Voice transformations using extended DSP techniques - Improving the Phase Vocoder

Anders Bargum, Stefania Serafin, Cumhur Erkut

March 6, 2023

The phase vocoder is a widely used and robust pitch-shifting technique for both music and speech. However, the quality of a shifted signal is heavily dependent on both vertical and horizontal phase coherence, which might make the output sound "phasy" and "smeared". This project investigates different techniques to further extend an already complex phase vocoder, using new phase-locking algorithms, true envelope improvements and transient preservation methods. The extensions are evaluated against the their original counterparts and finally compared to a state-of-the art commercial pitch/formant shifter.

1 Project Description

The phase vocoder might traditionally be the most popular technique used for time and pitch-scale modifications of signals and for voice transformation synthesis. The phase vocoder allows one to shift both pitch and timbre of audio and works by analyzing the input signal to extract magnitude and phase for different frequency components. Phase information is then transformed based on user defined pitch or timestretching values, and the modified output signal is reconstructed using the magnitude and newly calculated phase [7]. The vocoding process is most commonly carried out in the frequency domain using the short time fourier transform (STFT) for analysis and the inverse short time fourier transform (ISTFT) for synthesis. However, it can also be achieved using filterbanks representing the different frequency bins of the STFT [1].

When carrying out phase vocoding, it is important to both window and overlap the audio frames when analyzing and synthesising the signal. This is done for several reasons; firstly, it decreases spectral leakage and smoothens out the analysis along the time-axis. Secondly, it allows one to perform re-sampling that changes the pitch of the audio without changing its duration and timing. However, these techniques come with several phase propagation drawbacks, which affect the quality of the reconstructed signal making the stable parts of the audio sound "phasy" and the transients sound "smeared".

To avoid these artifacts, one needs to ensure coherent phases. We here distinguish between "horizontal phase-coherence" being the phases between the frames of the STFT, and "vertical phase-coherence" being the phases between neighboring channels/bins within any given frame [7]. Additionally, aforementioned pitch modification techniques will change a signals formants/spectral structure. Formants are concentrations of acoustic energy around particular frequencies, which naturally will be shifted along with the transformation process [9]. This effect is in particular undesired for speech as it accordingly will affect the articulation of the speech/voice signal and thus not only its pitch. The lack of formant preservation is the reason some people relate the phase vocoder with the

"mickey mouse" effect making the output sound very artificial for high pitches. Formant preservation therefore is another important aspect to consider when implementing the phase vocoder.

There exists several methods to overcome the artifacts introduced by pitch and formant shifts. Some of these are implemented in the phase vocoder used for voice transformations in the 'CHALLENGE' project at Khora VR [11]. To ensure vertical phase coherence, this audio engine among other things include scaled-phase locking, where phase is updated based on the local maxima of the peaks in the analysed frequency channels [7]. It additionally preserves formant structure by pre-warping the spectral envelope of each signal frame using the cepstral-based true envelope. While the output of the engine sounds good for moderate pitch-shifting scenarios, it decreases in quality for extreme settings resulting in a smeared, metallic and harsh output. With a starting point in these deficits, this report examines techniques to extend the already implemented methods in order to further improve the phase vocoder used in the 'CHALLENGE' project.

2 Reducing Phasiness

To ensure phase coherence, the phase vocoder in the 'CHALLENGE' project firstly implements 'scaled phase-locking', which synchronizes the phases between the vocoder channels/bins and the given frames. Both for constant-amplitude, constant-frequency sinusoids and for sinusoids with slowly varying frequencies it is assumed that the channels located around the sinusoid will have either identical or nearly equal analysis phases [7]. Additionally, it is assumed that a peak in the input spectrum often switches from channel $k0$ at frame $u-1$ to channel $k1$ at frame u. The 'scaled peak phase-locking' algorithm uses these assumptions by relating the phase of the peaks in the STFT magnitude representation to its surrounding bins. A so called 'region of influence' is thereby created, each determining phase-propagation for the new peak-bin in the next region. This gives the phase update equation [6]:

