POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Timbre Transfer and Timbre Interpolation Using a Conditional Convolutional beta-VAE Architecture

Laurea Magistrale in Music and Acoustic Engineering

Author: Silvio Pol

Advisor: Prof. Massimiliano Zanoni

Co-advisor: Luca Comanducci

Academic year: 2021-2022

1. Introduction

With the term "timbre", we refer to the perceptual qualities of a musical sound distinct from its amplitude and pitch. Modeling timbre is a hard task: it is a difficult job to define a physical or mathematical model of timbre since it is a perceptual and subjective characteristic of sound.
Most of state-of-the-art musical sound libraries used by studio composers are still obtained with high quality records of real instruments. However, building a music sound library with this methodology can be very a expensive and time consuming task. For this reason, there is in fact a substantial body of research in timbre modelling and synthesis. A particular sub-field of research is the one regarding timbre transfer.
Timbre Transfer techniques may find application in several different scenarios, particularly in music production environments. Having a tool that takes as input a signal of a recorded instrument and that gives as output the same recording but with a new timbre could be helpful to music producers. Different systems has been proposed in order to perform the timbre transfer task, most of them using generative Deep Learning models [3]. In this thesis, after giving an overview of the existing techniques and methodologies that pursue this goal, we propose a method which can effectively create a timbre space that permits to operate one to many timbre transfer. We do this by training a conditional convolutional beta-variational autoencoder architecture on a subset of the NSYNTH dataset [1] build by us. The system takes as input the module of the Short Time Fourier Transform of a note's signal with a given timbre, also known as spectrogram, and outputs multiple spectrograms of the same note with different timbres. It does that by constructing a navigable conditioned latent space representation of timbres and automatically encoding the pitch information. Given the possibility to move inside the latent space, we perform timbre interpolation, namely the morphing between two timbres, generating new samples that go from a starting timbre to an ending one, exploring the timbral space between them. We evaluate our system from different perspectives. In particular, we establish a twofold evaluation system based on classification and perceptual ratings. The experimental results show that the model is capable of performing the timbre transfer task having the generated samples that match the ground truth ones and that the conditioned latent space creates automatically

clusters based on timbre and pitch, giving the possibility to perform timbre interpolation by moving inside it.

The rest of this manuscript is structured as follows. In section 2, we provide the problem formulation, formalizing it. In section 3 we explain the technical details of the proposed method. In section 4 we describe the experimental setup along with an outline of our evaluation methods. In section 5 we examine the results: in the first place we give a visual inspection of the latent space, after that we present how the system perform in both timbre transfer and timbre interpolation. Finally, in section 5 we conclude our work.

2. Problem formulation

The problem we want to tackle in this thesis is twofold: firstly, we want to perform one-to-many timbre transfer from notes of a given timbre class and secondly we want to perform timbre interpolation between the notes generated performing the timbre transfer task, exploiting the properties of the latent space derived during the training phase of the timbre transfer task.

Let us first explicitly define the one-to-many timbre transfer problem. Given a discrete sound signal $\mathbf{x}^t$ characterized by a specific timbre class t, we calculate its time frequency representation $\mathbf{S}^t$.

$$\mathbf{S}^t = |STFT(\mathbf{x}^t)|. \tag{1}$$

Considering the network as a function f that takes $\mathbf{S}^t$ as input, the output will be:

$$\mathbf{S}^{t_1}, \ldots, \mathbf{S}^{t_n} = f(\mathbf{S}^t), \tag{2}$$

where the apexes $t, t_1, ..., t_n$ represent a discrete pre-defined set of timbre classes T, dataset and application dependent. These representations are transformed back using Griffin Lim Algorithm (GLA), ending up in a set of discrete audio signals:

$$\mathbf{x}^{t_1}, \ldots, \mathbf{x}^{t_n} = GLA(\mathbf{S}^{t_1}, \ldots, \mathbf{S}^{t_n}), \tag{3}$$

each one representing the audio signal $\mathbf{x}^t$ with only the timbral characteristic varied.

