Jewels: An Audio Effect for the Extension and Movement of Spectral Peaks

Chet Gnegy
Center for Computer Research in Music and Acoustics (CCRMA)
Stanford University
chetg@stanford.edu

ABSTRACT

In this paper, we introduce a spectral audio effect that searches for the most prominent frequency components in each frame of an input signal. These components are extended in time with exponential decay creating strong resonances in the signal. Peaks can be shifted in frequency over time following a constant, random, or oscillatory trajectory. The movement of spectral peaks creates many interesting sounds for input instrumental signals. Additionally, by using a white noise input, Jewels becomes a versatile noise synthesizer.

1. INTRODUCTION

Jewels is a real-time spectral audio effect that extends the peaks in the frequency spectrum forward in time to create a reverberant effect. Unlike a traditional convolution reverb, Jewels is not attempting to simulate an environment or create the auditory illusion that the sound is occurring in a particular space. Instead, the effect uses the properties of the signal itself to create resonances or other unique effects. The system diagram for Jewels is shown in Figure 1.The majority of the processing is done in the spectral domain and special considerations are taken to ensure that the audio output is independent of the system audio buffer size. The time extension of the spectral peaks may remain stationary with respect to frequency, creating a strong resonant effect. A spectrogram of this effect can be seen in Figure 2. The peaks can also be set to move with some trajectory, either constant, randomly, or sinusoidally. By simply manipulating the frequency shift of these bins with respect to time, a variety of different audio effects are produced ranging from Shepard tones to noise, to a shimmering effect. If instead of using a musical audio signal as an input, we use white noise, we obtain a versatile toolkit for noise synthesis.

2. SMART BUFFERS

Rather than simply taking the FFT of size N_s, the buffer size of the audio system, Jewels is designed to compute FFT frames of a fixed size, N_e. This ensures that the result of the effect will not change as system parameters change. To do this, we now introduce objects called smart buffers. The smart buffer allows data buffers of unequal length to be written and read. For example, consider a smart buffer, B_1, that uses a write buffer length of N_s and a read buffer length of N_e, where $N_s = 256$ and $N_e = 512$. In this case, we must write two buffers to B_1 before we can read one. The spectral effect can then be processed using a buffer size of N_e. Once processed, the data is fed into a second smart buffer, B_2, with write buffer length of N_e, and a read buffer length of N_s. The samples are read out of B_2 and returned to the sound card at the original buffer size, N_s. For values of $N_e < N_s$, we process several buffers for every incoming audio buffer. For values of $N_e > N_s$, multiple incoming audio buffers are required to compute a spectral frame. When $N_e = N_s$, we can retrieve a processed buffer for every incoming audio buffer. Note that the entire system assumes a latency associated with the larger of the two buffer sizes.

Additionally, the smart buffers allow for reading and writing with overlap. In terms of the Fourier transform, this corresponds to a hop size that is less than the size of the buffer. In all cases, our overlap was equal to $N_e/2$ samples.

The smart buffer uses a circular buffering scheme, the internal buffer size being equal to $2\max(N_s, N_e)$. Pointers corresponding to the oldest and newest elements are held in the smart buffer. At any point in time, these number of samples between these pointers determine whether there are enough frames to return a buffer or not.

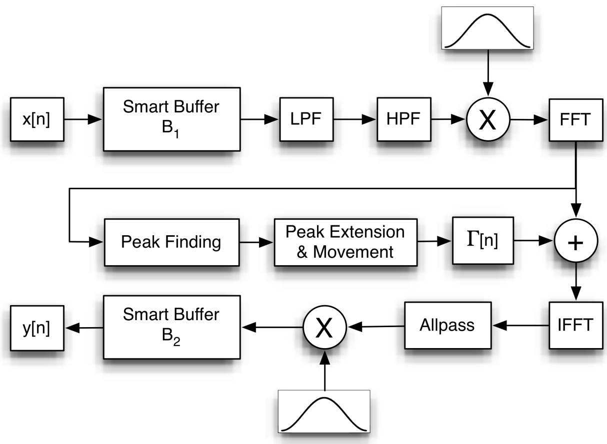

Figure 1: The flow chart for Jewels

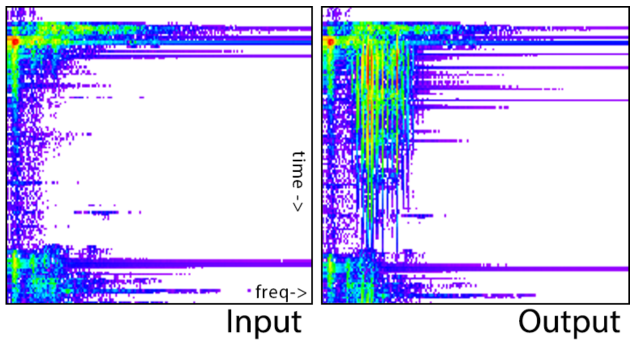

Figure 2: The peaks found in the input spectrum are time extended causing a reverberant sound. The positive frequency magnitude spectrum is shown with the time increasing from top to bottom.

