xCoAx 2022 10th Conference on
Computation, Communication, Aesthetics & X
2022.xCoAx.org
Coimbra, Portugal

Autolume-Live: Turning GANs into a Live VJing tool

Jonas Kraasch
Jkraasch@sfu.ca
Simon Fraser University,
Vancouver, Canada

Philippe Pasquier
pasquier@sfu.ca
Simon Fraser University,
Vancouver, Canada

Creative Artificial Intelligence is increasingly used to generate static and animated visuals. While there are a host of systems to generate images, videos and music videos, there is a lack of real time video synthesizers for live music performances. To address this gap, we propose *Autolume-Live*, the first GAN-based live VJing-system for controllable video generation. By analysing the needs of VJs and current Deep Learning-based audio-visual systems, we developed a live video synthesiser that extracts musical features, such as the amplitude, pitch and onset strength, and allows mapping these to generate trajectories in the frame latent space. These frames can further be manipulated through a MIDI-Interface which allows the user to improvise, manipulate the Neural Network and adjust the output to accompany live music performances in a co-creative way.

Keywords Creative AI, GAN, Video Generation, Interactive System, Audio-Visual, VJ

1. Introduction

With an increasing amount of generative models in AI and Deep Learning (DL), artists are finding ways to use these systems for their creative expression. This is extensively seen in musical and visual practices. Creative agents can be used to co-create live coding musical performances (Wilson et al. 2021). Using Generative Adversarial Networks (GANs) (Goodfellow et al. 2014; Karras, Laine, and Aila 2018; Karras et al. 2020)), both curated installations (Obvious 2018) and visual live installations (Klingemann 2019) have been created. With the rise of Creative AI, new ways to create experiences and channel creativity are emerging. One domain that is not influenced by Deep Learning so far is live audio-visual performances. For live musical performances, it is common to have a VJ use pre-recorded clips to create a responsive visual experience. We are interested in a way to combine VJing with GANs. We choose to use GANs over other generative models, such as Diffusion models, due to their high-fidelity, comparatively fast inference time and because they already have a large following of artists.

While there are audio-visual installations using GANs, they are bound to offline creation. Artists have used Style-transfer, Deep Dream and other frameworks to create music videos by manipulating pre-recorded videos (Hardcore-analhydrogen 2018). There are artistic works extracting music features from a track and mapping it to the latent space of a GAN (Siegelman 2019; Klingemann 2018; Jonathan 2021.; Alafriz 2021; Steenbrugge 2020). But all these systems either lack documentation or a user interface, which makes the systems inaccessible for new users. Furthermore, none of these systems tackle live video generation. To bridge the gap from automated offline systems to a realtime VJing Software we propose *Autolume-Live*, to our knowledge the first live Music Visualiser and VJing software based on GANs (Goodfellow et al. 2014; Karras, Laine, and Aila 2018; Karras et al. 2020). The program explores the latent space of trained models and drives the image generation through music. The algorithm uses audio features, such as amplitude, onset timing and pitches to influence the images generated by the model. Additionally, we incorporate a MIDI-controller as an interactive tool to allow artists to edit and manipulate the generation process using Network Bending (Broad, Leymarie, and Grierson 2021) and GANSpace (Härkönen et al. 2020). We show that with modern graphic cards and developments in generative models, interpretable AI and compression techniques, it is possible to create an interface that allows artists to accompany their live music performances with adjustable visuals.

We first discuss different types of visual systems and showcase offline approaches using Deep Learning in section 2. We focus on offline approaches, as there has been no research in the domain of Deep Learning based live audio-driven visual generation. We present *Autolume-Live* in section 3, describing its different modules and the VJing interface that we implemented while

focussing on the characteristic of VJing tools introduced by Hook et al. (Hook et al. 2011). In addition to the system, we have also created two installations using *Autolume-Live*. We briefly outline the process behind these pieces in section 4. Lastly, we discuss the possible changes and iterations that could be made on our system to improve the usability and the expressiveness of the tool (Section 5).

2. Background

2.1 Audio-reactive Visual Systems

Before discussing our framework, we first have to understand the different types of real time audio-driven visual software. First, there are *Music Visualisers*, which map musical feature data to parameters of a visual system. While Taylor et al. synthesise a complex scene with virtual characters and spaces that respond to audio features (2006), commonly music is visualised by showing the live spectrogram in different formats. Secondly, *Video Synthesisers* relate to the mapping of any signal to control visual generation. Most *Music Visualisers* and *Video Synthesisers* can be used offline and online. A specific application of these systems for live performance is *VJing software*. These generally combine the audio-reactive components of *Music Visualisers* and *Video Synthesisers* and add an interactive component that allows an artist to manipulate media live. By applying visual transformations or adjusting mappings, every performance can become a unique experience based on the performer, aka Visual Jockey (VJ). In their study, Hook et al. explore the ways VJs adapt and appropriate technology to create visual performances through dialogue and the creation of a documentary film (2011). They focussed on multiple artists and performances and extracted the essence of the medium for those artists. When it comes to VJing tools, there is a need for a physical, tangible, interface that responds with immediacy, and allows the user to accompany music live and improvise. The tool should allow the artist to control a variety of parameters, while keeping usability in mind. The benefit of a physical interface, such as a VJ controller, is that it can be seen as another instrument that can be incorporated into a performance.. Lastly, the visuals should be responsive to its context and surrounding, e.g. the music, and every performance should have the possibility to be unique.

