

AUDIO TRANSPORT: A GENERALIZED PORTAMENTO VIA OPTIMAL TRANSPORT

Trevor Henderson

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA USA
tfh@mit.edu

Justin Solomon

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA USA
jsolomon@mit.edu

ABSTRACT

This paper proposes a new method to interpolate between two audio signals. As an interpolation parameter is changed, the pitches in one signal slide to the pitches in the other, producing a portamento, or musical glide. The assignment of pitches in one sound to pitches in the other is accomplished by solving a 1-dimensional optimal transport problem. In addition, we introduce several techniques that preserve the audio fidelity over this highly nonlinear transformation.

A portamento is a natural way for a musician to transition between notes, but traditionally it has only been possible for instruments with a continuously variable pitch like the human voice or the violin. Audio transport extends the portamento to *any* instrument, even polyphonic ones. Moreover, the effect can be used to transition between different instruments, groups of instruments, or any other pair of audio signals. The audio transport effect operates in real-time; we provide an open-source implementation. In experiments with sinusoidal inputs, the interpolating effect is indistinguishable from ideal sine sweeps. More generally, the effect produces clear, musical results for a wide variety of inputs.

1. INTRODUCTION

A portamento, or musical glide, has been a significant expressive device in music for at least the past 200 years [1, 2]. Short portamenti can connect notes to make a passage sound more fluid, while long portamenti can draw out a transition with anticipation before finally arriving at the destination. The author in [1] claims that “portamento draws on innate emotional responses to human sound, as well as on our earliest memories of secure, loving communication, in order to bring to performances a sense of comfort, sincerity, and deep emotion.” Regardless of whether this text describes a universal experience, portamenti have a decidedly unique sound and musical significance.

Due to the nature of the sound, the only instruments that can produce portamenti are instruments that, like the human voice, can vary their pitch continuously. Certain electronic systems described in §1.1 are capable of producing the effect, but they are limited to particular situations (e.g. monophonic glide, offline processing). In this work, we present an audio effect titled, “audio transport,” which interpolates between *any* two audio streams in a way that sounds like a portamento, automatically and in real-time.

The audio transport effect relies on solving a 1-dimensional optimal transport problem. The solution to this problem determines how the pitches in one signal will move to pitches in the other. Copyright: © 2019 Trevor Henderson et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

other. We find that the effect works best on pairs of sounds that do not have sharp attacks or strong tremolo and have comparable brightness.

The paper is organized as follows. §2 gives a brief introduction to the optimal transport problem and its relevance to the rest of the paper. §3 presents the audio transport effect, including a number of techniques necessary to produce artifact-free audio. §4 details our implementation of the audio transport effect and provides perceptual results. Finally, §5 concludes with discussion of potential applications and future work.

1.1. Previous Work

Portamenti have existed in electronic music since its inception. One of the earliest electronic instruments, the theremin, is famed for the sweeping sounds it can produce from its continuous pitch control. Today, a pitch wheel can be found on almost all synthesizers as a way to bend a note’s pitch.

In addition to manually-controlled portamenti, many synthesizers have a “glide” parameter which automatically introduces portamenti between sequential notes. Typically this effect is monophonic, but some synthesizers support polyphonic glide using rule based systems [3].

As for sample-based instruments, the pitch of a sample can be changed by varying its playback speed. Alternatively, phase vocoders allow for a sample’s pitch to be changed independently of its speed [4, 5]. Both of these methods, like a pitch wheel, can produce a polyphonic portamento but they necessarily move all the pitches in the same direction at the same rate. As such, these techniques can not be used to slide between chords with different harmonies or instruments with different timbre.

Techniques involving phase vocoders [6], modulation vocoders [7], and popular but unpublished commercial products like Melodyne [8] allow for artists to vary pitches within a sample independently, which could conceivably be used to create polyphonic portamenti. This type of pitch manipulation, however, is not suited for real-time use because without manual input, the pitches have no destination.

While not related to portamenti, optimal transport has been applied to audio problems before. The authors in [9] describe how optimal transport can be used to perform spectral unmixing with application to musical transcription. The authors in [10] apply optimal transport to the problem of fundamental pitch estimation. Both of these papers focus on analysis rather than synthesis.

1.2. Contributions

We present audio transport, an audio effect that produces a portamento between arbitrary audio sources. The effect works by interpolating between the spectra of the two input signals according

to an optimal transport map. To our knowledge, this is the first work to apply optimal transport to audio generation and also the first that can achieve this type of portamento effect automatically and in real-time. In addition to the novel application of optimal transport, we present a technique based on time-frequency reassignment [11] that divides the audio spectrum prior to transport and we extend the phase accumulation technique from [5] to prevent phasing between windows.

