

Music Visualization Algorithm

Zachary Kieda
Advisor: Jesse Stiles

May 11, 2016

Our music visualization display is a closed form display that is generated with two different parameters. In this report we will give an overview of what the generated output is and the generating function. We will then discuss how the parameters modify the qualitative features of the generated output. In section three we discuss the mathematics behind the visualization that we see, and finally in the fourth section we discuss methods of controlling these two parameters in the context of music generation.

1 Program Output

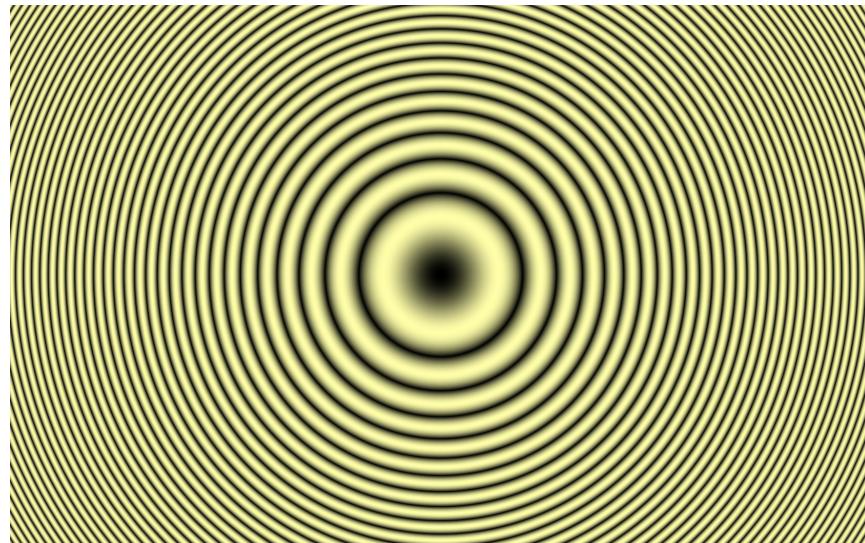


Figure 1: Visualization with $\alpha = .01$, $\lambda = 1$

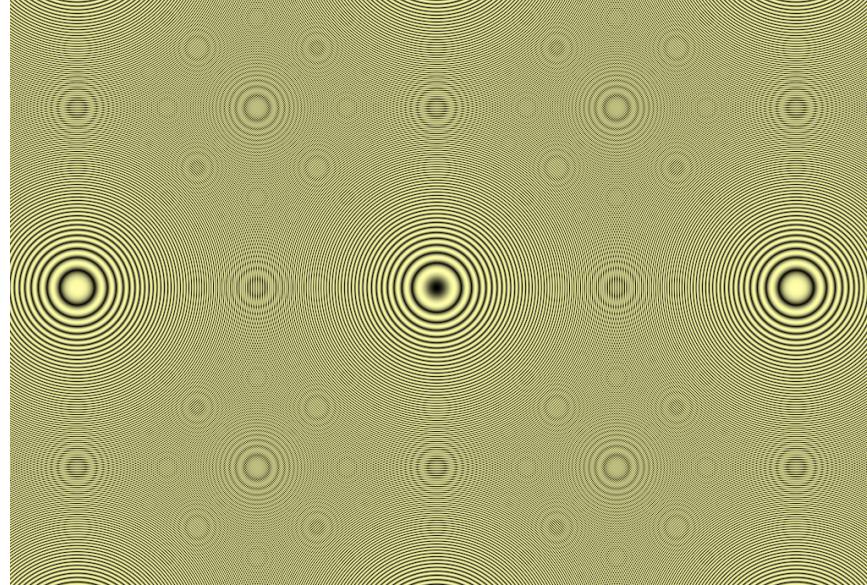


Figure 2: Visualization with $\alpha = .3, \lambda = 1$

The color that we output for each pixel (x, y) we generate a single grayscale color $f(x, y) \in \{k \in \mathbb{N} | 0 \leq k \leq 255\}$. We have example output when zoomed in [Figure 1], and output when zoomed out [Figure 2].