$$\angle Y(t_s^u, \Omega_{k1}) = \angle Y(t_s^{u-1}, \Omega_{k0}) + R_s \omega_{k1}(t_a^u), \tag{1}$$

where $\angle Y(t_s^u, \Omega_{k1})$ is the synthesis phase in the new peak bin, $\angle Y(t_s^{u-1}, \Omega_{k0})$ is the synthesis phase of the former peak in the former frame and $R_s \omega_{k1}(t_a^u)$ is the phase increment of the analysis time frame. Each bin within the peak region is then synchronized to equation (1), given by:

$$\angle Y(t_s^u, \Omega_k) = \angle Y(t_s^u, \Omega_{kl}) + \angle X(t_a^u, \Omega_k) - \angle X(t_a^u, \Omega_{kl}) \tag{2}$$

Here k constitutes all bins in the given peak's region of influence. While the 'scaled phase-locking' algorithm significantly improves the 'phasiness' of the traditional vocoder, the local maxima search it is build upon, often results in small artifacts heard as shallow bass and musical overtones [5].

2.1 Multi-Resolution Peak Picking

Rather than using the constant resolution peak-picking algorithm outlined above, this project investigates a *multi-resolution peak picking* stage, which is a technique grounded in human perception and the non-uniform characteristics of the human auditory system [5]. In the case of multi-resolution peak picking, one considers the frequency distribution of speech signals, whose spectral peaks often are concentrated in the lower-regions roughly following a logarithmic scale [5]. The 'regions of influence' are here defined to be non-uniform with a higher peak-picking resolution between 80 Hz and 2000 Hz, and a lower resolution above 2000 Hz. More specifically, for a frame size of 2048 sampled at 44100 Hz, the algorithm considers the first 8 bins (up to 172 Hz) as 8 individual components to which no surrounding bins are synchronized. For the next 8 bins (from 172 to 344 Hz) the algorithm looks at one neighbouring bin to define the peaks and synchronize the neighbours on each side of the bin.

The next 8 bins look at two neighbouring bins and so on. The comparison of pitch shifting a clap sample down -2.40 semitones using both the constant and multi-resolution peak picking algorithms, each without formant preservation, can be seen in figure 1. It is here clear that the "pre-echo" of the multi-resolution peak picking algorithm is lower than the constant-resolution algorithm. Additionally, the constant-resolution output is more "stretched" time-wise, which gives the undesired "phase smearing" characteristics. The effect of the multi-resolution peak picking algorithm is in this case audible as prominent transients, as well as more controlled bass-frequencies matching the ones of the original signal. However, applying the multi-resolution algorithm to speech is experienced to be less perceptual due to the constant masking and overlapping of different vowels and consonants happening in speech. Nonetheless, the algorithm has in this project heuristically been found to improve the "metallic" artifacts happening at extreme pitch shifting values.