3. EFFECT OUTLINE

3.1 Filtering and Windowing

Prior to passing our time domain signal into the smart buffer, we filter using second order low pass and high pass filter in sequence with cutoff frequencies f_{low} and f_{high}, respectively. This is done primarily to change the frequency range over which spectral peaks are likely to be found. For example, a low pass filter with cutoff $f_{low} = 5kHz$ will nearly eliminate the possibility of having a global maxima in the spectrum at 1kHz. The low pass filter is also useful for removing frequencies below 20Hz, especially because it is undesirable to have a DC component in the audio signal.

As soon as a length N_e buffer is retrieved from B_1, it is windowed by $w_{N_e}[n]$. We use a Hann window for every buffer, and we apply it such that the signal is point wise multiplied by the square root of the window coefficients once at the beginning of processing and again at the end. The general formula for a length M Hann window is seen in Equation 1. A length N_e window is used with an overlap of $N_e/2$, resulting in perfect reconstruction of the original signal when the signal is returned to the time domain. We obtain the FFT of the windowed signal, F, and look for a spectral peak using Equation 2. The maxima in the current spectrum is found in bin k_{max} and has a complex amplitude $F[k_{max}]$. If the spectral peak is not above some user specified energy threshold, α, it is ignored.

$$w_M[n] = 0.5\left(1 - cos\left(\frac{2\pi n}{M-1}\right)\right) \tag{1}$$

$$k_{max} = \underset{k\in\{0,N_e/2\}}{\arg\max}\ \{F[k]\} \tag{2}$$

3.2 Peak Extension

The spectral peaks with an amplitude greater in magnitude than α are stretched forward in time. This is done using a circular buffer of spectral frames, Γ, where $\Gamma[n]$ is the spectral data for the set of samples starting $nN_e/2$ samples in the future. The factor of 1/2 is present due to the overlap in spectral frames. We extend the peak forward in time by adding the complex amplitude, $F[k_{max}]$, seen at the peak into the next several frames, with a complex scaling factor to remove phase distortion and cause the peak to decay in time. This scaling factor, $\gamma_{n,k}$, is determined by the frequency bin and the index of the spectral frame, as seen in Equation 3. The relationship between $\Gamma_k[n]$ and $F(k)$ is shown in Equation 4. The factor $ne^{\frac{-8n}{\tau}}$ causes the peak to be extended forward into time with exponential decay by roughly τ seconds. The factor of -8 specifies that the signal will decay by 8 time constants before it is considered to be equal to zero, at which point, it is truncated to reduce computation. This form of decay was used because the exponential decay is seen in many natural systems, bells and percussive instruments, for example. Additionally, this function is smooth and does not introduce amplitude discontinuities into the signal. Phase matching is ensured across frame boundaries by the factor $e^{\frac{j\pi}{2}kn}$. In practice, this term reduces to $(-1)^{kn}$.

$$\gamma_{n,k} = ne^{\frac{-8n}{\tau}}e^{j\pi kn} = ne^{\frac{-8n}{\tau}}(-1)^{kn} \tag{3}$$

$$\Gamma_{k_{max}}[n] = \gamma_{n,k_{max}}F[k_{max}] \tag{4}$$

Because the spectral characteristics of a signal cannot be accurately described by the information in a single frequency bin, Equation 4 is applied to a bandwidth of $2\Delta k$ bins, centered around k_{max} (Δk bins on each side of k_{max}). We modify Equation 4 to include this bandwidth in Equation 5. The bins are windowed to reduce artifacts that would be caused by using a rectangular window (which would correspond to convolution with an ideal filter and would cause time aliasing). The effects of changing this parameter can be seen in the spectrograms in Figure 3.

$$\Gamma_k[n] = \gamma_{n,k}w_{2\Delta k}[k - k_{max} - \Delta k]F[k] \tag{5}$$

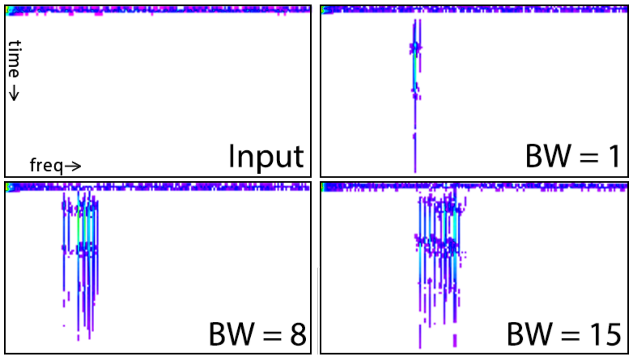

Figure 3: By changing the bandwidth of the applied effect, the texture of the signal is changed. Above we see the effect of setting the bandwidth to 1, 8 and 15 bins. Note that over several frames, the time extension of the peak may appear wider than this number. This is because each frame may find a peak in a different frequency bin, causing multiple superimposed extensions.