As anticipated, we also explore the possibility of performing interpolation between different timbres. While the interpolation task is not directly modeled during the training procedure, it is possible thanks to the latent space ordering enforced by the timbre-based conditioning. With interpolation we mean the generation of new notes that perform a gradual *morphing* between two given notes belonging to two different timbre classes: given a couple of generated signals $\mathbf{x}^{t_a}$ and $\mathbf{x}^{t_b}$ with $a, b \in T$ and $a \neq b$, the system will produce $m \in \mathbb{N}$ *timbre interpolations* between $\mathbf{x}^{t_a}$ and $\mathbf{x}^{t_b}$, with m adjustable. The first and the last of the m outputs will be, by convention, exactly $\mathbf{x}^{t_a}$ and $\mathbf{x}^{t_b}$ that can be defined as the starting point and the ending point of the interpolation.

3. Proposed Method

Figure 1 show the end to end representation of the system in the one-to-four timbre transfer scenario. In the following, we provide additional details about each of the three sub-structures of the system depicted in Figure 1.

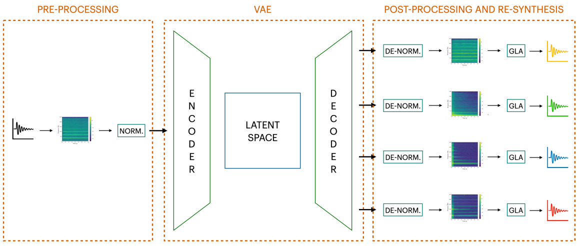


Figure 1: End to end representation of the system.

3.1. Pre-Processing

As time-frequency representation of the signals we opted for the log-Short Time Fourier Transform (log-STFT, whose result from now on will simply be called spectrogram) since it is the transform that better matches our needs, namely the possibility of high quality inversion and the fact that the frequency axis is not chroma-related, giving the same attention to all the frequencies arranged in a logarithmic scale. The spectrograms are then chopped along the time axis in order to match the input size expected by the network. Finally, we perform a 0-1 range min-max normalization for each set (train set, validation set and test set).

3.2. Network architecture

The architecture can be defined as a conditional, convolutional beta-variational autencoder. The encoder and the decoder have symmetrical architectures based on convolutional layers with ReLU activations followed each by a batch normalization layer. The network can be defined as a function f:

$$\mathbf{S}^c_{F\times T} = f(\mathbf{S}_{F\times T}, \mathbf{C}^c_{F\times T}, \mathbf{h}^c) \tag{4}$$

where $\mathbf{S}_{F\times T}$ is the input spectrogram with T the total number of time frames and F the total number of frequency bins, $\mathbf{C}^c_{F\times T}$ is the conditioning matrix concatenated on the channel axis with $\mathbf{S}_{F\times T}$, $\mathbf{h}^c$ is the one-hot vector concatenated with the input of the decoder and $\mathbf{S}^c_{F\times T}$ is the generated spectrogram with the new timbre. Each entry $\mathbf{C}^c_{f\times t}$ with $t = \{1,\ldots,T\}$ and $f = \{1,\ldots,F\}$ of the conditioning matrix has the same natural value $c \in \{0,\ldots,n-1\}$ associated to a specific timbral class and the vector $\mathbf{h}^c$ is the one-hot representation of that value.

3.3. Post-Processing

The first part of the post-processing phase is the de-normalization of the output of the network. We have to perform the inverse of process used on the normalization in the pre-processing phase. After that, we take the de-normalized spectrograms and render the frequency axis linear. The last step is the one of re-synthesis of the generated spectrograms. For this task, we use Griffin Limm Algorithm (GLA), an iterative method for phase estimation based on the spectrogram and re-synthesis.

3.4. Interpolation

Our architecture is built to map a spectrogram into a l-dimensional latent space. Each input signal is then represented by n points in the latent space called embeddings. Each of these embeddings is actually represented by two different points, namely the output of the `mu` layer and the output of the `log_var` layer, representing the bottleneck of the encoder architecture mentioned in section 3.2. It is possible for us to sample new points belonging into the latent space that aren't necessarily associated to the input of the networks and that will result in spectrograms that follow the distribution of the latent space itself.

In our work we perform timbre interpolation, meaning that we generated a series of new audio samples with a starting point and an ending point. Given the embeddings e_i^a and e_i^b with a, b belonging to the set of target timbres, constituting representations of the same input signal i, the interpolation function will yield to $m \in \mathbb{N}$ new embeddings, including the starting point e_i^a and the ending point e_i^b following the algorithm 1.