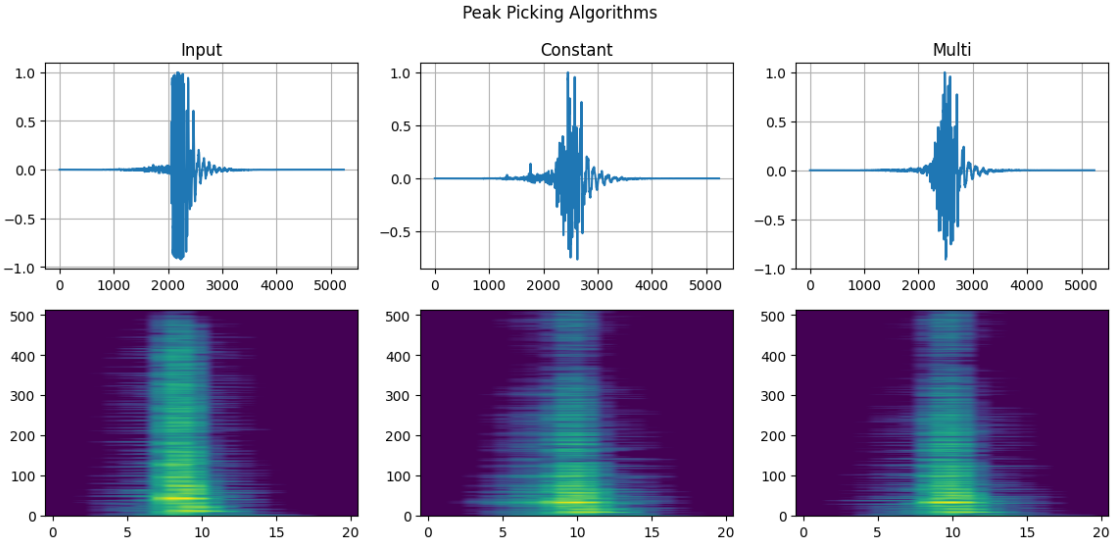

Figure 1: A clap sample affected by the constant and multi resolution peak picking algorithms

In [5], the multi-resolution technique is additionally said to reduce musical overtones. These however, could not be provoked by the constant-resolution peak picking algorithm in the first place.

2.2 Sinusoidal Trajectory Heuristics

Due to the nature of the phase continuation assumptions mentioned earlier, another disadvantage of the 'scaled phase-locking' algorithm is that it potentially might correlate two unrelated sinusoids in neighbouring bins across two sample frames. Here problems arise when the distance between the two bins is very large, in which case the peaks almost certainly do not belong to the same partial. According to [5], this often happens at note onsets or in frequency areas sparsely populated with peaks resulting in blurred note onsets and high frequency 'warbling' artifacts. To overcome this, one can use sinusoidal trajectory heuristics to check if the predecessor of a given peak is within a specific distance. Here, only peaks within a very short distance are linked by equation (1) and (2). The allowed distances are, similarly to the multi-resolution peak picking technique, logarithmically spaced based on human hearing. We implement this as a simple function checking whether both the current and the former peaks, stored in two different arrays, are within the same frequency region.

Listing 1: function for comparing peak distances

```
std::vector<float> subBandRanges = {0.0f, 172.0f, 345.0f, 689.0f, 1378.0f, 2756.0f,
5513.0f, 22050.0f};

float index2Freq(int i, float fs, int fftSize)
{
    return (float) i * (fs / (fftSize));
}

bool compareCurrentAndPreviousBins(int currentPeak, int previousPeak)
{
    for (int i = 0; i < 8 - 1; i++)
    {
        if (index2Freq(currentPeak, fs, fftSize) >= subBandRanges[i] &&
        index2Freq(currentPeak, fs, fftSize) <= subBandRanges[i+1]   &&
        index2Freq(previousPeak, fs, fftSize) >= subBandRanges[i]    &&
        index2Freq(previousPeak, fs, fftSize) <= subBandRanges[i+1])
        {
            return true;
        }
    }

    return false;
}
```

A heuristic evaluation of the sinusoidal trajectory heuristics implementation have shown no improvement in the overall output quality in this project, neither for speech nor musical/transient inputs, however it has been included in the final improvements for precaution and computational efficiency.

2.3 Resetting the Phases

The last suggestion of [5] is to reset the phases in silent passages of the signal. The reason for doing this comes from the fact that phases might accumulate incorrectly over time sequentially affecting the phase unwrapping algorithm. This can be fixed by resetting the signal phases during silence, allowing the phase propagation algorithm to get a fresh start once audio re-appears. In this project the phases are reset once the energy of the short time spectrum drops below a certain threshold, more specifically -20 dB. While the cumulative phase errors mostly happens after long periods of audio, this extension is rarely audible, however, it secures that the phases in the vocoder continues to propagate correctly.

