A NOVEL AUDIO REPRESENTATION USING SPACE FILLING CURVES

Alessandro Mari

EPFL, 1015, Lausanne, Switzerland
alessandro.mari@epfl.ch

Arash Salarian*

Logitech Europe S.A., 1015, Lausanne, Switzerland
arash.salarian@ieee.org

ABSTRACT

Since convolutional neural networks (CNNs) have revolutionized the image processing field, they have been widely applied in the audio context. A common approach is to convert the one-dimensional audio signal time series to two-dimensional images using a time-frequency decomposition method. Also it is common to discard the phase information. In this paper, we propose to map one-dimensional audio waveforms to two-dimensional images using space filling curves (SFCs). These mappings do not compress the input signal, while preserving its local structure. Moreover, the mappings benefit from progress made in deep learning and the large collection of existing computer vision networks. We test eight SFCs on two keyword spotting problems. We show that the Z curve yields the best results due to its shift equivariance under convolution operations. Additionally, the Z curve produces comparable results to the widely used mel frequency cepstral coefficients across multiple CNNs.

Index Terms— Space filling curve, audio representation, deep learning, MFCC

1. INTRODUCTION

The first step when building an audio model is to choose the input space. Usually the data is pre-processed to obtain an intermediate low-level representation, which is subsequently used as model input. Most of these methods are frequency-based. For example, Tanweer et al. [1] used mel frequency cepstral coefficients (MFCCs) for speech recognition [2]. These coefficients are computed over a sliding window and stacked together to obtain a two-dimensional time-frequency image. During the computation, only the magnitude of the complex numbers is kept and the phase is discarded. This approach has shown some limitations in speech enhancement [3] and performing the Fourier transform adds extra computational costs.

Since the success of deep neural networks in image classification [4] and the arrival of dedicated hardware to train large models in parallel, deep neural networks have been widely applied to audio signals outperforming previous approaches [5]. A large variety of image networks have been proposed, such as convolutional neural networks (CNN, [6]). These can be used in combination with time-frequency images, although the two axes are semantically different from the horizontal and vertical axes of an image.

Separating the construction of an appropriate audio representation from the design of the model architecture might not be optimal for the task at hand. Hence some authors (e.g. [7, 8]) used raw audio waveform representations as inputs of one-dimensional CNNs. Although the additional cost of the pre-processing step is suppressed, connecting signals that are far apart, typically requires deeper architectures to increase the receptive field (RF).

In this study, we investigate a novel approach as an alternative to frequency domain based inputs and to raw audio waveforms. Space filling curves (SFC) enable us to map audio waveforms to two-dimensional images. In our approach, the input signal is not compressed and no information is lost since the SFCs act as bijective maps. By converting the audio samples to two-dimensional images, it is possible to leverage advanced deep neural networks from Computer Vision (CV). SFCs also reduce the distance between indices compared to one-dimensional indexing schemes, which guarantees the same receptive field with fewer layers. Finally, in a potential hardware implementation, the mapping is constant and does not need any runtime computations.

SFCs have been used together with CNNs in the past. For example, Yin et al. [9] have used the Hilbert curve [10] to combine consecutive representation of k-mers for the prediction of the chromatin state of a DNA sequence. Tsinganos et al. [11] have used the Hilbert curve to associate surface electromyography (sEMG) signals with hand gestures. In their experiments, the input image was obtained by mapping the time series into a two-dimensional image for each sEMG channel, which is similar to our technique. SFCs were also used for malware classification and detection [12, 13]. In their work the authors have mapped the code of the program, which can be viewed as a sequence of bytes, to pixels of an image. To the best of our knowledge, SFC mappings have not been used in the context of audio representation.