TouchDesigner and Resolume are popular VJing software (Resolume 2022; Derivative 2022). By looking at current VJing practices, we see a trend to use video loops, shaders and generative algorithms. These softwares incorporate multiple levels of music analysis. Both low level features such as amplitude, onset strength, timbre etc., and high level features, for example affective features computed through sentiment analysis, can be used to modify parameters, such as colour, shape and position. When it comes to VJing, response time is important. Hence, TouchDesigner and Resolume focus on systems that are efficient

and can process information with little latency. For that reason these VJs have not used GANs and other DL based generative models, and to our knowledge, these approaches have not seen any use in any live performances.

2.2 Offline Audio-Reactive Systems

Our goal is to implement a Deep Learning based VJing tool. When it comes to creating music videos using AI it is either possible to do frame synthesis, i.e. generating every frame from scratch, or create collages, stitching together videos to fit the music (Fan et al. 2016). Our system is based on frame-by-frame (FBF) generation, hence we will be focussing on these music-driven video models.

By moving away from live visualisation, FBF approaches can do further audio feature extraction and can use more complex generative algorithms. For example, it is possible to use GANs to generate the visuals and use the audio features to manipulate the image output, without worrying about the runtime. The current literature comprises three different ways to generate audio-reactive videos using GANs: *Latent Space Traversal, Latent Space Interpolation, Chroma-Based Interpolation* (Brouwer 2020; Siegelman 2019; Alafriz 2021). Additionally, these offline frameworks describe further manipulations that can be performed on the video feed.

GANs use vectors sampled from a latent space to generate images. Vectors that are close together in said space are mapped to visually similar images. By moving around in the latent space it is possible to generate smooth videos interpolating between different key-frames. Researchers and artists have used this quality of GANs to create videos showcasing images morphing from one into another. The three mentioned offline audio-reactive systems use the topology of the latent space in different ways.

Fig. 1. Example generated with *Deep Music Visualiser* (Siegelman 2022). (https://www.youtube.com/watch?v=L7R-yBZ5QYc).

One way of mapping audio to video through movement in the latent space is based on a random walk. We call this approach *Latent Space Traversal* (Fig. 1) and a version of it was used in Deep Music Visualiser (DMV) (Siegelman 2022). In this case, the amplitude of the spectrogram and the shift in amplitudes between time steps are used to vary the step size of the random walk. This approach of audio-reactive image generation has the benefit that it can respond to music without any harmonic, percussive separation, but it is unable to capture repetition which is a common quality of music. As the model moves into random directions at every timestep it is unable to return previous points based on the audio. Furthermore, this approach lacks a variety of parameters, since only the audio features and a multiplier to the step-size are used.

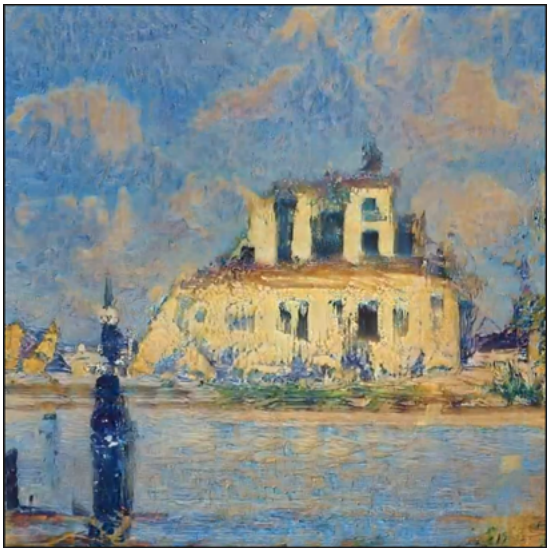

Fig. 2. Example generated with *Lucid Sonic Dreams* (Alafriz 2022). (https://www.youtube.com/watch?v=l-nGC-ve7sI).

Another way of navigating the latent space is Latent Space Interpolation, i.e. interpolating between preset positions in the latent space. While the previous *Latent Space Traversal* was unable to capture repetition, *Latent Space Interpolations* allow creating loops. Lucid Sonic Dream (LSD) (Alafriz 2022) is a system that uses this approach, similar to DMV the amplitude and shift in amplitudes are mapped to the speed of the interpolation. While looping is one of the strengths of *Latent Space Interpolations* this does create a repetitive experience. To circumvent this problem, LSD uses a combination of both *Latent Space Interpolations* and random walks. The harmonic and percussive tracks are separated. While the harmonic track is used to modulate the speed of the loops, the percussive track is used to introduce a momentary “pulse”. This “pulse” is created by performing a random step as seen in the Deep Music Visualiser for one time step and then returning to interpolating to the next latent vector. An example of a video generated by LSD is shown in Figure 2.

Fig. 3. Example *Chroma-Based Interpolation* as implemented by Brouwer (Brouwer 2020). (https://wavefunk.xyz/assets/audio-reactive-stylegan/rhodes.mp4).

Lastly, there is *Chroma-Based Interpolation*, showcased in Figure 3. Instead of using the amplitude of the audio signal, the pitches are calculated through a chromagram and mapped to the visuals. A chromagram is computed by clustering a spectrogram into frequency bins for every pitch (Ezzaidi et al. 2012; Ellis 2007). In western music theory, normally 12 bins are used. In LSD and DMV, conditional GANs can be used, where a class vector is appended to the input to define the content of the synthesised image (Oeldorf and Spanakis 2019; Mirza and Osindero 2014). It is then possible to use a chromagram to influence the content of the video, i.e. every pitch is linked to a different content class. A different approach to incorporate pitch is to map every pitch to a key-frame (Brouwer 2020). The resulting video consists of interpolations between keyframes based on the pitches present in the music. For example, if the note C is mapped to a blue frame and the note D to a red frame. At every point in the music where the letter C is played on its own the video would show the blue frame, but if multiple notes are played the two keyframes are interpolated, i.e. if we play C and D together a purple frame will be shown. The mixing ratio between keyframes is the amplitude of the responding pitches. While this approach implicitly captures repetition through the pitch, it is not able to represent atonal or monotonous musical. Without any pitch changes, the mapping will create a static video.