3 Transient Preservation

To further improve the transient smearing effects introduced by the phase vocoder, 'harmonic-percussive separation' has been investigated as means of preserving transient information. The task of decomposing an audio signal into its harmonic and its percussive components has received large interest in related literature as it often has proven useful to process transients and stationary regions of a signal individually. For this project, the motive has been to process the transient and harmonic parts of the input signal in parallel affected by a two phase vocoders with specifically tailored configurations. In particular, the transient segments will be processed by a phase vocoder initialized with smaller STFT frame sizes and thus a higher time resolution in the converted frequency domain. This allows one to preserve the transient nature of the percussive segments, with a tradeoff in the

pitch shifting quality. However, this is acceptable as the transients will be summed together with the audibly pitch shifted harmonics happening in the vocoder path with larger frame sizes.

The idea and steps of the harmonic- percussive separation method are mainly inspired by [3] and [2], which uses classical image processing techniques directly on the spectral information of an input. This is done by filtering the spectrogram of a signal in both the horizontal (to enhance harmonics) and vertical (to extract transients) direction. The process of harmonic-percussive separation can be divided into several steps:

(1) Firstly, we convert the input signal into frequency domain through an STFT. For this specific case a window size of 1024 samples and a hop size of 128 samples has proven sufficient.

(2) Thereafter we filter the spectrogram to obtain its horizontal and vertical components. This can be done using a median filter with a one-dimensional kernel, either stacked in rows (horizontal) or as a column (vertical). More specifically we obtain the horizontally filtered spectrogram Y^h and the vertically filtered spectrogram Y^v by [3]:

$$Y^h(n,k) = \mu_{1/2}((Y(n-(L^h-1)/2,k),...,Y(n+(L^h-1)/2,k))) \tag{3}$$

$$Y^v(n,k) = \mu_{1/2}((Y(n,k-(L^v-1)/2),...,Y(n,k+(L^v-1)/2))) \tag{4}$$

Where $\mu_{1/2}$ refers to the median, and L^h and L^v are the horizontal and vertical filter kernels, often defined by an odd size/length. In this project the filter size is 15 on both axes.

(3) We then convert the filtered spectrograms to a decibel scale and construct binary masks. This results in a binary percussive mask M^v that is 1 when its components are higher than its horizontal counterpart and 0 otherwise:

$$M^h(n,k) = \begin{cases} 1, & \text{if } Y^h(n,k) > Y^v(n,k) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$M^v(n,k) = \begin{cases} 1, & \text{if } Y^h(n,k) < Y^v(n,k) \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

(4) Thereafter morphological erosion are applied to the masks. Morphological erosion is a classical image processing technique that removes floating pixels and thin lines so that only substantive objects remain. By eroding the masks we simply avoid noisy behaviour.

(5) Then another binary separation process is applied to the eroded masks, in this case defined by a user-defined threshold. We here classify an entire time frame of the two masks as either percussive or harmonic if the summed energy in that frame is above a certain threshold.

(6) Lastly, we perform an element-wise multiplication of the eroded, thresholded binary masks with the original spectrogram from step 1 to obtain percussive and harmonic spectrograms. Thereafter an ISTFT is applied and the elements are added together in the time domain either before or after additional processing such as phase vocoding. All steps are exemplified on a percussive loop in figure 2 on the following page.