2. OPTIMAL TRANSPORT OVERVIEW

The optimal transport problem asks how to move probability mass from one configuration to another in a way that minimizes the amount of work (mass times distance) performed on each infinitesimal piece of mass. More formally [12], the problem seeks an optimal plan $\pi^*(x, y)$ that describes how much mass should be transferred from position x to y satisfying:

$$\pi^* = \arg \min_{\pi} \iint_{\mathbb{R}^2} \|x - y\|^p d\pi(x, y), \quad (1)$$

subject to nonnegativity as well as conservation of mass for source and target distributions ρ_v and ρ_w :

$$\int_{\mathbb{R}} \pi(x, y) dy = \rho_v(x) \quad \text{and} \quad \int_{\mathbb{R}} \pi(x, y) dx = \rho_w(y). \quad (2)$$

The p -th root of the optimal value provides an intuitive way to measure the similarity between two distributions known as the p -Wasserstein distance. In the rest of this paper, we will use $p = 2$. The corresponding “least squares” Wasserstein distance satisfies all metric axioms among other attractive properties [13, 12].

We use the optimal plan to perform *displacement interpolation* between two distributions [14]. This interpolation animates the mass assignment computed in Equation (1) by sliding each particle of mass between its two assignments. In computer graphics, this interpolation technique can be used to naturally transition between histograms, images, or meshes [15, 13, 16, 17, 18].

Consider Figure 1, which demonstrates two different ways to interpolate between distributions. On top, the distributions are interpolated linearly. If we imagine the distributions as audio spectra, then this transformation is simply fading one set of pitches out and another set in. On the bottom, the same distributions are transformed using displacement interpolation. The mass physically slides from one location to another. If these were audio spectra, this sliding would sound like a portamento.

It should be noted that solving the optimal transport problem is known to be computationally challenging for any dimension $d > 1$. Fortunately, solving the problem on the real line can be done in linear time [18].

3. AUDIO TRANSPORT

The audio transport effect works by performing displacement interpolation on input audio spectra, so that pitches in one signal slide to pitches in the other as an interpolation parameter is changed. To modify the spectra over time, the audio transport algorithm follows the phase-vocoder paradigm [4, 5, 6]. In detail, a sliding short-time Fourier transform (STFT) is applied to both input audio streams, producing complex spectra. These spectra are interpolated according to the optimal transport map and fed through an inverse STFT to form the output audio stream.

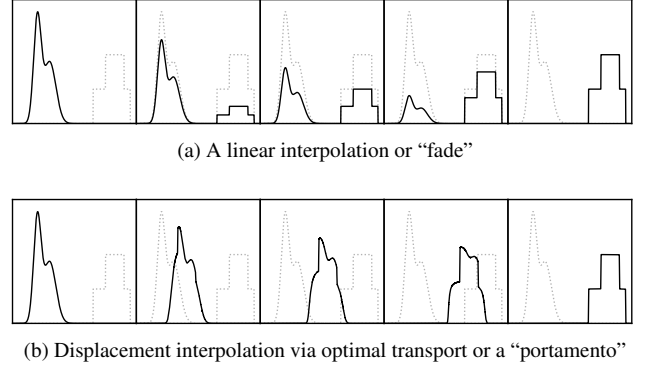


Figure 1: The distribution on the left is transformed into the distribution on the right with two different interpolation methods.

§3.1 describes an efficient way to interpolate between spectra using optimal transport. Alone, this method produces two artifacts which, borrowing from phase-vocoder literature, are known as *vertical incoherence* and *horizontal incoherence* [5]. Solutions to these two phenomena are described in §3.2 and §3.3, respectively.

3.1. Optimal Transport Between Spectra

Consider discrete spectra represented by complex vectors X, Y and corresponding frequency vectors ω^X, ω^Y . Analogously to the continuous optimal transport plan given in Equation (1), we can write the optimal transport plan between these discrete spectra as the plan $\pi^* \in \mathbb{R}^{|X| \times |Y|}$ minimizing:

$$\pi^* = \arg \min_{\pi \geq 0} \sum_{i,j} \left| \omega_i^X - \omega_j^Y \right|^2 \pi_{ij} \quad (3)$$

subject to the conservation of mass constraint

$$\sum_j \pi_{ij} = |X_i| \quad \text{and} \quad \sum_i \pi_{ij} = |Y_j|. \quad (4)$$

This problem assumes that $\sum_i |X_i| = \sum_j |Y_j|$. To treat spectra with different total magnitudes, the plan can be computed on normalized spectra; then, scaling is interpolated linearly over the interpolation.