In addition to using short-term features, amplitude and pitch, Brouwer incorporates long-term musical features to visualise long-term musical structures in the music. By applying Laplacian-Segmentation (McFee and Ellis 2014) on the audio track, the song is clustered into sections and different sets of latent vectors are used for the distinct parts in the music (Brouwer 2020).

3. Autolume-Live

We are interested in creating a live visual engine that can be controlled by discrete triggers and continuous controls. In particular, we focus on creating a VJing-tool that uses both audio and a controller as its input. As shown in Figure 4, our system uses an Audio Module to extract onset strength, amplitude and pitch and maps these either to the latent space W or the noise term that is added to every layer. The Controller Module processes the interactions with the MIDI-Controller, applying affine transformations to the Synthesis Modules layers or manipulating the latent space (Fig. 4.). Our system runs live to make it possible for the user to immediately see the results of their interaction. This opens the door to live improvisations and a feedback loop where the output of *Autolume-Live* impacts the user's creative vision.

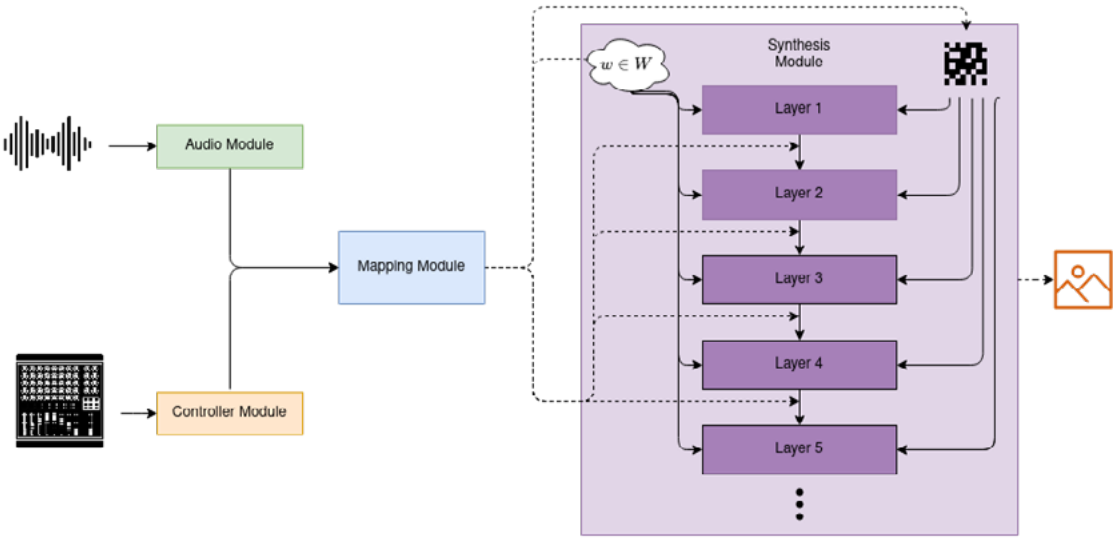

Fig. 4. *Autolume-Lives* framework showing how we mapped audio and controller signals into manipulations for the Synthesis Module.

3.1 Frame Synthesis Module

3.1.1 StyleGAN2-ada

Autolume-Live is using StyleGAN2(-ada) to generate an image at every timestep (Karras et al. 2020). StyleGan2 is an adaptation of GAN which uses a Discriminator to train a generative model by learning a boundary between real and generated samples. The training of these two models results in a two-player mini-max game. The Generator is trying to minimise the likelihood of the Discriminator correctly classifying its output as generatd. The Discriminator tries to maximise the probability of it correctly classifying a sample as real or generated.

We choose to use NVIDIA's StyleGAN2 framework for its high quality generative power. In particular we use StyleGAN2-ada to train our model. The generator stays the same, but to reduce the amount of necessary data further

augmentations are introduced in the Discriminator training process. The dataset size is important in the context of VJing, because our objective is to create a system that is usable by artists, who do not have access to large datasets or want to use their creations as training data. Furthermore, StyleGAN2(-ada) is already widely used by artists meaning they could plug in their already trained models and also use services, such as Runway ML (Runway 2022) to train their models without ever having to touch any code. Nevertheless, other generative image models could be used instead and in our framework, the model architecture is a module that can be exchanged in future iterations according to newer research.

StyleGAN2(-ada)'s Generator model can be divided into a *Mapping Network* and a *Synthesis Network*. The Mapping Network disentangles the latent space. It maps a normally distributed latent Vector *z* to an intermediate latent space *W*. By introducing an intermediate *Mapping Network*, the latent space is disentangled, this means that moving around in that space returns more predictable outcomes and smoother transitions. The *Synthesis Network* uses the mapped latent vector *w∈W* to generate an image. It incorporates an additional noise term at every resolution level. The latent vector dictates the content of the generated image, while the noise term adds variation to the output. Furthermore, the lower levels of the Synthesis Network generate the coarse structure of the image while the higher levels add fine details and textures. When generating videos with StyleGAN2, we realised that sampling noise at every time step creates jittering between frames. Because of the jitter, there is constant change in the visuals, which sometimes distracts from the audio-reactive nature of the algorithm. Hence, we decide to sample a single noise term on startup and reuse it throughout the video.