2. SPACE FILLING CURVES

In this paper, a SFC is a bijective mapping \( C_k : [N_k^2] \rightarrow [N_k] \times [N_k] \), where \( N_k = b^k \), \( [n] = \{1, \ldots, n\} \) and the SFC image representation of an audio sample \( s = \{s_i\}_{i=0}^L \) is obtained by \( I_k(s)(i,j) = s_{C_k^{-1}(i,j)} \). A SFC can be interpreted as an ordering of the pixels of an image, or as an indexing scheme for time series. Note that this definition differs from the usual one given by Peano [14], which refers to the limiting process \( (k \rightarrow \infty) \) after a proper normalization of the input and the output. We can distinguish two families of SFCs: recursive space filling curves (RSFC) and non-recursive space filling curves (NRSFC).
2.1. Recursive space filling curves

A RSFC is built recursively, where $C_{k+1}$ is obtained by subdividing the cell $(i, j)$ of the curve $C_k$ into $b^2$ cells and by modifying the curve $C_k$ according to a set of rules. A large number of RSFCs exist. In this paper, we focus on the following RSFCs: the Hilbert curve \cite{18}, the Z curve (also known as the Z-order \cite{15}), the Gray curve \cite{16}, the H curve \cite{17}, and a curve, that we will call OptR, proposed by Asano et al. \cite{18}. A representation of these curves with $k = 3$ can be found in Figure 1. Consecutive points $C_k(i), C_k(i + 1)$ are connected by a line. Dotted lines represent jumps.

\[ C_{k+1}(i, j) = \begin{cases} C_k(i, \lfloor\frac{j}{b}\rfloor) & \text{if } 0 \leq j < b^k \land 0 \leq i < b^k \land \frac{i}{b^k} - \frac{j}{b^{k+1}} \in [0, 1] \\ C_k(\lceil\frac{i}{b^k}\rceil, j) & \text{otherwise} \end{cases} \]

Fig. 1: Recursive space filling curves of order $k = 3$.

The Hilbert curve shows good performance in locality preservation, which is desirable in the context of audio samples: data points close in time should be close in the image representation in order to exploit the local nature of the convolution layers. Depending on the locality metric, the Hilbert curve outperforms the Z curve and the Gray curve \cite{19}, but is also inferior to the Z curve \cite{20}. Nonetheless, the Z curve and the Gray curve include jumps between consecutive indices, which don’t preserve locality. We included both of them to study the impact of this behavior on models.

Niedermeier et al. \cite{17} have shown that the H curve almost reaches the optimal lower bound among closed cyclic curves (i.e. $\|C_k(i) - C_k(i + 1)\|_\infty = 1, \forall i$). In particular, they have shown that $\|C_k(i) - C_k(j)\|_\infty, p = 1, 2, \infty$ roughly behaves like $\sqrt{|i - j|}$ in the worst case. This bound is smaller than $|i - j|$, which is the lower bound in the one-dimensional case achieved by the identity mapping. It shows the benefit from using two-dimensional inputs over one-dimensional inputs, as the distance between indices is reduced.

Note that the OptR curve is particularly complex, which might lead to poor performance, since regular patterns of the curve are harder to identify.

2.2. Non-recursive space filling curve

This category groups all curves that cannot be built recursively. We investigate three additional curves: the Scan curve, the Sweep curve, and the Diagonal curve \cite{21}. Figure 2 shows their representation when $k = 3$. The Sweep curve is a two-dimensional ordering scheme, where the sequence is cut into equally long intervals and stacked together to obtain an image. The Scan curve is obtained by reversing one interval out of two, which removes all jumps. Finally, the Diagonal curve is a 45 degrees rotated version of the Scan curve restricted to the $N_k \times N_k$ grid interior. These curves do not have particular locality preservation properties apart from being continuous mappings, i.e. $\|C_k(i) - C_k(i + 1)\|_\infty = 1$ (except for Sweep). Moreover, $\|C_k(i) - C_k(j)\|_\infty$ scales like $|i - j|$ in the worst case, which is much larger than $\sqrt{|i - j|}$.

\[ \text{Fig. 2: Non-recursive space filling curves of order } k = 3. \]

2.3. Pre-processing steps

We distinguish two pre-processing steps: functions $f \in \mathcal{P}$ that are applied on the raw audio signal and transformations $g \in \mathcal{T}$ that are performed on images. Assuming that the SFC mapping could be implemented in hardware, transformations in $\mathcal{T}$ should be preferred over functions in $\mathcal{P}$.