Once an optimal plan is computed, the spectra can be interpolated with parameter $k \in [0, 1]$ by placing each mass π_{ij}^* at the displaced frequency:

$$(1 - k)\omega_i^X + k\omega_j^Y \quad (5)$$

If multiple masses are placed at the same frequency, they are added together. The phase attributed to the mass is considered in §3.3.

In one dimension, the optimal transport plan is monotone or, in other words, no mass crosses over any other mass [19]. This allows for Equations (3) and (4) to be solved using the greedy strategy presented in Algorithm 1.

The algorithm begins with the initial bins of the two spectra. Since no mass can cross over any other, all of the mass in the smaller bin must be assigned to the larger. With this assignment done, one can imagine virtually removing the smaller bin and shrinking the mass of the larger by the mass assignment. The algorithm then continues inductively on the smaller problem. At every

Algorithm 1 Computing The Optimal Transport Matrix, π^*

```

 $\pi_{i,j}^* \leftarrow 0$ 
 $\rho_X, \rho_Y \leftarrow |X_0|, |Y_0|$   $\triangleright \rho$  is the mass left in a bin

loop
  if  $\rho_X < \rho_Y$  then
     $\pi_{i,j}^* \leftarrow \rho_X$   $\triangleright$  Assign as much mass as possible

     $i \leftarrow i + 1$   $\triangleright$  Refill the emptied bin
    if  $i \geq |X|$  then break
     $\rho_X \leftarrow |X_i|$ 

     $\rho_Y \leftarrow \rho_Y - \rho_X$   $\triangleright$  Decrease the capacity of the other

  else
    Symmetric to the case above

return  $\pi^*$ 

```

iteration, all of the mass in one bin becomes completely assigned. Therefore, the complexity of the algorithm is $O(|X| + |Y|)$. This runtime is efficient relative to the super-linear runtime of the fast Fourier transform.

3.2. Resolving Vertical Incoherence: Slicing the Spectrogram

One unfortunate effect of using an STFT is the necessary trade-off between time and frequency resolution. As the time resolution increases, the frequency domain becomes “smeared.”

The relation between a peak frequency and its smeared components is known to be important for perceptual quality. Treating these independently leads to phasing artifacts within a window known as vertical incoherence [5]. One method to solve this problem in phase vocoder literature is to “lock” regions surrounding a peak frequency so that the relative phase between bins within these regions remains unchanged [20, 5].

If Algorithm 1 were applied directly to audio spectra, it would introduce vertical incoherence by translating smeared components independently. So, applying the locking strategy, we will treat smeared regions as single units with collective magnitude in the transportation map.

It now remains to determine how exactly to choose the boundaries between smeared spectral regions. A common strategy is to use a heuristic to find local peaks and then assign the boundaries to be the midpoints of the peaks. Since displacement interpolation makes extreme changes to the spectra, however, this somewhat naïve plan [5, 6] is not sufficiently robust to produce a clean signal. We propose a more principled segmentation method based on frequency reassignment.

Frequency reassignment uses information in a signal’s phase to enhance its frequency resolution. Each spectral component with frequency ω_i is mapped to the reassigned frequency $\hat{\omega}_i$ that better reflects the true energy distribution [11]. Sinusoids that have been smeared across multiple bins become mapped to the same central frequency, which produces the plateaus shown in Figure 2.

With this view, an intuitive way to define sinusoidal regions is by the zero crossings of $\hat{\omega}_i - \omega_i$. Falling crossings indicate the center bin of a region while rising crossings indicate the boundaries. These can be computed at the cost of an additional STFT