3.1.2 Content-Aware GAN Compression to Reach Real Time Generation

Immediacy and responsiveness are important factors of any VJing-system. VJing-Softwares recommend a delay below 50-milliseconds that is reaction within a frame at 20 FPS or optimally for the delay to be unperceivable by the human eye under 20-milliseconds which corresponds to 60FPS. Since our audio and visual modules are linked together, it is important that the combination of audio feature extraction and image generation stays below this threshold. In our tests, using the standard styleGAN2 architecture on an NVIDIA QUADRO 5000 with 16GB of VRAM the highest resolution we can generate, while keeping a framerate around 60FPS is 128128. Generating images in a higher resolution makes the framerate drop drastically, e.g. with a resolution of 256x256 we reach a framerate of 40FPS and with 512x512 the framerate drops to 24FPS. One way to increase the inference speed of a model is to compress it as a preprocessing step of system.

The process of network compression focuses on (1) reducing the number of parameters by pruning the model (Blalock et al. 2020; Huang et al. 2018; Han et al.

2015), and (2) distillation (Hinton, Vinyals, and Dean 2015), using the original model as a teacher for a smaller model. Where these ideas work well for classification tasks, they do not work on GANs. But, by introducing content-aware pruning and distillation Y. Liu et al. have shown that it is possible to compress StyleGAN2(-ada) and reduce the complexity of the network counted in the number of floating point operations (FLOPS) by a factor of 11 (Y. Liu et al. 2021). The idea of content-aware compression is that a content-parsing network is trained to find the features in the model which have the biggest impact on the visuals. During the pruning and distillation process this model masks the generated image of both the pruned and teacher model. Then the distillation loss is applied on the masked image generated by the teacher and the pruned model. This enforces that the salient objects in an image are distilled into the pruned model.

3.2 Audio Module

We expand the offline techniques described in section 2 to run live. To do so, we use a monophonic audio stream with a chunk size of 1024, a sampling rate of 44100 and compute the mel-spectrogram, chromagram (Ellis 2007), onset detection and strength locally. For both the spectrogram and the chromagram, we use a FFT window size of 2048. We compute the mel-spectrogram instead of a normal spectrogram, as it better represents the perceived amplitude of the different frequencies by human ears. For efficiency we detect onsets and compute their strengths by approximating a threshold to track onsets with a moving Root Mean Square over the last second.

We originally computed the harmonic percussive decomposition (Driedger, Müller, and Disch 01 2014), but due to the small active window size and efficiency of the algorithm the quality of the decomposition did not justify the drop in framerate. As *Autolume-Live* takes a live signal as its input, the analysis is noisier than the previous offline approaches, which leads to flickering in the images. To reduce the noisiness and make the visuals respond smoother we compute the moving average over the last 3 steps of the audio signal before extracting audio features.

3.3 Controller Module

Autolume-Live is meant as an expressive interactive tool to accompany live audio performances. Jonathan Hook et al. explore the needs of VJs through an in depth qualitative study (Hook et al. 2011). They show the need for a physical interface that can be used for performances, i.e. an interactive system that is visible to the audience and a tool that gives haptic feedback. This is why we chose to integrate a MIDI-Controller, which is a common tool for VJing, into our system. Through its buttons the controller allows the user to toggle binary interactions, while its faders are a useful tool for continuous manipulations. For our working system we

use the Behringer BCF2000 controller which has eight faders, 16 buttons and 8 infinite knobs. An additional benefit of using a physical controller in comparison to a digital interface is the possibility of parallel interaction using multiple hands and fingers. This allows the user to adjust multiple parameters at the same time where a mouse would limit the intractability.

3.4 Mapping Module

We use the features we extract from the live audio feed and the input from the controller to drive the image generation. In the following, we describe the mappings we have implemented so far, but due to the modality of the system artists are able to use their own mappings for both the audio-reactive images and the interactions with the controller.

3.4.1 Audio Mapping

Autolume-Live is implementing *Chroma-Based Interpolations* (Fig. 5), *Latent Space Traversal* and *Latent Space Interpolation* (Fig. 6), which can be switched between on the fly. But, the system is written to be modular, where future iterations could allow the user to reconfigure the mappings to their liking. Additionally, we use the strength of the onset, i.e. the amplitude of the spectrogram when an onset was detected as a multiplier to the standard deviation of the noise that is injected in StyleGAN2(-ada). This creates a perceivable change in the image.

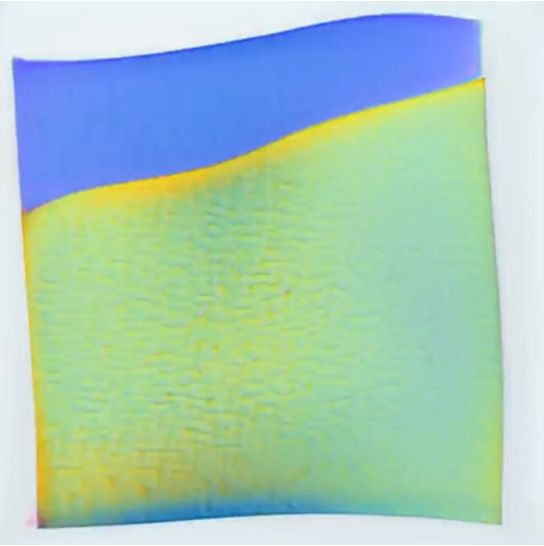

Fig. 5. Example of *Chroma-Based Interpolation* given a digital piano input to our system. (https://vimeo.com/670850243).

For both the chroma-agnostic *Latent Space Traversal* and *Interpolation* (Section 2.2.) approaches, we use the same audio mapping, only adjusting the directional vector to either be random or pointing to a specific seed. We normalise the directional vector, so that the step size before applying audio mappings is 1. The vector

is then multiplied by both the average amplitude of the spectrogram at the current time step and the difference between the current average amplitude and the previous. This leads to bigger steps when the amplitude is high or the amplitude shifts drastically, hence the biggest step would occur when a high amplitude, loud sound, follows a low amplitude, e.g. silence.