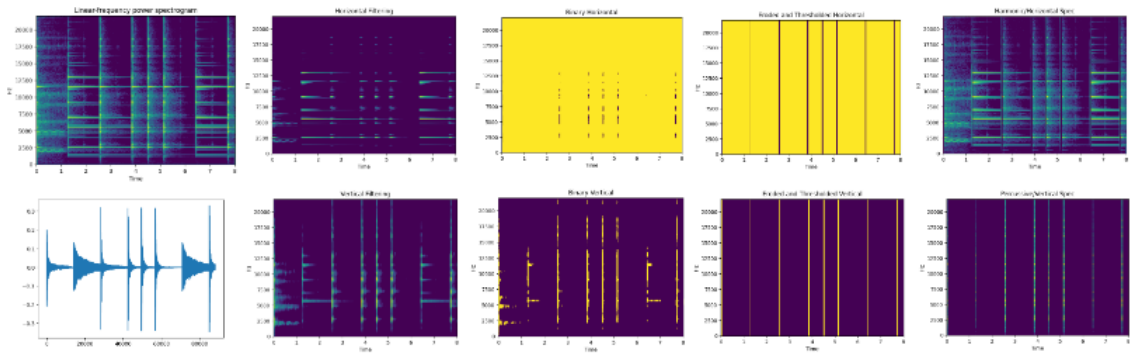

Figure 2: The different steps of the harmonic-percussive separation process

The process of separating percussive and harmonics parts of the signal additionally introduces the possibility of scaling the different components to better align in volume and thereby match the structure of the input. To showcase harmonic-percussive separation we apply it on an excerpt of the percussion loop used in figure 2. In this case we compare the input to its -4.5 semitone shifted equivalents using the traditional phase vocoding technique and the harmonic-percussive separation technique fed into two parallel vocoding processes. Here an fft size of 1024 is used for the harmonic phase vocoding and a size of 256 is used for the percussive phase vocoding. Both processes utilises a hop-size of 1/8 of the given framesize. As seen below, the signal with harmonic-percussive separation clearly preserves the transients of the input, additionally avoiding the smearing, slow attack and pre-echo artifacts provoked by the traditional phase vocoder.