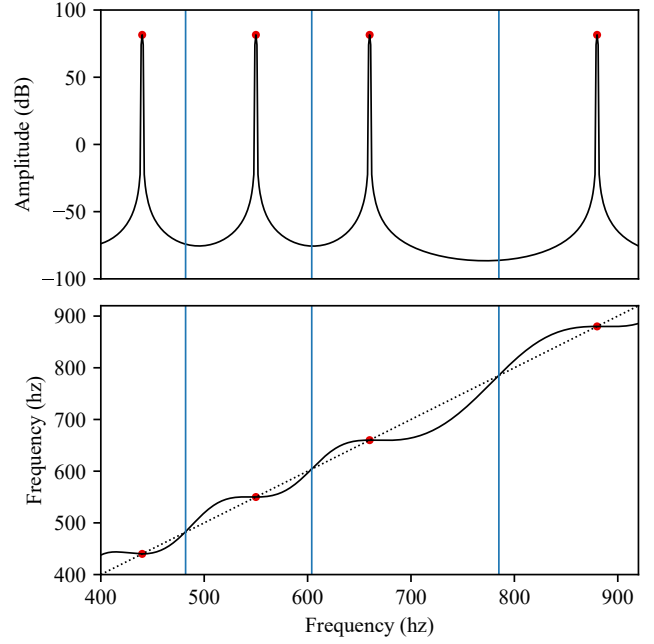


Figure 2: Dividing the spectrum of a sinusoidal A major chord consisting of the notes A_4 , $C\sharp_5$, E_5 and A_5 . The spectrum is displayed on top. On the bottom, the reassigned frequency $\hat{\omega}$ (solid line) is plotted against the frequency ω (dashed line). The intersections of these lines indicate the boundaries between groups (vertical lines) and their pitch centers (dots).

with the following formula [11]:

$$\hat{\omega}_i - \omega_i = \Im \left\{ \frac{X_i^{T^h} \cdot X_i^*}{|X_i|^2} \right\}. \quad (6)$$

X^{T^h} is the STFT computed using a time-weighted analysis window.

3.3. Resolving Horizontal Incoherence: Phase Accumulation

Finally, we reintroduce phase to the spectra. In doing so, we will be concerned with the phase relations between consecutive windows rather than the phase relations within a window. Inter-window phase relations carry information about short-time events like transients and hence ignoring these relations can create a blurry sound in some cases as discussed in § 4.2.

When a particular spectral region is transposed, its phase rotates at a different rate. Thus, applying the phases of consecutive windows in the original signal to the corresponding windows of the transposed signal causes interference known as horizontal incoherence [5].

In phase vocoders, this is resolved by integrating the reassigned frequency over the window difference. In other words, the phase φ_i^t in bin i and window t can be estimated from the phase φ_i^{t-1} in window $t - 1$ as follows:

$$\varphi_i^t = \varphi_i^{t-1} + \hat{\omega}_i^{t-1} \cdot \Delta, \quad (7)$$

where Δ is the delay between the windows in seconds. This update is applied to *center* bin of a region as described in §3.2. The other

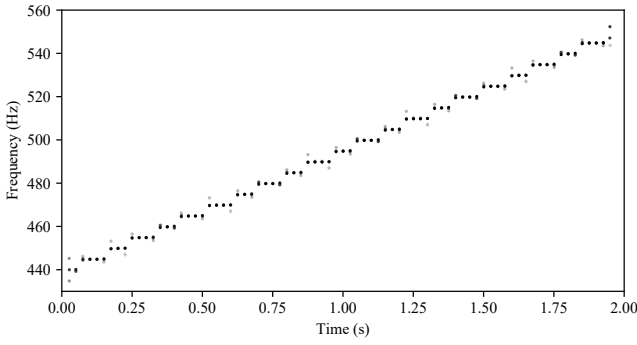


Figure 3: A (reassigned [11]) spectrogram of the audio transport effect being used to interpolate between sinusoids at A_4 and $C_{\sharp 5}$. The “stair-stepping” effect is due to the frequency resolution of ≈ 5 Hz, but this can be reduced arbitrarily by padding each window with additional zeros.

phases in each region are modified accordingly to maintain the same relative phase with respect to the center bin.

Some small modifications must be made to apply Equation (7) to the audio transport effect. First of all, it is possible that many spectral regions overlap on a particular bin, which makes the terms $\hat{\omega}_i^{t-1}$ and φ_i^{t-1} ambiguous. To resolve this we simply choose the frequency and phase of the loudest overlapping region. Additionally, since the audio transport effect consists of rapidly-moving pitches, we can minimize phasing by averaging the current reassigned frequency and the previous reassigned frequency:

$$\varphi_i^t = \varphi_i^{t-1} + \frac{\hat{\omega}_i^t + \hat{\omega}_i^{t-1}}{2} \cdot \Delta. \quad (8)$$

4. RESULTS

We implemented the audio transport effect described in §3 for real-time audio interpolation. We tested our implementation on synthetic sounds described in §4.1 as well as on a variety of complex and natural sounds described in §4.2.