We might first center audio recordings in the middle of the frame. The procedure consists of first computing the weighted average energy in separate windows of size $w$ using a Gaussian kernel with standard deviation $\sigma$, and aligning the signal that is above the threshold $th$ with the middle of the output sequence. Note that center $\in \mathcal{P}$.

We might also use two additional data augmentations: mixup \cite{22} and shift. Mixup encourages the model to behave linearly in-between two training examples by taking random convex combinations. This operation is parametrized by $\lambda$, which follows a Beta distribution $\lambda \sim B(\alpha, \alpha)$, where $\alpha > 0$ is some hyperparameter. In the audio setting, this method can be applied either to the raw audio waveform or to the image representation. Since the SFC mapping is a linear operator, both approaches are equivalent. We choose to apply it to the image (i.e. mixup $\in \mathcal{T}$). Shift consists of applying random shifts $s_w$ from a uniform distribution $s_w \sim U(-L/4, L/4)$, where $L$ is the input length, to reduce the location dependency. This operation requires that the input is already centered in order to lose less information.

3. EXPERIMENTS AND RESULTS

3.1. Experimental setup

We used MFCC, one of the most common audio representations, as a baseline in this study\cite{1}. We focused on keyword spotting using two different datasets: Google Speech Commands V2 \cite{23} and a dataset based on the 1000 most frequent words of LibriSpeech’s “clean” split \cite{24}, for which Beckmann et al. \cite{24} have provided aligned labels\footnote{Code available at https://github.com/amari97/sfc-audio}. The Speech Commands dataset contains 105829 words classified into 35 classes, where 84483, 9981, 11005 of them were respectively used for training, validation, and test. Among the 1000 most frequent words in LibriSpeech, we selected 31 words that were appearing between 4000 and 8000. \footnote{Available at https://github.com/bepierre/SpeechVGG}
times in order to have a balanced dataset and spoken words of similar duration. We ended up with 164674, 2519, 2400 training/validation/test examples. All audio recordings were sampled at 16 kHz and had a duration of one second.

To properly compare both approaches, chosen models should process inputs of different sizes without changing their structure. This was achieved by introducing an average pooling layer with an area equal to the input size. Several CV networks using this layer were tested: the small MobileNetV3 network (Mo) [20], ShuffleNetV2 (Shuffle) [27], SqueezeNet (Squeeze) [28], MixNet (Mix) [29] and EfficientNet B0 (Eff) [30]. Additionally, the representations were evaluated using Res8 [31].

Our implementation of Squeeze accepted three channel images (RGB), we added a pointwise convolution to expand the input, followed by a fully connected layer. After the squeezing layer and the classifier was replaced by a BatchNorms activation layer and SqueezeNet normalization layer [32]. Our implementation of Squeeze added BatchNorms after the squeezing layer and the classifier was replaced by a fully connected layer. Finally, the pooling area of Res8 was extended to 4 × 4 to increase the RF, and the dilation of the convolutions was set to 1,1,2,2,2, following the increasing dilation strategy of the Res15 network [31]. As a result, the RF of Res8 was changed from 54 to 78.

Unless specified otherwise, all models were trained with stochastic gradient descent (SGD) with learning rate 0.5 and batch size 256 on 2 GPUs with distributed data parallel. We used an early stopping rule with 50 waiting epochs and we set to 300 the maximal number of epochs. We set the mixup hyperparameter α = 0.2, as in [22]. We set τh = 0.0001, such that only signals that were 1.6 higher than a flat audio signal exceeded it. We also set the Gaussian kernel parameters to w = 100 and σ = 25, such that boundary values had a weight close to zero. We used 40 MFCCs that were computed on frames of 0.025 seconds with overlaps of 0.015 seconds [31,33]. Finally, we used curves of order k = 7 to represent one second of audio, since 16000 < 16384 = 2^7 × 2^7.