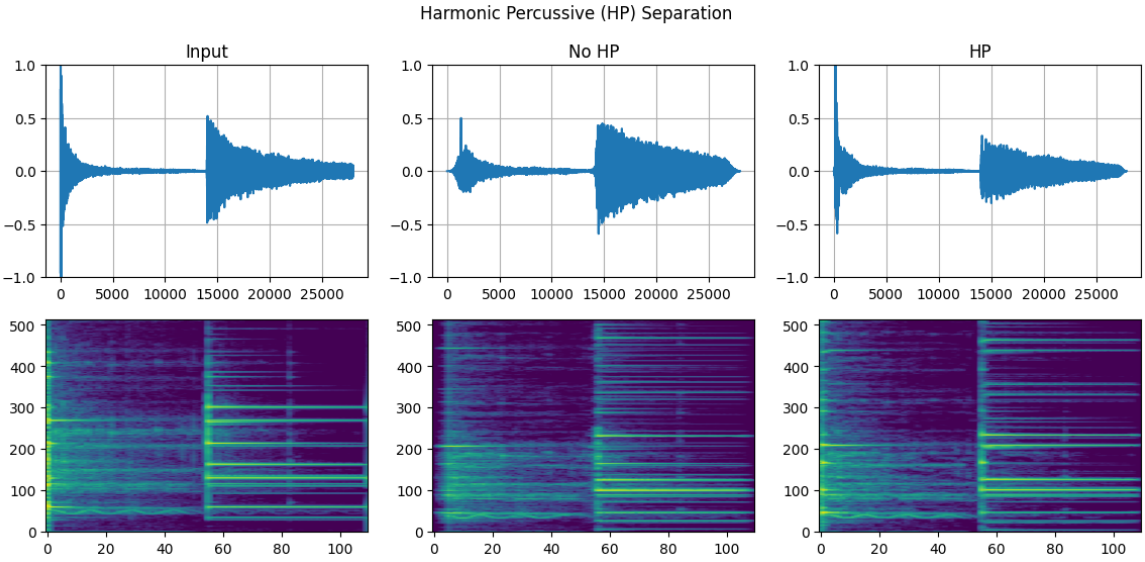

Figure 3: Harmonic percussive separation on a percussive excerpt

4 True Envelope

The final extension of the phase vocoder is the act of preserving the spectral envelope as a mean of keeping the formant structure of a pitched signal. As mentioned in the project description earlier, the prominent peaks of the spectrum, constituting formants, will automatically be shifted proportionally to the pitch, resulting not only in tonal but also timbral changes. A straightforward method to prevent timbre modification consists in pre-warping the spectral envelope of each signal frame in the spectral domain before the transposition is actually performed. In this way a formant move is done in the opposite direction of the pitch shifted result, providing the transposition calculation process with magnitude values that equalizes the structural spectral changes happening in the transformation process. The spectral envelope is a curve in the frequency-amplitude plane fitting the outline/contour of the magnitude spectrum. In order to avoid oscillations in the envelope it is important that the envelope is; *regular* i.e, the envelope is giving a general (non detailed) idea of the distribution of the signal's energy over frequency, and *steady* i.e has no corners/derivative jumps [10]. The audio engine of the "CHALLENGE" project already implements one of the most prominent and complex ways of estimating the spectral envelope, namely the 'true enevelope'. The true envelope is a spectral envelope estimation technique, which iteratively updates the smoothed input spectrum with the maximum of the original spectrum and the current cepstral representation [4]. The backbone of the true envelope therefore is the filtered cepstrum. The cepstrum is a fourier representation of the log amplitude spectrum of a signal and is highly effective for examining periodic structures and peaks in the frequency domain [8]. We calculate the cepstrum $C(l)$ by:

$$C(l) = \sum_{k=0}^{K} log(|X(k)|)e^{\frac{i2\pi kl}{K}} \tag{7}$$

Where k is the individual bins for the K-point DFT and $e^{\frac{i2\pi kl}{K}}$ represents the ISTFT. By filtering the cepstrum, one retrieves an envelope following the mean of the spectrum, which then iteratively can be used to update the target spectrum and obtain a contour matching the peaks and valleys of the spectrum [9]. Even though the iterative process of estimating the true envelope is rather computationally demanding, the true envelope has proven to outperform other state of the art spectral estimators in terms of precision. The current implementation of the 'CHALLENGE' project therefore has been difficult to improve [9]. However, the algorithm for calculating the true envelope leaves several values behind to experiment with, in order to optimize the estimation process of the spectral envelope:

- Pc: Pc is the cepstral order and defines the number of harmonics used for estimating the envelope of the non negative frequency axis. It can be calculated by $\frac{Fs}{2\delta F}$.
- δF: The largest distance between two of all neighboring spectral peaks. This value basically is the fundamental frequency F0 for harmonic sounds.
- S: The smoothing factor deciding how much the envelope should smooth between the peaks and valleys of the spectrum. The smaller the value S, the more precise the contour of the peaks are followed. A smoothing factor of 1 results in the exact fft structure of the input, which is not desired.

To improve the control of the true envelope and keep the resulting spectral envelope stable over time, we implement a process to dynamically estimate δF for each frame. This is done by iteratively checking the distance between all detected peaks. The maximum value between the peaks will define δF and be parsed to the true envelope function.

