stasko, Alex Endert, Michael Gleicher, and Remco Chang. 2019. A User-based Visual Analytics Workflow for Exploratory Model Analysis. *Computer Graphics Forum (CGF)* (2019), 185–199. https://doi.org/10.1111/cgf.13681

[6] Franco Caspe, Andrew McPherson, and Mark Sandler. 2022. DDX7: Differentiable FM Synthesis of Musical Instrument Sounds. In *Proc. Conf. Int. Society for Music Information Retrieval (ISMIR)*. 608–616. https://doi.org/10.5281/zenodo.7343063

[7] John M Chowning. 1973. The synthesis of complex audio spectra by means of frequency modulation. *Journal Audio Engineering Society (AES)* (1973), 526–534.

[8] Rene Cutura, Christoph Kralj, and Michael Sedlmair. 2020. DRUID$_{JS}$ – A JavaScript Library for Dimensionality Reduction. In *Proc. IEEE Visualization Conf. (VIS)*. 111–115. https://doi.org/10.1109/VIS47514.2020.00029

[9] Palle Dahlstedt. 2001. Creating and exploring huge parameter spaces: Interactive evolution as a tool for sound generation. In *Proc. Int. Computer Music Conference (ICMC)*. 1–8.

[10] Palle Dahlstedt. 2007. *Evolution in Creative Sound Design*. Springer, 79–99. https://doi.org/10.1007/978-1-84628-600-1_4

[11] Jonathan Driedger, Meinard Müller, and Sascha Disch. 2014. Extending Harmonic-Percussive Separation of Audio Signals.. In *Proc. Conf. Int. Society for Music Information Retrieval (ISMIR)*. 611–616. https://doi.org/10.5281/zenodo.1415226

[12] Eric Easthope. 2023. SnakeSynth: New Interactions for Generative Audio Synthesis. In *Proc. Int. Conf. New Interfaces for Musical Expression (NIME)*. 612–619. https://doi.org/10.5281/zenodo.11189319

[13] Alex Endert, M Shahriar Hossain, Naren Ramakrishnan, Chris North, Patrick Fiaux, and Christopher Andrews. 2014. The human is the loop: New directions for visual analytics. *Journal of Intelligent Information Systems* (2014), 411–435. https://doi.org/10.1007/s10844-014-0304-9

[14] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. 2017. Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders. In *Proc. Int. Conf. Machine Learning (ICML)*. 1068–1077. https://doi.org/10.5555/3305381.3305492

[15] Antti Eronen and Anssi Klapuri. 2000. Musical instrument recognition using cepstral coefficients and temporal features. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing. (ICASSP)*. 753–756. https://doi.org/10.1109/ICASSP.2000.859069

[16] Bent Flyvbjerg[1]. 2011. What is a case study? *The Sage handbook of qualitative research* (2011), 1 – 301.

[17] Leandro Garber, Tomás Ciccola, and Juan Cruz Amusategui. 2021. AudioStellar, an open source corpus-based musical instrument for latent sound structure discovery and sonic experimentation. In *Proc. Int. Computer Music Conf. (ICMC)*. 62–67.

[18] Oscar Gomez, Kaustuv Kanti Ganguli, Leonid Kuzmenko, and Carlos Guedes. 2020. Exploring Music Collections: An Interactive, Dimensionality Reduction Approach to Visualizing Songbanks. In *Proc. Int. Conf. Intelligent User Interfaces (IUI)*. 138–139. https://doi.org/10.1145/3379336.3381461

[19] Frank Heyen, Quynh Quang Ngo, and Michael Sedlmair. 2023. Visual Overviews for Sheet Music Structure. In *Proc. Conf. Int. Society for Music Information Retrieval (ISMIR)*. 692–699. https://doi.org/10.5281/zenodo.10265383

[20] Timothy L Hubbard. 2010. Auditory imagery: empirical findings. *Psychological Bulletin* (2010), 302–329. https://doi.org/10.1037/a0018436

[21] Colin G. Johnson. 1999. Exploring the sound-space of synthesis algorithms using interactive genetic algorithms. In *Proc. Symp. on Musical Creativity*. 20–27.

[22] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. 2008. Visual analytics: Definition, process, and challenges. In *Information Visualization*. Springer, 154–175. https://doi.org/10.1007/978-3-540-70956-5_7

[23] John Kominek, Tanja Schultz, and Alan W. Black. 2008. Synthesizer voice quality of new languages calibrated with mean mel cepstral distortion. In *Proc. Speech Technology for Under-Resourced Languages (SLTU)*. 63–68.

