Audio Transport

Trevor Henderson

May 15, 2017

Abstract

yoyoyo.

1 Introduction

2 Contributions

I developed an algorithm that

techniques to reconstruct an audio signal

I use a new technique to segment the spectrum into smaller pieces. Phase reconstruction

I also implemented the algorithm in real-time to

3 Related Work

Vocoders existed in the 70's. vocoders take a monophonic source and spread it out over a range of pitches. requires harmonic similarity between the sources. Voices have lots of noise so it is easy - Daft punk.

realtime pitch shifting has become very popular in software and hardware. since the early 2000's. Traditional pitch shifters were granular. Updates to algorithms use phase vocoders.

Real time pitch shifting is a key element to autotune, which charachterized the music of the mid 2000's and is still popular today.

Harmonizers sound similar to vocoders. Take a pitch and pitch shift it into new chords.

In 2008 Melodyne pitch shifts individual elements without distoring the sound. They must be processed beforehand and then can be manipulated in realtime

The 2012 product can manipulate polyphonic audio in realtime. http://www.zynaptiq.com/pitch

Most of this work has been proprietary and out of research.

So far as I can

processing but no synthesis. `http://www.maths.lu.se/fileadmin/maths/personal_staff/Andreas_Jakobsson/ElvanderAKJ16_icassp_final.pdf`

http://remi.flamary.com/biblio/flamary2016optimal.pdf

4 Optimal Transport

Discussion of optimal transport.

Distance.

It also allows for you to interpolate between the two shapes along the trajectory.

In 1 dimension it is a solved problem.

For any dimensionality higher than 1, fast computation of the wasserstein2 distance is

Optimal transport is however only defined for, nonnegative masses. Audio has phase. Initial attempts to tackle the problem of phase involved formulating the transport problem differently. But you can't hear the distance between two sinusoids that are played at the same pitch but at different phases But you can hear the "distance" between two pitches.

5 Phase Vocoders

As noted by, audio quality is .

Vertical and horizontal incoherence.

6 Algorithm Overview

6.1 STFT

Like almost all spectral algorithms, this one begins with a short-time Fourier transform (STFT) and ends with it's inverse (ISTFT). At the cost of reduced temporal resolution, the STFT gives us access to the spectral contents of the signal in time.

The STFT fourier transform performs a discrete time Fourier transforms on windows of size M of the original signal $x(n)$. These windows are separated by a hop size of R. The mth frame of the STFT is

$$X_m(\omega) = \sum_{n=-\frac{M-1}{2}}^{\frac{M-1}{2}} x_m(n)w(n)e^{-i\omega n} \tag{1}$$

$x_m(n) = x(n + mR)$ is a buffer of size M of the original signal shifted R samples from the previous buffer. $w(n)$ is the synthesis window of size M.

The ISTFT can then reconstruct the original signal by

In the absence of spectral modifications, we can achieve perfect reconstruction of a signal given that the synthesis and analysis windows obey

$$\sum_m w(n - mR)f(n - mR) = 1, \ \forall n \in \mathbb{Z} \tag{2}$$

I chose to use the root Hann window, which gives perfect reconstruction when $R = \frac{M}{2k}$ for $k \in \mathbb{Z}$.

$$w(n) = \cos(\pi n/M), n \in \left[-\frac{M-1}{2}, \frac{M-1}{2}\right]$$

6.2 Spectrum Segmentation

As we In order to resolve the vertical incoherence — the phasing issues within the frame — my solution like others is to lock regions of the spectrum where phase relations are important. Rather than performing a transport over all bins, bins which are used to represent the same spectral information are lumped together.

TODO Many applications use simple peak finding, which is good in most appliccations but fails in certain settings. Instead I determine where to split the window based on the *reassigned frequency*. The reas

In order to fix the spec Techniques mention peak finding. However peak finding does not give a great sense of where to put boundaries. The reassigned frequency can be computed by:

$$\hat{\omega}(\omega) = \omega + \Im\left\{\frac{X_{\mathcal{D}}(\omega) \cdot X^*(\omega)}{|X(\omega)|^2}\right\} \tag{3}$$

where $X_{\mathcal{D}}$ is the spectrum of the x with the window $w_{\mathcal{D}}(n) = \frac{d}{dt}w(n)$:

$$X_{\mathcal{D}}(\omega) = \text{FFT}\left(w_{\mathcal{D}}(n)x(n)\right) \tag{4}$$

d(root hann window)

$$w_{\mathcal{D}}(n) = -\frac{\pi f_s}{M}\sin\left(\frac{\pi n}{M}\right)$$

Nice plot.