Algorithm 1 Interpolation of embeddings

```
$m$ = # of interpolations
$ratios$ = array of $m$ equally space float numbers $\in [0,1]$
$embeddings$ = []
$v1$ = starting embedding
$v1$ = ending embedding
for $i = 0$ to $m - 1$ do
    $v = (1.0 - ratio[i]) \times v1 + ratio[i] \times v2$
    append $v$ to $embeddings$
end for
return $embeddings$
```

4. Experimental Setup and Evaluation Methods

Each STFT consists in a 512 × 256 frequency-time representation of the associated audio signal. The resulting spectrogram covers 2.048s of the signal. We empirically found the best arrangement of parameters for our network with *learning rate* = 0.0001, *Adam* optimizer, *batch size* = 64, *latent dimension* = 64 and the parameter β has been set to 2. The hyper parameter $\beta \in \mathbb{R}$ is a value that weights the reconstruction loss L_R and the KL loss L_{KL} [2] so that the full loss function of the network can be formulated as:

$$LOSS = L_R + \beta \cdot L_{KL}. \tag{5}$$

We imposed an early stopping on the training procedure when the validation loss did not decrease for more than 20 epochs saving the model with the best validation loss. The network converged around epoch 396 and the duration of the training has been around 90 minutes.

4.1. Dataset

The dataset is a subset of NSYNTH dataset, a large-scale and high-quality dataset of annotated musical notes. The NSYNTH dataset contains a total of 305,979 musical notes, each with a unique pitch, timbre, and envelope. Each sample is four seconds long, monophonic, sampled at 16 kHz and is generated from one over 1,006 instruments taken from commercial sample libraries.

In our work we used a subset of the aforementioned dataset in order to perform the specific case of one-to-four timbre transfer. As input we used the *flute acoustic* class and as output the *string acoustic*, *keyboard acoustic*, *guitar acoustic* and *organ electronic* classes. A dataset split policy of 80-10-10 was used with 708 samples for training, 88 for validation and 88 for test for each timbral class. Indeed, the total number of samples in the train set is 2832, 352 for validation and 352 for test since we have 4 output timbral classes.

In the training phase the couples input-output spectrogram have the same pitch value enabling the automatic clustering of both timbre and pitch. In order to do that we built the dataset so it has the same number of samples for each pitch, that, in our case, can have a value that goes from 68 to 100.

4.2. Evaluation Methods

A twofold system evaluation method has been set that combines an objective assessment based on a timbral classifier and a perceptual one based on subjective ratings gathered with a questionnaire. The perceptual evaluation is needed since the timbre of a sound is primarily a perceptual characteristic that can be judged in its integrity only by a human response.

The classifier is trained on the same dataset defined in the previous section and is designed to classify the timbral class of a spectrogram belonging to one of the for classes *string*, *keyboard*, *guitar* and *organ*. It is used as a means both for the evaluation of the timbre transfer and timbre interpolation. In the former case we classify the outcomes of the network, namely the spectrograms generated in the inference phase carried out by test set. In the latter, we analyze the probability of belonging to the starting timbre or the ending one for each of the interpolation spectrograms.

Forty-three subjects, ranging from 24 to 36 years participated to the perceptual evaluation. The questionnaire was divided in two parts. In the first one, the subject is made to listen to 20 (5 strings, 5 keyboards, 5 guitars and 5 organs) couples of samples consisting each on a ground truth audio sample follow by 2 seconds of silence and a generated sample with the same pitch and belonging to the same timbral class. The subject is asked to rate the similarity of the two audio samples on a five point scale where the value 1 corresponds to "not similar" and 5 to "identical". In the second one, we perceptually evaluate timbre interpolation. The subject is made to listen to 12 triplets of generated samples. Each triplet consists of three generated samples: the first one representing the starting timbre, the second one the interpolation and the third one the target timbre. Each sample is spaced one second apart from the others. Since in our experiment we perform interpolation having $m = 5$ interpolation points, we ask the subject if the timbre of the sample in the middle (the second one) is more similar to the timbre of the first sample or to the timbre of the last one (the third) in a scale that ranges from 1 to 5 where 1 means "the timbre is identical to the one of the first sample" and 5 means "the timbre is identical to the one of the last sample".