[24] Gordan Krekovic. 2019. Insights in habits and attitudes regarding programming sound synthesizers: A quantitative study. In *Proc. Sound and Music Computing Conf. (SMC)*. 316–322.

[25] J. B. Kruskal. 1964. Nonmetric Multidimensional Scaling: A Numerical Method. *Psychometrika* (1964), 115–129. https://doi.org/10.1007/BF02289694

[26] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brebisson, Yoshua Bengio, and Aaron Courville. 2019. MelGAN: Generative adversarial networks for conditional waveform synthesis. In *Proc. Int. Conf. Neural Information Processing Systems (NeurIPS)*. 14910–14921. https://doi.org/10.48550/arXiv.1910.06711

[27] Gwendal Le Vaillant, Thierry Dutoit, and Sébastien Dekeyser. 2021. Improving synthesizer programming from variational autoencoders latent space. In *Proc. Int. Conf. Digital Audio Effects (DAFx)*. 276–283. https://doi.org/10.23919/DAFx51585.2021.9768218

[28] Gwendal Le Vaillant, Thierry Dutoit, and Rudi Giot. 2020. Analytic vs. holistic approaches for the live search of sound presets using graphical interpolation. In *Proc. Int. Conf. New Interfaces for Musical Expression (NIME)*. 227–232. https://doi.org/10.5281/zenodo.4813330

[29] Sean Luke and Victoria Hoyle. 2023. Co-creative Music Synthesizer Patch Exploration. In *Proc. Int. Conf. Cognitive Computing (ICCC)*. 412–416.

[30] Naotake Masuda and Daisuke Saito. 2023. Quality-diversity for synthesizer sound matching. *Journal of Information Processing* (2023), 220–228.

[31] Max V. Mathews. 1963. The Digital Computer as a Musical Instrument. *Science* (1963), 553–557. https://doi.org/10.1126/science.142.3592.553

[32] Brian McFee, Colin Raffel, Dawen Liang, Daniel P.W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and Music Signal Analysis in Python. In *Proc. Python in Science Conference (SciPy)*. 18–24. https://doi.org/10.25080/Majora-7b98e3ed-003

[33] Gerardo Meza. 2023. Exploring the potential of interactive Machine Learning for Sound Generation: A preliminary study with sound artists. In *Proc. Int. Conf. New Interfaces for Musical Expression (NIME)*. 604–607. https://doi.org/10.5281/zenodo.11189313

[34] Matthias Miller, Julius Rauscher, Daniel A. Keim, and Mennatallah El-Assady. 2022. CorpusVis: Visual Analysis of Digital Sheet Music Collections. *Computer Graphics Forum (CGF)* (2022), 283–294. https://doi.org/10.1111/cgf.14540

[35] Fabian Mörchen, Alfred Ultsch, Mario Nöcker, and Christian Stamm. 2005. Databionic Visualization of Music Collections According to Perceptual Distance.. In *Proc. Conf. Int. Society for Music Information Retrieval (ISMIR)*. 396–403. https://doi.org/10.5281/zenodo.1417967

[36] Fabian Mörchen, Alfred Ultsch, Mario Nöcker, and Christian Stamm. 2006. Visual mining in music collections. In *From Data and Information Analysis to Knowledge Engineering*. 724–731. https://doi.org/10.1007/3-540-31314-1_89

[37] Tamara Munzner. 2009. A nested model for visualization design and validation. *IEEE Trans. Visualization and Computer Graphics (TVCG)* (2009), 921–928. https://doi.org/10.1109/TVCG.2009.111

[38] Kasper Nielsen. 2020. Practical linear and exponential frequency modulation for digital music synthesis. In *Proc. Int. Conf. Digital Audio Effects (DAFx)*. 1–8.