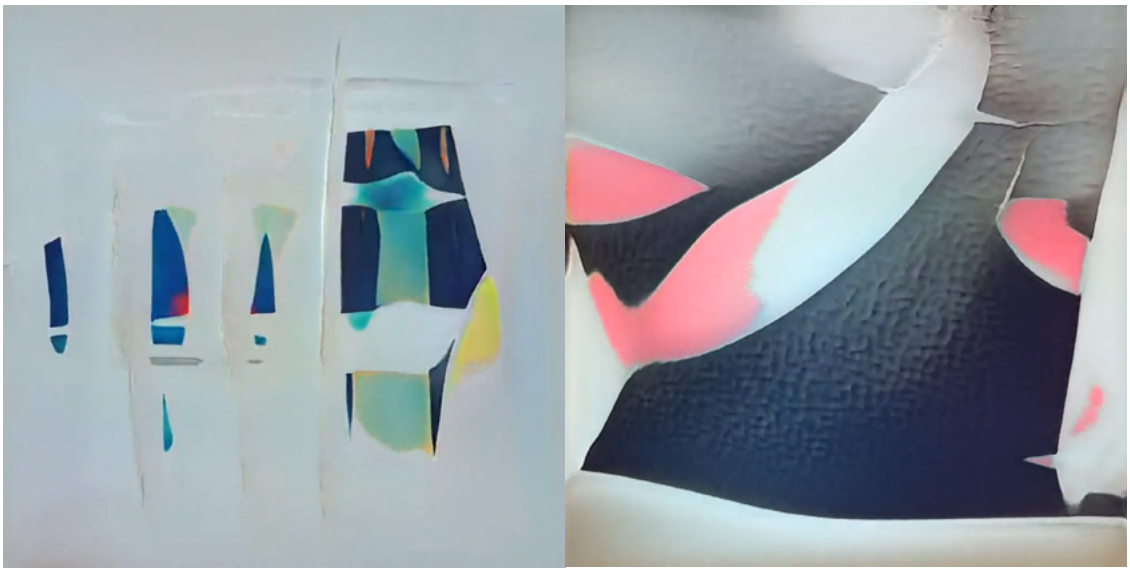

Fig. 6. (left) Example *Latent Space Traversal* using *Autolume-Live* (https://vimeo.com/670850199). (right) Example of a *Latent Space Interpolation* using *Autolume* (https://vimeo.com/670858023).

3.3.1 Frame Manipulation

By understanding the generation process, it is also possible to manipulate it. By going astray from the linear generation process and playing with the network's activations, noise and latent vector injections, it is possible to edit images increasing how much the media is manipulable, also called Network Bending (Broad, Leymarie, and Grierson 2021; Bau et al. 2018). The most basic version of this consists in applying affine transformations to the models activation. By doing so, we can translate, rotate, and zoom in on the image. In addition to applying transformations to layers 'activations, it is also possible to fuse features of two images by injecting the latent vector resulting in one image for the first resolution levels and the latent vector for the other in the rest of the layers. This creates a fusion of both images, where the vector used for the lower levels decides the structure of the image and the second the colours and textures (Brouwer 2020). Lastly, StyleGAN2-ada has a truncation value, which decides the sampling space from the latent space. A low truncation value reduces the diversity of the visual content, while a truncation value greater than 1 results in more abstract, expanded visual space. We use all these techniques to give the artist the potential for affecting and controlling the performance (Hook et al. 2011) and adjust the visuals to better suit the audio performances.

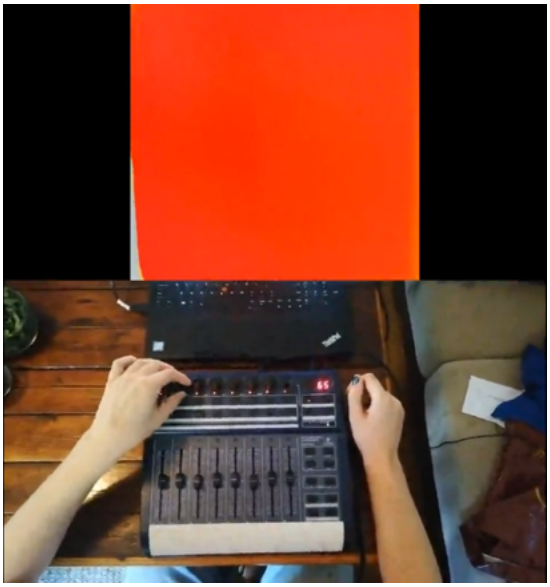

Fig. 7. Using the 8 knobs at the top of the MIDI-controller it is possible to apply Network Bending on the visuals generated by *Autolume-Live* (https://vimeo.com/670849859).

3.3.2 GANSpace

In addition to generating frames automatically, we want the user to be able to directly manipulate not just the rotation and positioning of the image, but the entire content . Although StyleGAN2(-ada) already disentangles the latent space, moving into random directions will not correspond to any semantics. We believe that this could interfere with how well the user can understand the interface. By finding semantically meaningful manipulations we improve the manipulability of the images, as the user can understand what the different directional manipulations mean. One way of finding directional vectors in the latent space that have interpretable meaning is *GANSpace (Härkönen et al. 2020).*

GANSpace identifies interpretable directions in the latent space using PCA. This method is lightweight and does not need any supervision (Shen et al. 2019). It finds the basis of the latent space in which the generated images have a high variance. We decided to use GANSpace as a preprocessing step instead of other approaches, because it has a low overhead, where other approaches introduce additional computational expenses during inference (Shen et al. 2019; Roich et al. 2021). GANSpace returns a list of vectors that can be plugged into a pretrained model without increasing inference time. Furthermore, this algorithm is model invariant, meaning that in future iterations the GAN architecture can be replaced and GANSpace could still be used to find interpretable manipulations on the images. In Figure 7 we showcase an example where the user can manipulate abstract images using the sliders of the VJ controller.