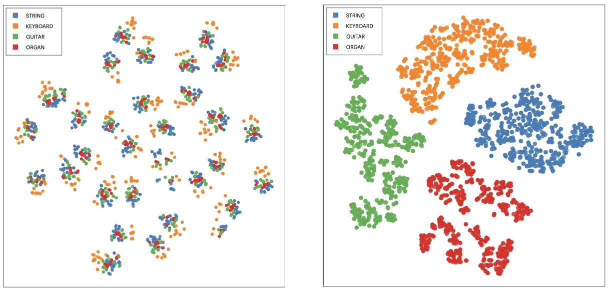
5. Results

In this section we expose the results obtained from the experimental setup just explained. We first present the latent space topology along with t-SNE graphics, then the timbre transfer and timbre interpolation tasks' performances are shown on both the classification and perceptual evaluation's methodologies.

5.1. Latent Space Topology

As a preliminary result, we bring a visual inspection of the latent space of the trained architecture. The latent space is actually bipartite: vae architectures [4] map the input data into a latent space consisting of two 1-D layers, representing the encoded normal distributions, so the encoder is trained to return the mean ($\boldsymbol{\mu}$) and the covariance ($\boldsymbol{\sigma}$) that describe these Gaussians.

To visualize the latent space, we used t-SNE dimensionality reduction on the 64-D $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ vectors representing the training set embeddings, respectively shown in Fig. 2a and Fig. 2b. As we can see, a peculiar form of clustering happens. In the t-SNE representation of the $\boldsymbol{\mu}$ latent vectors there are 33 clusters representing the 33 pitches ranging from 068 to 100 used in the training set. In the t-SNE representation of the $\boldsymbol{\sigma}$ latent vectors, we can clearly identify 4 clusters associated with the four timbre classes *string*, *keyboard*, *guitar* and *organ*. From this, we infer that the architecture perform automatically in the latent space a pitch clustering in the `mu` layer and a timbre clustering associated to the `log_variance` layer. This results permits a latent space navigation in both dimension, namely moving in a *timbre space* by sampling the log variance latent space but also moving in a *pitch space* by sampling the $\boldsymbol{\mu}$ latent space having conditioned only the timbre information in the training phase.



(a) $\boldsymbol{\mu}$ latent space

(b) $\boldsymbol{\sigma}$ latent space

Figure 2: t-SNE on `mu` layer and `log_variance` layer

5.2. Timbre Transfer

The first evaluation of the timbre transfer outcomes obtained in our experiment happens via classification. As explained in section 4.2, we trained a classifier to discern between the 4 target timbre classes used in the experiment. Figure 3 shows the results of the classification on the test set outcomes of the network in a form of confusion matrix. As we can see, we obtained perfect classification for the classes *string*, *keyboard* and guitar and a single misclassified sample for the *organ* class, interpreted as a *guitar* sample by the network.

The first part of the perceptual test is dedicated to the evaluation of timbre transfer. The results has been aggregated by timbral class, in order to inspect the quality of the audio reconstruction depending on the class. As we can see from the box-plots shown in Figure 4, we have quite uniform results, with the *keyboard* class reaching the best score. Since the *string* class is actually composed by recordings of different instruments (violins, violas, cellos) played with different techniques such as legato, détaché and staccato, the architecture has a harder time modeling the class, justifying the lowest score of the class. The white triangle represent the mean score obtained for each class.

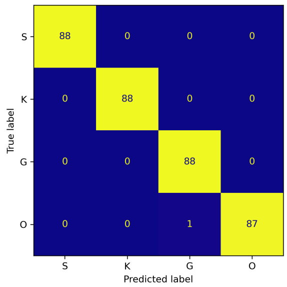


Figure 3: Confusion matrix on timbre transfer predictions.