3.3 Peak Movement

The peaks can be given a trajectory as they are stretched forwards in time. This is done by manipulating the index of the bin, k, when calculating $\Gamma_k[n]$. We do this using an offset term, $k_{off}[n]$, seen in Equation 6. Jewels allows for values of k_{off} that cause the signal to have a constant, random, or sinusoidal drift. A constant drift in k_{off} creates the effect that the sound is moving upwards or downwards in frequency. The random drift feature increments k_{off} by an accumulating random positive or negative amount for every additional spectral frame, in other words, the peak takes a random walk. The random drift, or wander, is similar to injecting random noise into the signal. The peaks can also move sinusoidally. For higher frequency peaks, this creates a shimmering effect. To account for non-integer values of k_{off}, we use linear interpolation when calculating $\Gamma_{k+k_{off}[n]}[n]$. Frequencies that are less than zero are reflected back into the positive side of the spectrum. After processing, the positive frequencies are copied into the negative frequency bins so that under all cases the spectrum remains symmetrical .

$$\Gamma_{k+k_{off}[n]}[n] = \gamma_{n,k}w_{2\Delta k}[k - k_{max} - \Delta k]F[k] \tag{6}$$

3.4 Post Filtering and Long Term Behavior

The discussion thus far has been for peak finding and trajectory mapping of a single spectral frame, but it is important to note that this computation is run for every incoming window and the computed outputs are superimposed. This has many consequences for the effect. For example, the trajectory drift feature creates sounds that resemble a Shepard tone when a new peak is triggered in several sequential windows. The oscillatory peak trajectories also

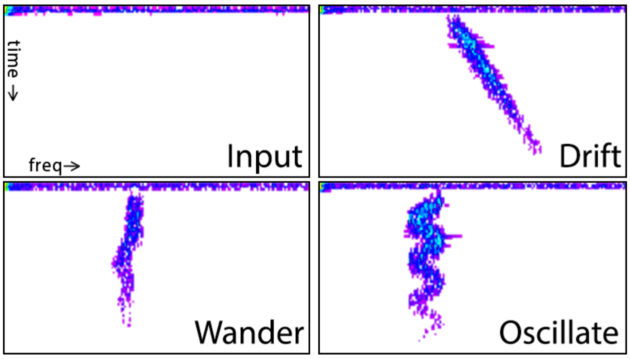

Figure 4: The spectrograms for peak drifting, wandering, and oscillation using length 2000 samples of Hann windowed exponential noise. The precise location of the found peak varies for each burst of noise, but the behavior of each variety of peak trajectory does not depend on the bin number.

sound somewhat noisier than one may expect because the sinusoidal paths are not in phase with each other.

Finally, the signal is passed through a cascade of all-pass filters. These filters make the sound less metallic and add a bit of reverberation, in the same manner that is used for the popular Freeverb reverberator. [1]

Seen in the Appendix and in Figures 5 and 6 are the settings and spectrograms for the `Shimmer` and `Shush` presets. The `Shimmer` effect features some subtle high frequency ringing with slight positive drift and oscillation. By increasing the decay parameter, the effect becomes much more apparent. `Shush`, as the name suggests, sounds similar to a person saying "Shush" whenever a sound is made. The main contributor to this sound is the wander parameter which is set at a maximum.

4. NOISE SYNTHESIS

In addition to using Jewels as an audio effect for an instrument, it can be used as a noise synthesis tool. Using a white noise source as an input, we expect that peaks will occur over the entire audio spectrum due to the flat, non-deterministic spectrum of white noise. This is only the case when the pre filtering stage is at the extremes, $f_{high} = 20Hz$ and $f_{low} = 20kHz$. For other cases, the peaks will be within the passband defined by these two filters. Many interesting effects can be obtained by changing the parameters. The Appendix gives the parameters for two synthesized noise presets, `Shepard` and `Artifacts`. `Shepard` produces tones that seem to endlessly rise or fall due to the constant retriggering of rising peaks. `Artifacts` produces a sound similar to the artifacts that occur when a signal is time stretched. The spectrograms for these presets given a white noise input are seen in Figures 7 and 8.