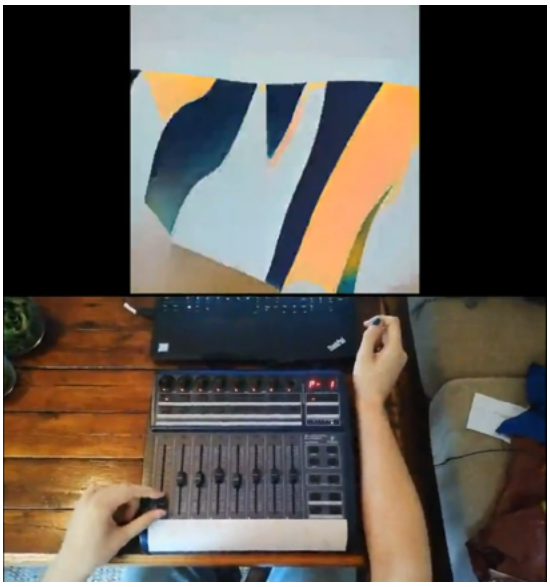

Fig. 8. Moving through the latent space using the faders of the MIDI-Controller. Every fader's values range between -1 and 1 weighting the first 8 directions found by GANSpace (https://vimeo.com/670850116).

3.3.3 Presets

It is normal for VJs to pre-record and save visuals that they want to use one or more times in a performance. Although we do not allow users to pre-record and save video clips, we allow the user to preset certain parameters of *Autolume-Live*. First of all, it is possible to predefine a set of latent vectors that are used by the *Latent Space Interpolation* and can be accessed during performances (Fig. 9). Then, it is also possible to predefine the content that can be generated. As *Autolume-Live* uses a StyleGAN2(-ada) Generator, the visuals that are generated are dependent on the data the model was trained on. Mostly these Generators synthesise images with only a few varying contents. Hence, to increase the visual variety, a set of models can be given to the software, all with their own predefined set of key-frames.

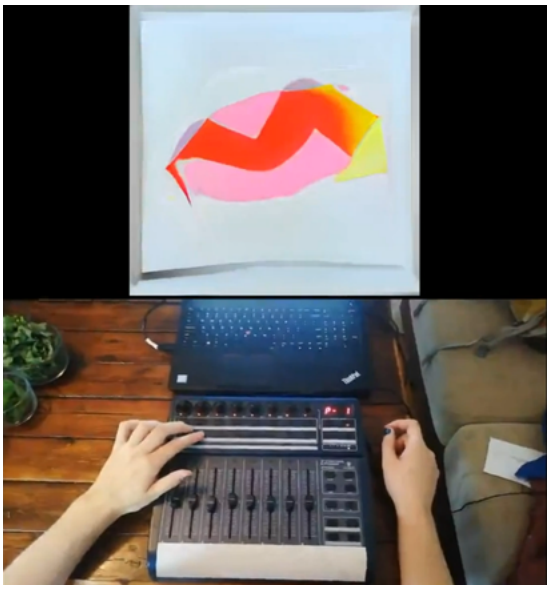

Fig. 9. Accessing key-frames using the Midi-Controller (https://vimeo.com/670850154).

3.3.4 VJ-Controller

We decide to link all these manipulations to a MIDI-Controller as this gives the user a visual representation of the current states and tangible feedback, which are both important interface features for VJs (Hook et al. 2011). The user can edit the current image according to salient directions, using the eight directional vectors we extracted with *GANSpace* to ever fader, where moving the fader up or down will set the coefficient for the weighted sum off all directional vectors that are added to the current latent vector. The knobs toggle further manipulations, i.e. rotation, translation, zoom and truncation value changes. By pressing the knobs the transformation is applied and the intensity can then be adjusted by turning the knob. Multiple transformations can be applied at the same time. We also linked the different audio-visual mappings to the controller. It is possible to switch between mappings pressing on one of the knobs. For both *Latent Space Interpolation* and *Latent Space Traversal* the same knob allows the user to add an additional multiplier to the step size, i.e. allows them to manipulate the sensitivity of the visuals reactions to the audio.

To access the different key-frames we use the array of buttons on the controller. On startup, every button is linked to a latent vector that can be predefined by the user or overwritten during the performance. These latent vectors are the positions the *Chroma-Based Interpolation* passes through. By pressing a button a position is loaded and the visualisation process proceeds from that key-frame. Furthermore, when multiple buttons are pressed at a time the average of all these seeds is used to create the visuals. Lastly, it is possible to iterate through the list of models by clicking the space bar of the computer that is running *Autolume-Live*.

4. Examples of Artworks made with Autolume-Live

Due to the 2020-2022 situation surrounding COVID-19, we were unable to use our system to accompany live performances. We have used different iterations of *Autolume-Live* to create two installations. We recorded some curated sessions and displayed them at the *Distopya sound art festival* in Istanbul 2021 (Dystopia Sound and Art Festival 2021) and *Light-Up Kelowna* 2022 (ARTSCO 2022). In both iterations, we let the audio mapping automatically generate the video without using any of the additional image manipulations. These installations show that the system on its own is already able to generate interesting and responsive visuals for a musical piece.