1.1 Generating Function

Let (x, y) be pixels on the screen such that the origin $(0, 0)$ is at the center. Let $\lambda, \alpha \in \mathbb{R}$. The grayscale pattern formed is

$$f(x, y) = \alpha(x^2 + y^2)^\lambda \mod 255$$

Mathematically, this function is uninteresting. If we can view this with infinite precision and $\lambda = 1$, we would merely get a hypergeometric paraboloid that goes back to zero in bands. The spacing of these bands decrease as x^2 and y^2 increase. Indeed, we see this occur when α is very small [Figure 1]. However, we get a more interesting result when α gets large. We start to see the function take on a recursive behavior even though the solution is closed form! As α grows, we see a large variation in behavior – which is interesting coming from a single function.

We note that when $\alpha = 255 + \alpha'$,

$$\begin{aligned} f(x, y) &= (255 + \alpha')(x^2 + y^2)^\lambda \mod 255 \\ &\equiv 255(x^2 + y^2)^\lambda + \alpha'(x^2 + y^2)^\lambda \mod 255 \\ &\equiv \alpha'(x^2 + y^2)^\lambda \mod 255 \end{aligned}$$

Mathematically, we have shown that the function recurses back to its original function when α goes past 255. This is actually not true for very large α due to floating point errors, which will produce variation for large α , but we'll assume that they're roughly equivalent.

1.2 Visualization at various α

We show a variety of figures where $\alpha \approx 11$, $\alpha \approx 50$, $\alpha \approx 150$, and $\alpha = 254.85$. Note that in the last one this is similar to a smaller value of α .

Note: Zoom in to see the details of the image – the details change as you zoom in (and for a good reason, which we'll explain in Section 3).