$|S| \leq N$ Segment into how at at most N segments m with arrays X_m with center of masses c_m and masses ρ_m.

$$\sum_{m \in S} \rho(m) = 1 \tag{5}$$

6.3 Transport

After two audio signals have been segmented into S^0 and S^1 we then compute a transformation matrix $T : S^0 \times S^1 \to \mathbb{R}$ that represents the transfer of mass from one segment to the other. We want this matrix to minimize the distance the centers of each segment move. We do not want to transport a negative mass or more mass than is available:

$$0 < T(m^0, m^1) \leq \max(\rho(m^0), \rho(m^1)) \tag{6}$$

Additionally, we want to constrain that all of the mass receives an assignment:

$$\rho(m^0) = \sum_{m^1 \in S^1} T(m^0, m^1) \ \forall m^0 \in S^0 \tag{7}$$

$$\rho(m^1) = \sum_{m^0 \in S^0} T(m^0, m^1) \ \forall m^1 \in S^1 \tag{8}$$

Given constraints 6, 7, 8 we want to find the assignment T that minimizes the 2-wasserstein distance between the segments.

$$\min_{T} \sum_{\substack{m^0 \in S^0 \\ m^1 \in S^1}} T(m^0, m^1)|c(m^0) - c(m^1)|^2 \tag{9}$$

We can satisfy the constrains and approximate the using the following algorithm:

Plot of overlapping pieces

Using this algorithm, we guarantee that the number of assignments is bounded by $|T| \leq 2N - 1$.

6.4 Phase Accumulation

When we move a segment of the spectrum to a new pitch, the phases within that section rotate either slower or faster. This change does not make an audible difference within a window itself, but it can cause interference in the overlap between windows known as *horizontal incoherence.*

Include plot

One solution to this in time-stretching is to estimate the phase of the next frame based on its new instantaneous frequency. If a bin with phase θ_i^τ is oscillating at exactly $\hat{\omega}_i$, the phase of that bin in the next window $\theta_i^{\tau+1}$ will be

$$\theta(i, \tau + 1) = \theta(i, \tau) + \frac{M}{f_s}\hat{\omega}$$

This model is good for , but it breaks down at transients. Reinitialization steps.

Fortunately however, we have at our disposal the phases of not one but two pitches. The center of will oscillate at $(1 - k)\hat{\omega}(m_0) + \hat{\omega}_m^1$. Therefore we can interpolate.

For every bin i keep track of the accumulated phase θ_i^τ which we want to have the following properties:

$$\Theta_i^\tau \approx \sum_{t=0}^{\tau} \frac{R}{f_s}\hat{\omega}_i^t$$
$$\Theta_i^\tau = \theta_i^\tau \pmod{2\pi}$$

Therefore we want the synthesis phase of θ_i at the center of mass of segment m is

$$\theta_{c_m} = (1 - k)\Theta_{c_m^0} + k\Theta_{c_m^1}$$

To preserve the vertical coherence between all bins within a singular we add $\theta'_{c_m} - \theta_{c_m}$ to all pitches

$$\theta'_i = \theta_i + \theta'_{c_m} - \theta_{c_m}$$

This way all the local relationships between phases are kept.

6.5 Resynthesis

Once the phases of each segment have been centered around the synthesis phase, we can synthesize a new spectrum:

```
for $n \in \left[0, \frac{M-1}{2}\right]$ do
    $Y(n) \leftarrow 0$                                    ▷ Clear the output
end for

for $m_0 \in S_0$, $m_1 \in S_1$ do

    $\hat{n} \leftarrow (1-k)c(m_0) + kc(m_1)$              ▷ $\hat{n}$ is the new COM index
    $\hat{\Theta} \leftarrow (1-k)\Theta(c(m_0)) + k\Theta(c(m_1))$   ▷ $\theta'$ is the new COM phase

    for $n \in U(m_0)$ do

        $n' \leftarrow n + \hat{n} - c(m_0)$
        $\Theta' \leftarrow \Theta_0(n) + \hat{\Theta} - \Theta_0(c(m_0))$

        $Y(n') \leftarrow Y(n') + (1-k)\frac{T(m_0,m_1)}{\rho(m_0)}|X_0(n)|e^{i\theta'}$

    end for
end for
```

reentered by the phase accumulation, we can simply linearly interpolate the audio and add it to the spectrum. Since the

7 Implementation

I implemented the above algorithm in real-time as an audio effect in C++. The program listens to two streams of audio which can be sent to it from a any diginal audio workstation, like Ableton Live. It then produces a new stream of audio that is interpolated between the two original streams according to a MIDI value.

Real time. `fftw`

`portaudio` `portmidi`

video

github link

8 Conclusion

If one of the sources is time varying the source material quickly deteriorates. Great for constant sounds.

9 References

Fundemental theory
that phase vocoder one
1d transport paper?
STFT
weighted overlap add
fftw
portaudio
portmidi
cite justin solomon