3.2. Results

**Curve comparison** As shown in Table 1, the data augmentations improved the accuracy of the model, even if this effect was slightly less evident in the case of LibriSpeech. The benefits of these augmentations were larger for SFCs: the generated images were very different, improving the selection of invariant features. The Z curve yielded the best results in all situations and reached the baseline performance with Data Aug.

**Ablation study** As shown in Table 2, centering the audio clip inside the one second frame improved the accuracy for almost all curves. In particular, the improvement was larger for recursive curves. The Shift step provided a large gain by forcing the model to select features that are shift invariant in time. Mixup still improved the previous accuracy.

**Model comparison** Table 3 gives the results of the comparison between the Z curve and the MFCC approach. Except for Res8, there was no difference with the baseline. The best results were unexpectedly obtained with the largest network (Eff), but Res8 achieved impressive results in terms of efficiency and accuracy with MFCCs.

### Table 1: Test accuracy of each SFC on Mo. SGD is the baseline that only includes the SFC/MFCC computation. Data Aug. includes mixup, shift and center. *trained with an early stopping rule with 100 waiting epochs.

| Dataset       | Speech Commands | LibriSpeech |
|---------------|-----------------|-------------|
|               | SGD Data Aug.   | SGD Data Aug. |
| MFCC          | 89.3            | 97.9        | 98.2 |
| Z             | 80.1            | 89.0        | 94.9 | 94.1 |
| Z             | 86.2            | 92.8        | 97.4 | 98.3 |
| Gray          | 82.5            | 90.4        | 97.0 | 96.2 |
| H             | 82.8            | 91.6        | 96.0 | 97.8 |
| OptR          | 78.9            | 88.1        | 93.8 | 93.0 |
| Sweep         | 83.2            | 90.3        | 94.3 | 95.5 |
| Scan          | 83.8            | 90.3        | 94.7 | 96.1 |
| Diagonal      | 83.4            | 90.0        | 95.9 | 95.6 |

### Table 2: Test accuracy on Mo. SGD is the baseline that only includes the SFC/MFCC computation. *trained with an early stopping rule with 100 waiting epochs.

| Number of parameters | SGD*           | Data Aug.     |
|----------------------|----------------|---------------|
| Mo                   | 111K, 1'554K   | 1'289K        |
| Shuffle              | 2'653K         | 4'052K        |
| Mix                   | 128K           | 384K          |
| Eff                   | 740K           | 2'037K        |

**Receptive field influence** Figure 3 shows the average output probability p(s) of the true class as a function of time shifts s. The figure revealed that p(s) was lower when the audio signal was centered (s = 0), and when the RF was much smaller than the size of the input image (78 < 128). This effect was alleviated when increasing the RF to 125 by setting the dilation of the Res8 filters to 1,1,2,2,4,4. In this case, the model accuracy reached 86.5.

**Number of parameters** Figure 4 shows the model accuracy after changing the number of parameters by shrinking or expanding the width of the network (i.e. the number of channels of each convolution layer) by a factor width_mult. The large gap with the baseline persisted on Res8. Moreover, the difference remained similar when reducing the number of parameters. Due to its prohibitively large computational cost, the standard deviations were only computed for Mo using cross-validation, but they showed that the gap with the baseline was not significant.