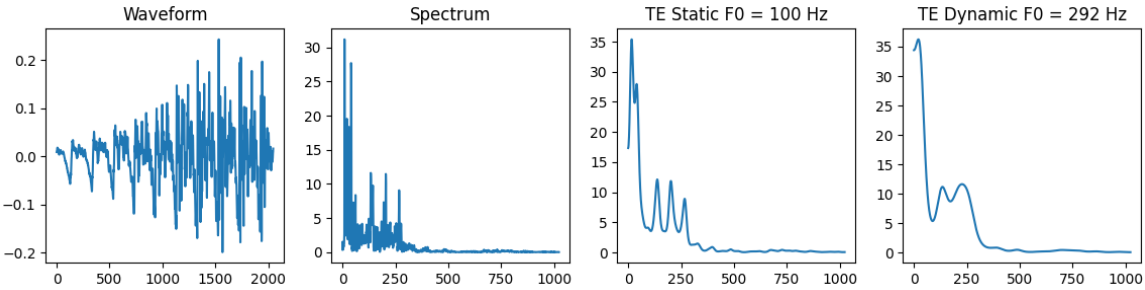

Figure 4: Spectral envelope estimation

In figure 4 above the true envelope of a signal frame (size = 2048 samples) is depicted. Both envelopes has a static smoothing factor of 0.4, however, the frequency δF, and thus the cepstral smoothing order P_c, is either statically defined to be 100 Hz as done in the current "CHALLENGE" audio engine, or calculated based on maximum peak distance, which for this frame is 292 Hz. As seen, the envelope based on the peak distance calculation is both more regular and more steady than its static counterpart. This is simply due to the fact that the $F0$ of the signal excerpt is not equal to the static value of 100 Hz. Dynamically changing this value for each frame thus allows the true envelope, and thereby the pitch pre-warping process, to stay robust towards drastic changes in the frequency spectrum.

5 Evaluation

To evaluate the proposed extensions, we implement the aforementioned true envelope and phase coherence techniques into the current 'CHALLENGE' audio engine. While 'transient preservation' showed promising results, it has been decided not to include it into the evaluation process as the current 'CHALLENGE' implementation does not afford the extra amount of CPU required by such a process and therefore cannot be integrated into the engine for real-time usage. We evaluate the original, constant peak-picking vocoder and the new implementation either individually or against the state-of-the-art pitch/formant shifter "elastique pitch V2"[1], which we in this case use as a baseline. The different versions are evaluated using professionally verified objective metrics specifically tailored speech and voice data, such as:

- **PESQ**: Perceptual evaluation of speech quality (PESQ) is a recognized industry standard for comparing audio quality between a prediction and a target. It takes into consideration characteristics such as: audio sharpness, background noise, clipping and audio interference. PESQ returns a score between -0.5 and 4.5 with the higher scores indicating a better quality.
- **SI-SDR**: Scale-invariant signal-to-distortion ratio (SI-SDR) is an overall measure of how good a signal "sound" compared to a target. The value is expressed in dB, where higher is considered better.
- **MOS-Net:** MOSnet is a deep learning based objective assessment metric for voice conversion. It is perceptually based as it is trained to predict human ratings of converted speech. MOSnet

[1] https://www.thomann.de/dk/zplane_elastique_pitch.htm

is an 'absolute' measure meaning that a target is not required. Higher results are considered better quality.

- **SRMR:** Speech-to-reverberation modulation energy ratio (SRMR) is an absolute, non-intrusive metric for speech quality and intelligibility based on a modulated spectral representation of the speech signal. The higher the SRMR value the better intelligibility.

We compare the original, proposed and state-of-the-art implementations in different scenarios/-configurations with samples around 3.5 seconds long. Since the main problems of the audio in the 'CHALLENGE' project happens when signals are shifted down e.g. for female-to-male conversions, we only use configurations reflecting this. We inspect the techniques on two input-signals: male and female speech, in three different configurations each. The best score for each objective metric is highlighted in bold.

Configuration (Pitch, Formant)	**Model**	**PESQ**	**SI-SDR**	**MOSNet**	**SRMR**
Male (-7, -2)	Original	1.222	-13.145	2.892	**4.91**
	Proposed	**1.246**	**-11.853**	**3.522**	4.054
Male (-5, -5)	Original	**2.009**	-14.529	2.559	5.698
	Proposed	1.489	**-11.319**	**2.717**	**5.892**
Male (-3.5, -1)	Original	**1.783**	-33.734	**3.008**	**4.238**
	Proposed	1.595	**-11.377**	2.857	3.621
Female (-7, -2)	Original	1.207	-48.947	**4.516**	9.744
	Proposed	**1.278**	**-39.35**	3.11	**10.083**
Female (-5, -5)	Original	**2.111**	-26.345	**3.538**	**14.158**
	Proposed	1.734	**-19.319**	3.173	12.863
Female (-3.5, -1)	Original	1.346	**-26.856**	3.081	9.97
	Proposed	**1.414**	-39.741	**3.095**	**11.168**