[39] Geoffroy Peeters, Bruno L. Giordano, Patrick Susini, Nicolas Misdariis, and Stephen McAdams. 2011. The Timbre Toolbox: Extracting audio descriptors from musical signals. *Journal Acoustical Society of America* (2011), 2902–2916. https://doi.org/10.1121/1.3642604

[40] Simeon Rau, Frank Heyen, Benedikt Brachtel, and Michael Sedlmair. 2025. MAICO: A Visualization Design Study on AI-Assisted Music Composition. *IEEE Trans. Visualization and Computer Graphics (TVCG)* (2025), 1–16. https://doi.org/10.1109/TVCG.2025.3539779

[41] Simeon Rau, Frank Heyen, Stefan Wagner, and Michael Sedlmair. 2022. Visualization for AI-Assisted Composing. In *Proc. Conf. Int. Society for Music Information Retrieval (ISMIR)*. 151–159. https://doi.org/10.5281/zenodo.7316618

[42] Roger Redgate. 2019. Do you hear what I hear? Audiation and the Compositional Process. *Principles of Music Composing* (2019), 19–27.

[43] Gerard Roma. 2024. Sound matching using synthesizer ensembles. In *Digital audio effects conf. (DaFx)*. 612–619.

[44] Dominik Sacha, Leishi Zhang, Michael Sedlmair, John A. Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C. North, and Daniel A. Keim. 2017. Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis. *IEEE Trans. Visualization and Computer Graphics (TVCG)* (2017), 241–250. https://doi.org/10.1109/TVCG.2016.2598495

[45] Jean C. Scholtz. 2017. *User-Centered Evaluation of Visual Analytics*. Springer. https://doi.org/10.1007/978-3-031-02605-8

[46] Emery Schubert and Joe Wolfe. 2006. Does timbral brightness scale with frequency and spectral centroid? *Acta acustica united with acustica* (2006), 820–825.

[47] Diemo Schwarz. 2006. Concatenative sound synthesis: The early years. *Journal of New Music Research* (2006), 3–22. https://doi.org/10.1080/09298210600696857

[48] Michael Sedlmair, Christoph Heinzl, Stefan Bruckner, Harald Piringer, and Möller Torsten. 2014. Visual Parameter Space Analysis: A Conceptual Framework. *IEEE Trans. Visualization and Computer Graphics (TVCG)* (2014), 2161–2170. https://doi.org/10.1109/TVCG.2014.2346321

[49] Michael Sedlmair, Petra Isenberg, Dominikus Baur, and Andreas Butz. 2011. Information visualization evaluation in large companies: Challenges, experiences and recommendations. *Information Visualization* (2011), 248–266. https://doi.org/10.1177/1473871611413099

[50] Michael Sedlmair, Miriah Meyer, and Tamara Munzner. 2012. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE Trans. Visualization and Computer Graphics (TVCG)* (2012), 2431–2440. https://doi.org/10.1109/TVCG.2012.213

[51] Keisuke Shiro, Ryotaro Miura, Changyo Han, and Jun Rekimoto. 2019. An Intuitive Interface for Digital Synthesizer by Pseudo-intention Learning. In *Proc. Int. Audio Mostly Conf. (AM)*. 39–44. https://doi.org/10.1145/3356590.3356598

[52] Ben Shneiderman. 2007. Creativity support tools: accelerating discovery and innovation. *Communications ACM* (2007), 20–32. https://doi.org/10.1145/1323688.1323689

[53] Nikhil Singh. 2020. *Sifting sound interactive extraction, exploration, and expressive recombination of large and heterogeneous audio collections*. Ph. D. Dissertation. Massachusetts Institute of Technology.

[54] Hiroko Terasawa, Malcolm Slaney, and Jonathan Berger. 2005. A timbre space for speech. In *Proc. Interspeech*. 1729–1732. https://doi.org/10.21437/Interspeech.2005-285

[55] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)* (2008), 2579–2605.

[56] Matthew Ward, Chun-houh Chen, Wolfgang Karl Härdle, and Antony Unwin. 2008. Multivariate Data Glyphs: Principles and Practice. In *Handbook of Data Visualization*. Springer, 179–198. https://doi.org/10.1007/978-3-540-33037-0_8

[57] Yuting Yang, Zeyu Jin, Connelly Barnes, and Adam Finkelstein. 2023. White Box Search Over Audio Synthesizer Parameters.. In *Proc. Conf. Int. Society for Music Information Retrieval (ISMIR)*. 190–196. https://doi.org/10.5281/zenodo.10265255

[58] Matthew John Yee-King, Leon Fedden, and Mark d'Inverno. 2018. Automatic Programming of VST Sound Synthesizers Using Deep Networks and Other Techniques. *IEEE Trans. Emerging Topics in Computational Intelligence* (2018), 150–159. https://doi.org/10.1109/TETCI.2017.2783885

[59] Robert K Yin. 2018. *Case Study Research and Applications*. Sage Publication, Inc.