For the installation at the *Distopya sound art festival* we trained a StyleGAN2(-ada) model on abstract paintings and rendered a video using the described *Latent Space Traversal* mapping. For this particular piece we ran a super-resolution model on the final video as the original video output was in

512x512 and the wanted resolution was 4k. For our piece at *Light-Up Kelowna* we ran *Autolume-Live* with the *Latent Space Interpolation* mapping. The display included three urban screens, which allowed us to showcase three renders at the same time. We composed a video triptych using a dataset of figure drawings, a dataset of medical sketches and to tie the two videos together a model trained on a mixture of both datasets.

Fig. 10. Installations exhibited at *Distopya sound art festival* (left: https://vimeo.com/527564204) and *Light-Up Kelowna* (right: https://vault.sfu.ca/index.php/s/VBu760eFjNne1I6).

5. Future Work

Autolume-live is a prototype VJing tool using GANs. We are only using fundamental signal processing to extract audio features for the visualisation. This has already resulted in responsive visuals, but we hope that in the future these can be expanded on. Our Audio Module, we have coded in fundamental audio feature extractions and mapped them to features of the Synthesis Module. By replacing the Audio Module with a Neural Network it might be possible to connect its latent space to the GAN's latent spaces which could result in improving the expressiveness of the visuals. More high-level characteristics of the music could be extracted, such as the emotions and themes, which could then influence the content of the video. Especially text-to-image (Marinos Koutsomichalis 2021; Ramesh et al. 2021) and text-to-video (Wu et al. 2021) approaches could improve our framework. Text-to-image models could be used to more easily find latent vectors that the user wants to use in their performance. Instead of moving around the latent space using the sliders or doing prior work to find an image with certain features using CLIP based optimization, the user could simply describe the preset images they want to generate instead of using faders to move around the latent space (SOMNAI 2022; Murdock 2022).

In addition to richer audio-visual mappings, there are still improvements to be made to the system as a VJing tool through its Controller Module. An important aspect for VJs is the *interface reconfigurability*. Currently our system has a variety of transformations and editing possibilities linked to the MIDI-controller, but all these interactions are hard coded. In the future, we would want the user to be able to rearrange the MIDI-controller and also define their own transformations. Furthermore, the user should be able to freely define

their own audio-visual mappings, creating a completely modular interface. For that it might be interesting to incorporate our system into a preexisting VJing software to build on their preexisting audio feature extraction and visual manipulation algorithms.

When it comes to generative models trained on data it is always important to discuss what was used to train the model. The datasets allow the user to define the content of the visual performance. At the moment there are three options to aggregating images with varying advantages and disadvantages. The first and easiest way is to use datasets that have already been curated. While this approach is the least time consuming it opens up questions of copyright and also takes away from the novelty of the resulting pieces. Secondly, it is possible to collate images from artists under open licenses. This process takes longer, as the dataset has to be manually selected, but it allows for more control on the content and the data can be unique for the installation without the need to create every image. Lastly, the most time consuming approach is to create every image for the dataset. This removes the problems of copyright and allows for complete control over the data used to train the model. In our installations we followed the second approach, collecting images and editing them to create a cohesive dataset. We believe that in the future exploring the data as an additional tool for creative expression allows for new collaborations. For our upcoming pieces, we are collaborating with artists to create never seen before corpora to create completely novel experiences.

Lastly, we have only been able to use the system for private rehearsals and curated exhibitions due to the lack of live music performances during the COVID-19 pandemic. Once live events are more common we plan on using *Autolume-Live* to accompany live music performances at festivals. We also plan on inviting VJs to use our tool for further additions and changes to the tool to better suit live performances.

6. Conclusion

We sought out to create a live system for audio-reactive image generation and composed a tool for live visual synthesiser. *Autolume-Live* fulfills the requirements posed in a previous qualitative study on the expressive interactions VJs look for in their interfaces. Where previous GAN based music visualisers can only be deployed offline, our system can accompany live performances following three different audio mapping styles. In addition to automated video generation, we expanded our system as a VJing tool, to better encompass live performances. We allow the user to improvise and manipulate the medium live using a MIDI-controller. By deploying *Network Bending* and *GANSpace* the user has control over the visual performance in a responsive and understandable manner. Furthermore the user can store and call upon checkpoints and switch between generative models

on the fly allowing them to practice before performances and finetune the visual content to fit the current musical content.

While our system is able to accompany musical performances without user intervention, we believe that incorporating a MIDI-controller makes the system ascent in its usability. Increasing the intractability lets an artist collaborate and fine-tune the visual performance. With further advancements in making GANs more efficient and developments in explainable AI we believe that we will find new ways of manipulating the generation process, increasing the variety in the content that can be generated. By improving the inference time of GANs or incorporating different synthesis models, the tool will become more accessible to a wider range of users as currently a high performing GPU is needed to run the system in real time.

References

Alafriz, Mikael.
2021. "Lucid Sonic Dreams." Accessed February 8, 2022. https://github.com/mikaelalafriz/lucid-sonic-dreams

Bau, David, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba.
2018. "GAN Dissection: Visualizing and Understanding Generative Adversarial Networks." *arXiv [cs.CV].* arXiv. http://arxiv.org/abs/1811.10597

Blalock, Davis, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag.
2020. "What Is the State of Neural Network Pruning?" *arXiv [cs.LG].* arXiv. http://arxiv.org/abs/2003.03033

Broad, Terence, Frederic Fol Leymarie, and Mick Grierson.
2021. "Network Bending: Expressive Manipulation of Deep Generative Models." *arXiv [cs.CV].* arXiv. http://arxiv.org/abs/2005.12420

Brouwer, Hans.
2020. "Audio-Reactive Latent Interpolations with StyleGAN." In the 4th *Workshop on Machine Learning for Creativity and Design at NeurIPS 2020.* https://jcbrouwer.github.io/assets/audio-reactive-stylegan/paper.pdf

Derivative.
2022. "Touch Designer." Accessed February 8, 2022. https://derivative.ca/

Driedger, Jonathan, Meinard Müller, and Sascha Disch.
2014. "Extending Harmonic-Percussive Separation of Audio Signals." In the 15th International Society for Music Information Retrieval Conference (ISMIR 2014).