Table 1: *Evaluation of the original and the proposed techniques*

As seen on Table 1, the phase vocoder including the proposed extensions performs better than the original engine on 3/4 metrics for the extreme configurations, whereas it evidently is less stable for pitch/formants shifts in more controlled ranges. These differences were furthermore heuristically audible, where extreme shifts showed to sound less-stretched in the proposed case, but also less stable for smaller pitch/formant shifts. Based on these observations it has been decided to activate the currently proposed changes when extreme values are used. Thus when the pitch and formants shifts reach a certain threshold, the new extensions activate in order to improve the more metallic sounds provoked by such configurations, whereas smaller pitch/formant shifts are manipulated by the current audio engine.

6 Conclusion

This project has successfully investigated, implemented and evaluated possible extensions for the traditional phase vocoder. As suggested by related literature it is shown that phase locking based on multi-resolution peak picking and transient preservation methods, improve phase coherence and thus mitigate "smearing" and pre-echo artifacts. The already implemented true envelope is additionally examined and new features are suggested for the spectral envelope estimations process to be more robust to large harmonic changes in the input frames. Objective metrics have been used to evaluate

the final extensions against the original audio engine and a state of the art pitch shifter, showing that the researched techniques are useful in extreme settings only, but should be considered with care for normal pitch and formant shifting values.

References

[1] Mark Dolson. The phase vocoder: A tutorial. *Computer Music Journal*, 10:14, 1986.

[2] Jonathan Driedger, Meinard Müller, and Sebastian Ewert. Improving time-scale modification of music signals using harmonic-percussive separation. *Signal Processing Letters, IEEE*, 21: 105–109, 01 2014. doi: 10.1109/LSP.2013.2294023.

[3] Derry FitzGerald. Harmonic/percussive separation using median filtering. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 246–253, Graz, Austria, 2010.

[4] Rong Gong, Philippe Cuvillier, Nicolas Obin, and Arshia Cont. Real-time audio-to-score alignment of singing voice based on melody and lyric information. In *Interspeech*, 2015.

[5] Thorsten Karrer, Eric Lee, and Jan Borchers. Phavorit: A phase vocoder for real-time interactive time-stretching. In *Proc. Intl. Computer Music Conf.*, 11 2006. URL `https://hdl.handle.net/2027/spo.bbp2372.2006.142`.

[6] J. Laroche and M. Dolson. Improved phase vocoder time-scale modification of audio. *IEEE Transactions on Speech and Audio Processing*, 7(3):323–332, 1999. doi: 10.1109/89.759041.

[7] J. Laroche and Mark Dolson. Phase-vocoder: about this phasiness business. In *Proc. WASPAA*, page 4 pp., 11 1997. ISBN 0-7803-3908-8. doi: 10.1109/ASPAA.1997.625603.

[8] Axel Roebel and Xavier Rodet. Real time signal transposition with envelope preservation in the phase vocoder. In *Proc. Intl. Computer Music Conf.*, 2005. URL `http://hdl.handle.net/2027/spo.bbp2372.2005.200`.

[9] Axel Roebel and Xavier Rodet. Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation. In *Proc. Intl. Conf. Digital Audio Effects (DAFx)*, Madrid, Spain, 2005.

[10] Diemo Schwarz. *Spectral Envelopes in Sound Analysis and Synthesis*. PhD thesis, IRCAM, 06 1998.

[11] Luis Viera, Peter Fisher, Simon Lajboschitz, Stefania Serafin, and Merete Nordentoft. The challenge project: Fighting auditory hallucinations by using virtual reality. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 103–108, 2021. doi: 10.1109/VRW52623.2021.00027.