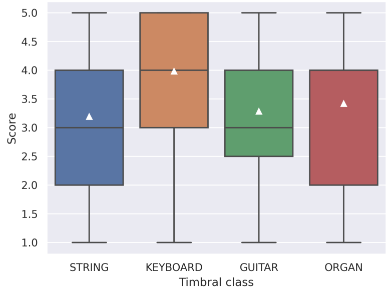
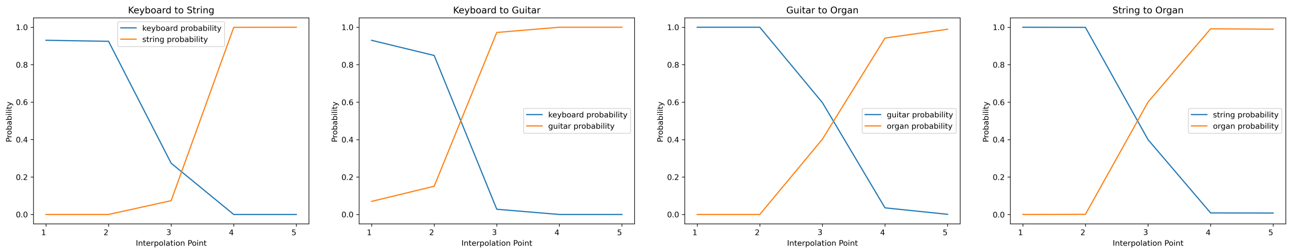


Figure 4: Perceptual ratings for timbre transfer task.

5.3. Timbre Interpolation

The first evaluation of the timbre interpolation is also obtained via classification. This time, we get the probability to belong to each one of the 4 timbral classes for each interpolation. With $m = 5$, we denominated the interpolation points as 1, 2, 3, 4 and 5 where the point 1 represents the starting point, point 5 the ending point and the points 2, 3, 4 the points in between in the latent space, calculated following the algorithm 1.

To visualize the results, given a certain interpolation that goes from timbre t_a to timbre t_b, we set up a graphic with the evolution of probabilities of the interpolation points (represented on the x-axis) to belong to the starting timbre t_a (blue line) and the ending timbre t_b (orange line). We show these plots for all the interpolations aggregated in Figure 5. From these plot we infer the fact that, while the point 3 is averagely



(a) Keyboard to String (b) Keyboard to Guitar (c) Guitar to Organ (d) String to Organ

Figure 5: Probability of interpolation points to belong to starting or ending timbre.

in the middle between the two classes, the points 2 and 4 are tended to be classified with a strong confidence as belonging respectively to the starting timbre class and to the ending one. This could be happening because of the low generalization capabilities of the classifier that, since it is trained on our dataset, does not have great potential with new unseen data.

The quality of the timbre interpolations has been evaluated in the second part of the perceptual test. In this case, the results has been aggregated for all points 2, 3 and 4 since points 1 and 5 are the actual outcomes of the timbre transfer task, already evaluated in section 5.2. Figure 6 shows the box plots with the results. The scores reveal that perceptually the interpolation works, having the interpolating points matching the scores given by the subjects. We can notice that the point three has a mean score slightly under 3, while point 2 and point 4 have values respectively near the starting timbre (score = 1) and the ending timbre (score = 5). A possible explanation for this fact is derived from the topology of the latent space: two points with different timbres are well separated in the latent space, as depicted in Figure 2b but the path going from one to the other could lead to a non-smooth evolution.

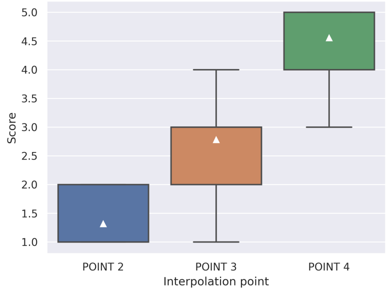


Figure 6: Perceptual ratings for timbre interpolation task.

6. Conclusions

In this work we presented a method to perform timbre transfer and timbre interpolation based on a conditional convolutional beta-VAE model. The results based on classification and perceptual evaluation show that the system is capable of performing both timbre transfer and interpolation. In addition we showed that the system is able to cluster in the latent space both the timbre and the pitch giving only the timbre information as condition. Further studies may focus on pitch conditioning or may extend the presented work on longer musical signals.

References

[1] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi. Neural audio synthesis of musical notes with wavenet autoencoders, 2017.

[2] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[3] Shulei Ji, Jing Luo, and Xinyu Yang. A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions. *arXiv preprint arXiv:2011.06801*, 2020.

[4] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.