Dystopie Festival.
2021."Dystopia Sound and Art Festival." Accessed February 8, 2022. https://www.dystopie-festival.net/2021/l

Ellis, Dan.
2007. "Chroma Feature Analysis and Synthesis". April 21, 2007. https://www.ee.columbia.edu/~dpwe/resources/matlab/chroma-ansyn/

Fan, Jianyu, William Li, Jim Bizzocchi, and Philippe Pasquier.
2016. "DJ-MVP: An Automatic Music Video Producer." In the 13th *International Conference on Advanced in Computer Entertainment Technology.*

Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio.
2014. "Generative Adversarial Networks." *arXiv [stat.ML].* arXiv. http://arxiv.org/abs/1406.2661

Han, Song, Jeff Pool, John Tran, and William J. Dally.
2015. "Learning Both Weights and Connections for Efficient Neural Networks." *arXiv [cs.NE].* arXiv. http://arxiv.org/abs/1506.02626

Hardcoreanalhydrogen.
2018. "Music Video and Artificial Intelligence." February 21, 2018. https://hah-music.com/video-et-intelligence-artificielle/

Härkönen, Erik, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris.
2020. "GANSpace: Discovering Interpretable GAN Controls." In *Proc. NeurIPS.*

Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean.
2015. "Distilling the Knowledge in a Neural Network." *arXiv [stat.ML].* arXiv. http://arxiv.org/abs/1503.02531

Hook, Jonathan, David Green, John McCarthy, Stuart Taylor, Peter Wright, and Patrick Olivier.
2011. "A VJ Centered Exploration of Expressive Interaction." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1265–74.

Huang, Qiangui, Kevin Zhou, Suya You, and Ulrich Neumann.
2018. "Learning to Prune Filters in Convolutional Neural Networks." *arXiv [cs.CV].* arXiv. http://arxiv.org/abs/1801.07365

Jonathan, Fly.
2021. "Scatman John - Scatman (Wav2Lip Interpolation Video)." Accessed Feb 10. 2022. https://twitter.com/jonathanfly/status/1300976651380109313

Karras, Tero, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila.
2020. "Training Generative Adversarial Networks with Limited Data." *arXiv [cs.CV].* arXiv. http://arxiv.org/abs/2006.06676

Karras, Tero, Samuli Laine, and Timo Aila.
2018. "A Style-Based Generator Architecture for Generative Adversarial Networks." *arXiv [cs.NE].* arXiv. http://arxiv.org/abs/1812.04948

Light Up Kelowna.
2022. "Light Up Kelowna." Accessed February 8, 2022. https://fccs.ok.ubc.ca/about/events-workshops/light-up-kelowna/

Klingemann, Mario.
2018. "Automatically Generated BigGAN Music Video." Accessed February 8, 2022. https://www.youtube.com/watch?v=U7Jz17ZZyOg

_____.
2019. *Uncanny Mirror.* Accessed February 8, 2022. https://artsandculture.google.com/asset/mario-klingemann-uncanny-mirror-mario-klingemann/AQGCavNzwcOlEg?hl=en

Marinos Koutsomichalis, Alexia Achilleos.
2021. "Cyprus as AI Saw It: Digital Colonialism and AttnGAN Text to Image Synthesis." *xCoAx 2021 9th Conference on Computation, Communication, Aesthetics & X.*

Murdock, Ryan.
2022. "The Big Sleep." Accessed February 8, 2022. https://colab.research.google.com/github/levindabhi/CLIP-Notebooks/blob/main/The_Big_Sleep_BigGANxCLIP.ipynb

Obvious.
2018. *La Famille De Belamy.* Accessed February 8, 2022. https://obvious-art.com/la-famille-belamy/

Ramesh, Aditya, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever.
2021. "Zero-Shot Text-to-Image Generation." *arXiv [cs.CV].* arXiv. http://arxiv.org/abs/2102.12092

Resolume.
2022. "Resolume." Accessed February 8, 2022. https://resolume.com/

Roich, Daniel, Ron Mokady, Amit H. Bermano, and Daniel Cohen-Or.
2021. "Pivotal Tuning for Latent-Based Editing of Real Images." *arXiv [cs.CV].* arXiv. http://arxiv.org/abs/2106.05744

Shen, Yujun, Jinjin Gu, Xiaoou Tang, and Bolei Zhou.
2019. "Interpreting the Latent Space of GANs for Semantic Face Editing." *arXiv [cs.CV].* arXiv. http://arxiv.org/abs/1907.10786

Siegelman, Matt.
2019. "Deep Music Visualizer." Accessed February 8, 2022. https://github.com/msieg/deep-music-visualizer

SOMNAI.
2022. "Disco Diffusion v4.1." Accessed February 8, 2022. https://colab.research.google.com/drive/1sHfRn5Y0YKYKi1k-ifUSBFRNJ8_1sa39

Steenbrugge, Xander.
2020. "Experience Your Sound." Accessed February 8, 2022. https://wzrd.ai/

Wilson, Elizabeth, Shawn Lawson, Alex Mclean, and Jeremy Stewart.
2021. "Autonomous Creation of Musical Pattern from Types and Models in Live Coding." *Proceedings of the 9th Conference on Computation, Communication, Aesthetics & X.*

Wu, Chenfei, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan.
2021. "GODIVA: Generating Open-DomaIn Videos from nAtural Descriptions." *arXiv [cs.CV].* arXiv. http://arxiv.org/abs/2104.14806