**Curve Comparison on Res8** Table 4 shows the performance of each curve with Res8. Non-recursive curves yielded the best results, while the Z curve outperformed the other recursive curves. The gap with the baseline was significant.
we have $I_k(s)(i,j) = s_{a_m b_m \ldots a_0 b_0}$. Shifting the output sequence by $d_2 \equiv d_2 \ldots d_2 0 \ldots 0$ doesn’t modify the digits $a_1 \ldots a_0, b_1 \ldots b_0$. Therefore $I_k(s')(i,j) = I_k(s)(i,j) = s_{a_m b_m \ldots a_0 b_0} = s_{a_m b_m \ldots a_1 b_1 \ldots a_0 b_0}$. Similarly, using a stride of $2'$ only modifies digits that are after the 2l-th index. Let’s consider the case $i = 0, j = 0$. Then $O_l \circ I_k(s')(0,0) = O_l \circ I_k(s)(d_{2k-1} \ldots d_{2l+1}, d_{2k-2} \ldots d_{2l})$. Unfolding the image finally yields $I_{k-1} \circ O_l \circ I_k(s)(d_{2k-1} \ldots d_{2l+1}, d_{2k-2} \ldots d_{2l+1}) = I_{k-1} \circ (O_l \circ I_k(s)(0,0)) = I_{k-1} \circ O_l \circ I_k(s)(0)$, which proves the lemma for $i = 0, j = 0$. For the general case the result follows because we have $O_l \circ I_k(s'(i,j)) = O_l \circ I_k(s)(a_m \ldots a_1, b_m \ldots b_1)$.

On the other hand, curves like the Hilbert curve (Figure 1) include some rotation of its elementary block, which destroys the equivariance property. This intuition is further assessed by the fact that the Z curve performed well without imposing some shift invariance with data augmentations (Table 1), which shows that the model has already built a coherent feature extraction.

Table 3 shows that the SFC approach was competitive with the baseline when trained with Data Aug., except for the Res8 network, for which the large jump in the middle of the Z mapping had an influence on the accuracy (Figure 3). Increasing the RF of the network improved the accuracy from 85.3 to 86.5, but remained substantially smaller than the baseline (86.5 ≪ 94.0). The number of parameters was not sufficient to justify the discrepancy observed in Table 3 since the gap persisted on Figure 3. It might be related to the network architecture, which was specifically designed for MFCC inputs and which contained a small number of layers. Indeed Res8 worked best with non-recursive curves (Table 4), which have an image structure similar to the MFCC representation: the image can be decomposed into two perpendicular axes — a time axis and a feature axis — which is not possible for the recursive curves (see Figure 7).

4. CONCLUSION

We have proposed an alternative audio representation to frequency-based images and raw audio waveforms using space filling curves. We have shown that it achieves comparable performance to the widely used MFCC representation when combined with deep CNNs in the context of keyword spotting tasks. In particular, the Z curve yields the best results, which is probably due to its shift equivariance under convolution operations. Our study suggest that to leverage DNNs time-frequency decomposition should not be considered as a central dogma and simpler one-dimensional to two-dimensional mappings such SFCs might perform just as well. Future work could aim at checking the robustness of the image representation under noisy inputs and at generalizing the method to variable input lengths.
5. REFERENCES

[1] S. Tanweer, A. Mobin, and A. Alam, “Analysis of Combined Use of NN and MFCC for Speech Recognition,” International Journal of Computer and Information Engineering, vol. 8, pp. 1736–1739, 2015.

[2] European Telecommunications Standards Institute, “Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms,” Tech. Rep. ES-201-108, v1.1.3, ETSI, 2003.

[3] K. Paliwal, K. Wójcicki, and B. Shannon, “The importance of phase in speech enhancement,” Speech Communication, vol. 53, no. 4, pp. 465–494, 2011.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in Proc. NIPS, 2012, vol. 1, pp. 1097–1105.

[5] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal processing magazine, vol. 29, no. 6, pp. 82–97, 2012.

[6] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation Applied to Handwritten Zip Code Recognition,” Neural Computation, vol. 1, no. 4, pp. 541–551, 1989.

[7] J. Lee, T. Kim, J. Park, and J. Nam, “Raw Waveform-based Audio Classification Using Sample-level CNN Architectures,” arXiv preprint arXiv:1712.00866, 2017.

[8] P. Ghahremani, V. Manohar, D. Povey, and S. Khudanpur, “Acoustic Modelling from the Signal Domain Using CNNs,” in Proc. Interspeech, 2016, pp. 3434–3438.

[9] B. Yin, M. Balvert, D. Zambrano, A. Schoenhuth, and S. Bohte, “An image representation based convolutional network for DNA classification,” in Proc. ICLR, 2018.