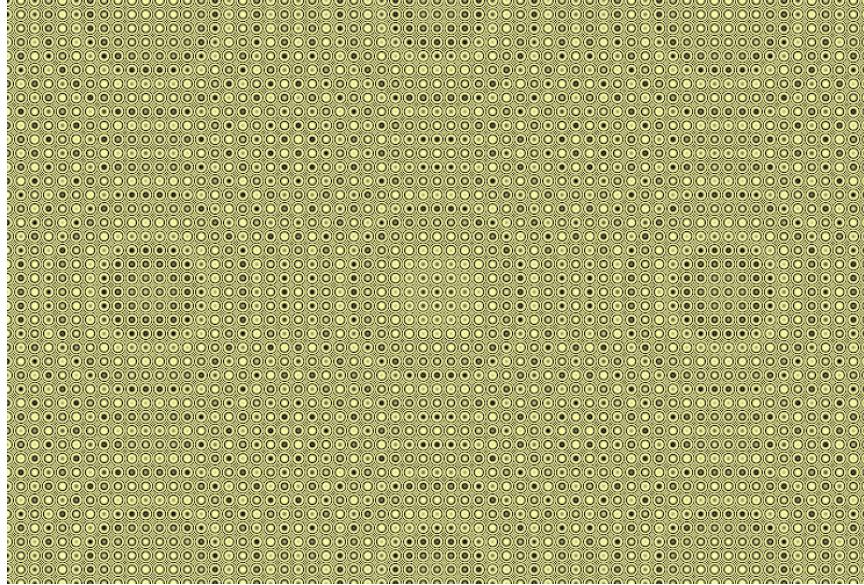


Figure 3: Visualization with $\alpha \approx 11$ and $\lambda = 1$

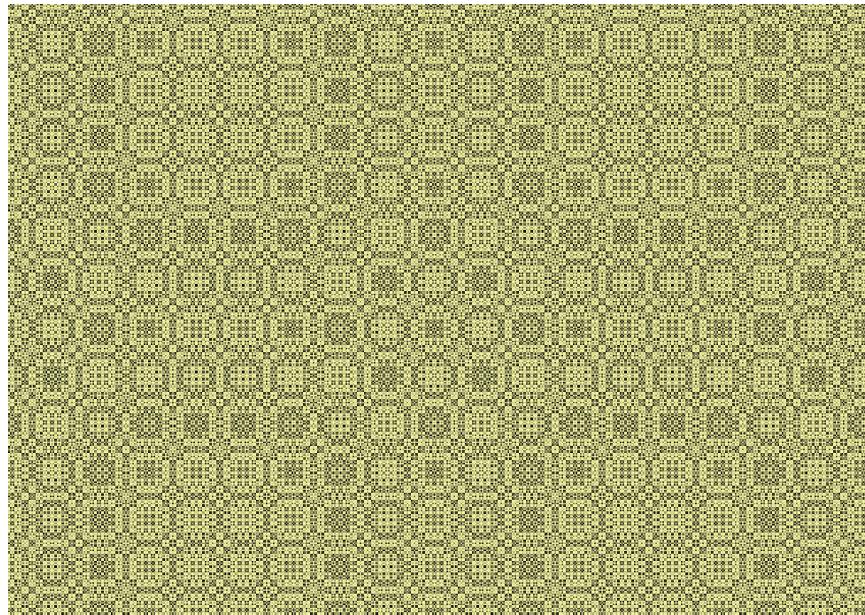


Figure 4: Visualization with $\alpha \approx 50$ and $\lambda = 1$

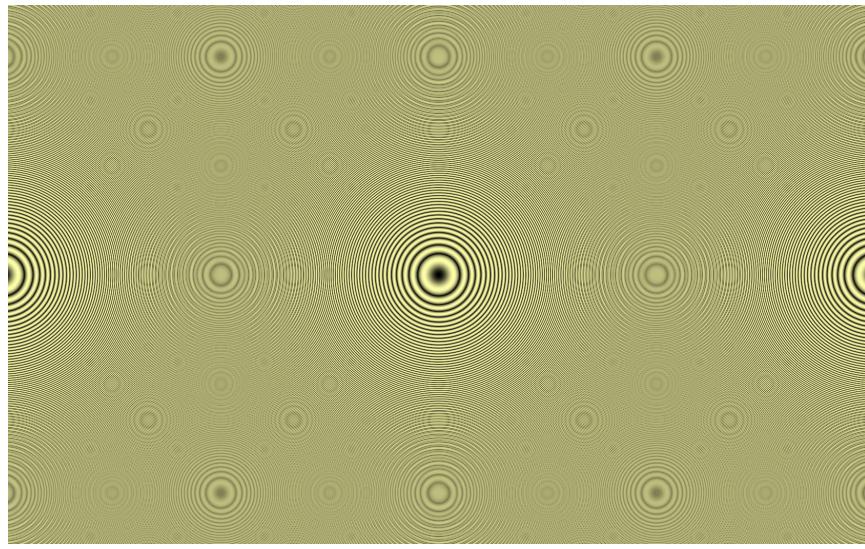


Figure 5: Visualization with $\alpha = 254.85$ and $\lambda = 1$

2 Additional Qualitative Features and the λ Parameter

The λ parameter skews our visualization around the center depending on its value. We define several key intervals

$$\text{Look} = \begin{cases} \text{Inverted} & \lambda < 0 \\ \text{Boring} & \lambda = 0 \\ \text{Concave/Inverted} & 0 < \lambda < .5 \\ \text{Linear} & \lambda = .5 \\ \text{Concave} & .5 < \lambda < 1 \\ \text{Normal} & \lambda = 1 \\ \text{Convex} & 1 < \lambda \end{cases}$$

In our cases when lambda is equal to a particular number, we can easily reason about mathematically. We will not discuss the case **Normal** here, as we have discussed it above.

2.1 Case Boring, When $\lambda = 0$.



Figure 6: Boring. Visualization with $\alpha = .1$ and $\lambda = 0$

We show an example screenshot of when $\lambda = 0$ in [Figure 6].

Note how this scene is rather boring. This is because at this lambda $f(x, y) = \alpha(x^2 + y^2)^0 \bmod 255 = \alpha \bmod 255$. This is in line with the fact that our visualization is uniform.

2.2 Case Linear, When $\lambda = .5$.

This is also an uninteresting case, since the visualization is rather uniform through most values of α . Mathematically, we have $\alpha(x^2 + y^2)^{.5}$ which is equivalent to the distance from $(0, 0)$ to (x, y) .