All of our results are performed on 44.1 kHz audio with a window size of 0.05 s or 2206 samples. We use a Hann analysis window with 50% overlap and no synthesis window. Additionally, the windows are padded with zeros to increase the frequency resolution of the FFT to ≈ 5 Hz.

Our implementation is open source and available at https://github.com/sportdeath/audio_transport.

4.1. Interpolating Sinusoids

We used the audio transport effect to interpolate between single sinusoids. Intuitively, this should sound exactly like a sine sweep between the input pitches. We performed listening experiments for interpolations at a variety of speeds and with inputs spanning the entire perceptible range. The spectrogram of one such interpolation is shown in Figure 3. Almost all of the interpolations were indistinguishable from real sine sweeps. In extreme cases where the interpolations were faster than 2000 Hz s^{-1} we perceived some phase distortion, but these situations would be rare in normal use.

The audio does exhibit “stair-stepping” between frequencies due to the frequency resolution of the FFT and the time resolution of the windows as demonstrated in Figure 3. Due to the small

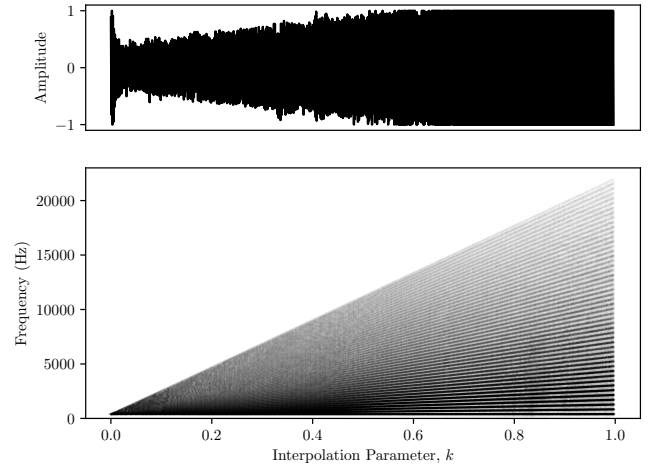


Figure 4: Interpolating between a sine wave ($k = 0$) and a saw wave ($k = 1$) leads to a drop in volume for values of k close to but not equal to zero as shown in the audio signal on top. The spectrogram shows how the sine’s single peak splits into many peaks, which interfere when they are close together.

time-frequency resolution of the steps, however, we were unable to perceive them in the listening experiments.

4.2. Interpolating Natural Sounds

We tested the audio transport effect on a variety of sounds, some of which are available at https://soundcloud.com/audio_transport. The effect intuitively sounds like a portamento, even when applied to unnatural cases like a piano note gliding into a human voice. It is applicable to monophonic and polyphonic sounds. We even had success using it to transition between entire songs. The audio typically sounds artifact-free for many classes of audio signals, with a few exceptions noted below.

The audio transport effect does not guarantee temporal consistency between transport maps. So interpolating between sounds with dynamic spectra, like a pair of wavering orchestral chords, can produce a fluctuating pitch.

Another artifact that can occur is a sudden drop in volume when the interpolation parameter is close to but not equal to either zero or one as shown in Figure 4. This happens when one single frequency is mapped to a large range of frequencies. As the single frequency separates, its components interfere with each other, reducing the volume. This artifact is most prevalent when interpolating between sounds with vastly different spectral complexity as is the case with pairs of bright and dark sounds.

It is also worth reminding the reader that this method is intended for static sounds and will blur transients, as mentioned in §3.3. This artifact can be subtle, but when we directly compared the output of the audio transport effect with an interpolation factor of 0 to the corresponding input we consistently picked out the original audio when it had sharp transients like hi-hats. We suspect that this could be fixed using phase reinitialization techniques [5], but this exploration is left to future work.

5. CONCLUSION

In this paper, we introduced the *audio transport* effect which can create a portamento-like transition between any two audio signals. The effect produces a novel but intuitive sound and it is controlled by a single interpolation parameter. As a result, it is accessible for musicians to incorporate into both live performances and studio recordings.

In our live experiments, we controlled the interpolation parameter using a MIDI pitch fader, but really it can come from any source. For example, an instrument could be constructed where the velocity of a note controls the interpolation parameter. As much as we have described the effect as a portamento, the input pitches do not need to have a different fundamental. The effect also produces interesting interpolations between signals with the same fundamental pitch but different timbre.