[10] D. Hilbert, Über die stetige Abbildung einer Linie auf ein Flächenstück, pp. 1–2, Springer, Berlin, Heidelberg, 1935.

[11] P. Tsinganos, B. Cornelis, J. Cornelis, B. Jansen, and A. Skodras, “A hilbert curve based representation of semg signals for gesture recognition,” in Proc. IWSSIP, 2019, pp. 201–206.

[12] Z. Ren, G. Chen, and W. Lu, “Malware visualization methods based on deep convolution neural networks,” Multimedia Tools and Applications, pp. 1–19, 2019.

[13] S. O’Shaughnessy, “Image-based Malware Classification: A Space Filling Curve Approach,” in IEEE Symposium on Visualization for Cyber Security, 2019, pp. 1–10.

[14] G. Peano, “Sur une courbe, qui remplit toute une aire plane,” Mathematische Annalen, vol. 36, no. 1, pp. 157–160, 1890.

[15] G. M. Morton, “A computer oriented geodetic data base and a new technique in file sequencing,” Tech. Rep., IBM Ltd., Ottawa, 1966.

[16] C. Faloutsos, “Multiattribute Hashing Using Gray Codes,” in Proc. SIGMOD, 1986, pp. 227–238.

[17] R. Niedermeier, K. Reinhardt, and P. Sanders, “Towards optimal locality in mesh-indexings,” Discrete Applied Mathematics, vol. 117, no. 1, pp. 211–237, 2002.

[18] T. Asano, D. Ranjan, T. Roos, E. Welzl, and P. Widmayer, “Space-filling curves and their use in the design of geometric data structures,” Theoretical Computer Science, vol. 181, no. 1, pp. 3–15, 1997.

[19] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, “Analysis of the clustering properties of the Hilbert space-filling curve,” IEEE Transactions on Knowledge and Data Engineering, vol. 13, no. 1, pp. 124–141, 2001.

[20] H. K. Dai and H. C. Su, “On the Locality Properties of Space-Filling Curves,” in International Symposium on Algorithms and Computation, 2003, pp. 385–394, Springer, Berlin, Heidelberg.

[21] M. F. Mokbel and W. G. Aref, “Space-filling curves,” in Encyclopedia of GIS, S. Shekhar and H. Xiong, Eds., pp. 1068–1072. Springer US, Boston, MA, 2008.

[22] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond Empirical Risk Minimization,” in Proc. ICLR, 2018.

[23] P. Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” arXiv preprint arXiv:1804.03209, 2018.

[24] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in Proc. ICASSP, 2015, pp. 5206–5210.

[25] P. Beckmann, M. Kegler, and M. Cernak, “Word-level embeddings for cross-task transfer learning in speech processing,” EUSIPCO (Submitted), 2021.

[26] A. Howard, M. Sandler, G. Chu, L. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, “Searching for MobileNetV3,” in Proc. ICCV, 2019, pp. 1314–1324.

[27] N. Ma, X. Zhang, H. Zheng, and J. Sun, “ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design,” in Proc. ECCV, 2018, pp. 122–138.

[28] F. Iandola, S. Han, M. Moskewicz, K. Ashraf, W. Dally, and K. Keutzer, “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size,” arXiv preprint arXiv:1602.07360, 2016.

[29] M. Tan and Q. V. Le, “MixConv: Mixed Depthwise Convolutional Kernels,” in BMVC, 2019, p. 74.

[30] M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in Proc. ICML, 2019, vol. 97, pp. 6105–6114.

[31] R. Tang and J. Lin, “Deep Residual Learning for Small-Footprint Keyword Spotting,” in Proc. ICASSP, 2018, pp. 5484–5488.

[32] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in Proc. ICML, 2015, vol. 37, pp. 448–456.

[33] N. Ryan, M. Slaney, M. Liberman, E. Shriberg, and J. Yuan, “Highly accurate mandarin tone classification in the absence of pitch information,” Proc. International Conference on Speech Prosody, pp. 673–677, 2014.