5. CONCLUSIONS

In this work, we have presented a spectral effect for locating prominent frequency components and giving them a trajectory in time. The usage of smart buffer objects allows for constancy of the effect behavior even when the size of the system audio buffer has changed. The motion of spectral peaks has proven to produce a very wide range of auditory effects for musical audio signals as well as noise signals. Whether the user is looking for a more resonant quality for their signal or an unusual new variety of noise, Jewels is a good solution.

[1] https://ccrma.stanford.edu/~jos/pasp/Freeverb_Allpass_Approximation.html

APPENDIX

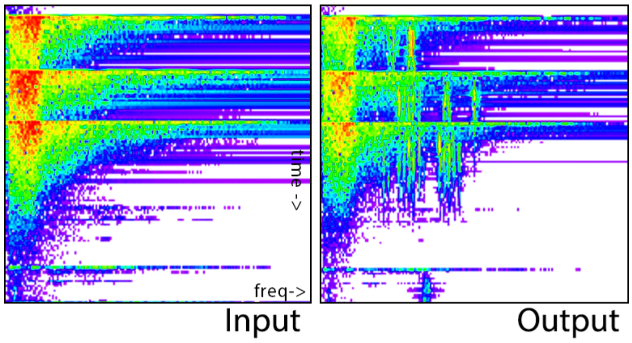

Figure 5: The spectrogram for the `Shimmer` preset shows upper frequency peaks that are extended with a small amount of drift and oscillation. The input signal is three loud claps.

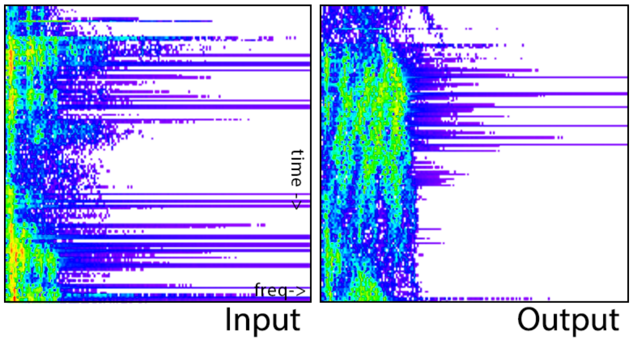

Figure 6: The spectrogram for the `Shush!` preset shows noisy peaks that are created using a wide bandwidth and an oscillating center frequency. The input signal is background chatter in a public area.

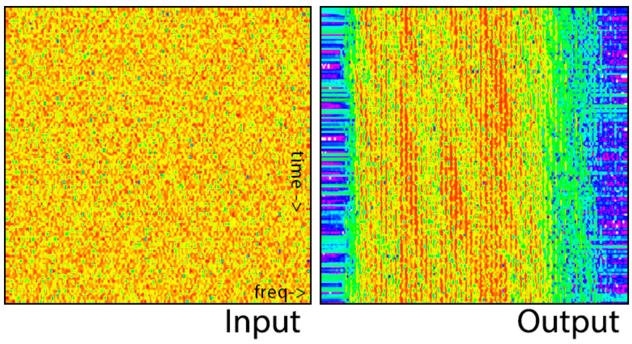

Figure 7: The spectrogram for the `Shepard` tone shows and endless stream of peaks that are constantly rising in frequency. White noise is the input signal.

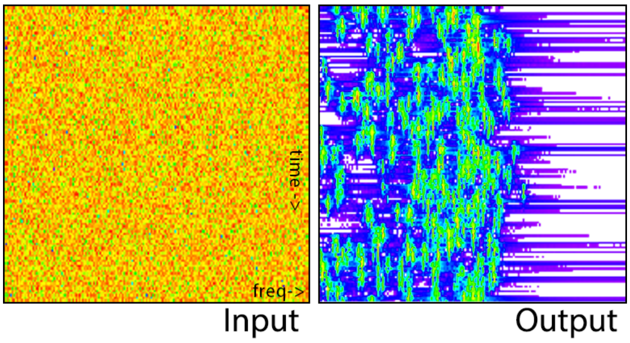

Figure 8: The spectrogram for the `Artifacts` present shows small peaks that quickly decay. The output sounds much like the undesired artifacts that are created when time stretching a signal. White noise is the input signal.

Preset	`Shimmer`	`Shush!`	`Shepard`	`Artifacts`
BW	1	6	1	7
Decay (τ)	0.482s	1.176s	3.0s	0.085s
Thresh. (α)	0.5	0.3	0	0
HPF f_{high}	9kHz	4.7kHz	1.7kHz	10Hz
LPF f_{low}	18kHz	18kHz	20kHz	20kHz
Drift	0.68	0	± 0.2	0
Wander	0	2.0	0	0
LFO Amp	.93	.93	0	0
LFO Freq	5Hz	2.7Hz	0	0
Noise Mix	0 %	0 %	100 %	100 %
Mix	60 %	60 %	100 %	100 %