Our work on audio transport suggests several other use cases beyond those explored in our experiments. For example, consider a single audio source that is fed as one input of the audio transport effect, and the output of the effect is fed back into the other input. By keeping the interpolation parameter constant, the pitches in the output should lag behind the input pitches similar to synthesizer “glide.” This setup leads to several questions: What would happen if other effects were added to the feedback chain? Is it interesting to use multiple audio transport effects at the same time? The latter may be supported by the notion of *barycenters* in optimal transport [21].

The audio transport effect as described still produces artifacts for certain classes of sounds. Future work to resolve these could investigate ways to sharpen transients, make transport maps temporally consistent, and reduce the effects of energy cancellation. For a wide variety of inputs, however, our effect sounds smooth, musical and inspiring.

Acknowledgments. The authors acknowledge the generous support of Army Research Office grant W911NF-12-R-0011, of Air Force Office of Scientific Research award FA9550-19-1-0319, of National Science Foundation grant IIS-1838071, from an Amazon Research Award, and from the MIT-IBM Watson AI Laboratory. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of these organizations.

6. REFERENCES

- [1] Daniel Leech-Wilkinson, “Portamento and musical meaning,” *Journal of Musicological Research*, vol. 25, no. 3-4, pp. 233–261, 2006.
- [2] Mark Katz, “Portamento and the phonograph effect,” *Journal of Musicological Research*, vol. 25, no. 3-4, pp. 211–232, 2006.
- [3] Ralph Deutsch and Leslie J Deutsch, “Constant speed polyphonic portamento system,” Oct. 19 1982, US Patent 4,354,414.
- [4] James L. Flanagan and R.M. Golden, “Phase vocoder,” *Bell System Technical Journal*, vol. 45, no. 9, pp. 1493–1509, 1966.
- [5] Jean Laroche and Mark Dolson, “Improved phase vocoder time-scale modification of audio,” *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 323–332, 1999.
- [6] Jean Laroche and Mark Dolson, “New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects,” in *Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 1999, pp. 91–94.
- [7] Sascha Disch and Bernd Edler, “An amplitude-and frequency modulation vocoder for audio signal processing,” in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, 2008.
- [8] Peter Neubaecker, “Sound-object oriented analysis and note-object oriented processing of polyphonic sound recordings,” 2009, US8022286B2.
- [9] Rémi Flamary, Cédric Févotte, Nicolas Courty, and Valentin Emiya, “Optimal spectral transportation with application to music transcription,” in *Advances in Neural Information Processing Systems*, 2016, pp. 703–711.
- [10] Filip Elvander, Stefan Ingi Adalbjörnsson, Johan Karlsson, and Andreas Jakobsson, “Using optimal transport for estimating inharmonic pitch signals,” in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 331–335.
- [11] Kelly R Fitz and Sean A Fulop, “A unified theory of time-frequency reassignment,” *arXiv:0903.3080*, 2009.
- [12] Cédric Villani, *Optimal Transport: Old and New*, vol. 338, Springer Science & Business Media, 2008.
- [13] Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas, “Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains,” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 66:1–66:11, 2015.
- [14] Robert J McCann, “A convexity principle for interacting gases,” *Advances in Mathematics*, vol. 128, no. 1, pp. 153–179, 1997.
- [15] Nicolas Bonneel, Michiel Van De Panne, Sylvain Paris, and Wolfgang Heidrich, “Displacement interpolation using Lagrangian mass transport,” in *ACM Transactions on Graphics (TOG)*. ACM, 2011, vol. 30, pp. 158:1–158:12.
- [16] Hugo Lavenant, Sebastian Claiici, Edward Chien, and Justin Solomon, “Dynamical optimal transport on discrete surfaces,” *ACM Trans. Graph.*, vol. 37, no. 6, pp. 250:1–250:16, Dec. 2018.
- [17] Bruno Lévy and Erica L Schwindt, “Notions of optimal transport theory and how to implement them on a computer,” *Computers & Graphics*, vol. 72, pp. 135–148, 2018.
- [18] Gabriel Peyré, Marco Cuturi, et al., “Computational optimal transport,” *Foundations and Trends in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.
- [19] Filippo Santambrogio, *Optimal Transport for Applied Mathematicians*, vol. 55, Springer, 2015.
- [20] Miller Puckette, “Phase-locked vocoder,” in *Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, 1995, pp. 222–225.
- [21] Martial Agueh and Guillaume Carlier, “Barycenters in the Wasserstein space,” *SIAM Journal on Mathematical Analysis*, vol. 43, no. 2, pp. 904–924, 2011.