For smaller α we have a series of evenly spaced rings [Figure 7], then for

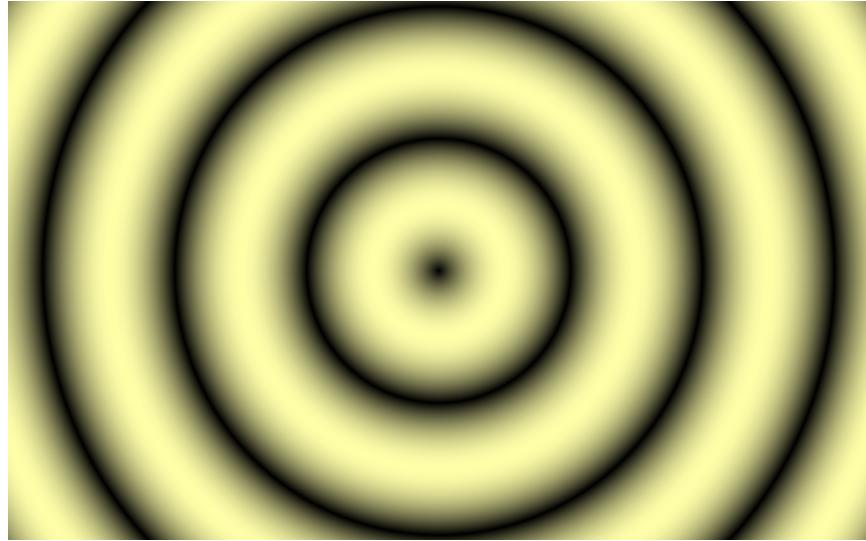


Figure 7: Linear. Visualization with $\alpha = .001$ and $\lambda = .5$

large α the image deforms to the pixels on the screen, causing a mildly amusing pattern [Figure 8].

2.3 Other Cases

For the inverted cases, as we expect, there's a higher amount of detail as we approach the point $(0, 0)$. Note that the concave case makes our function pulls more to the middle, whereas the convex case pushes the middle outwards.

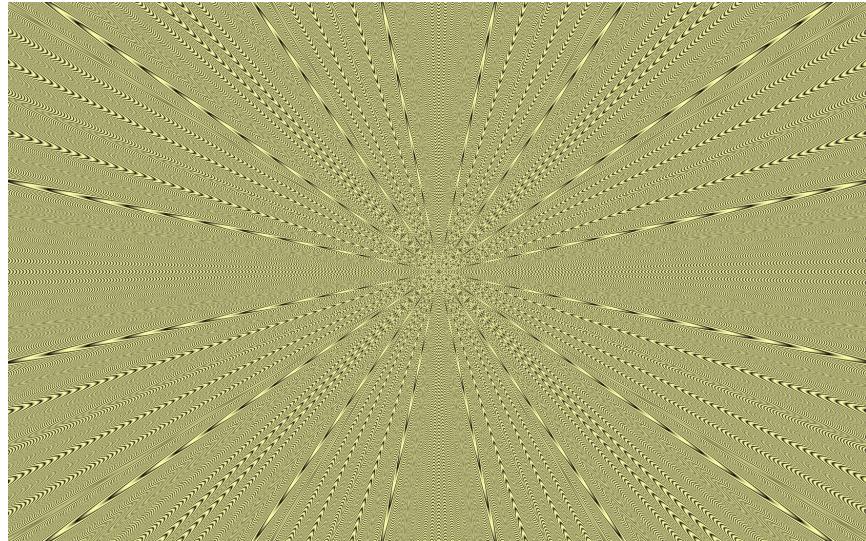


Figure 8: Linear. Visualization with $\alpha = 10$ and $\lambda = .5$

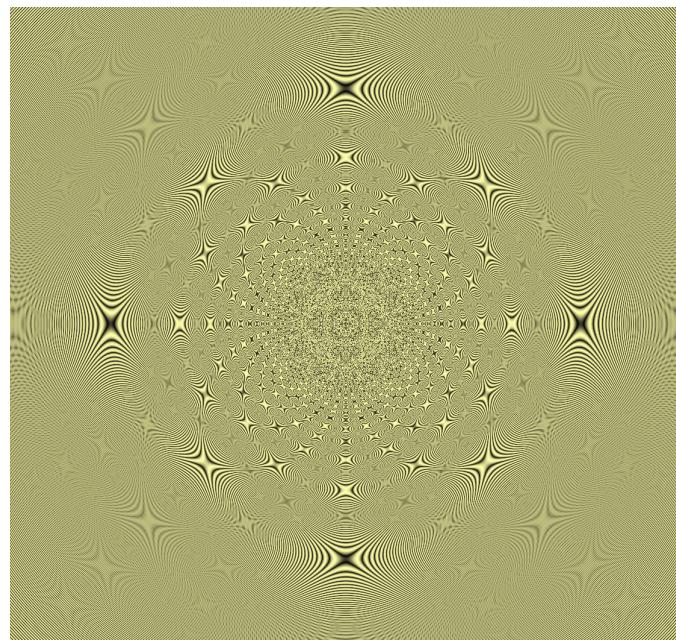


Figure 9: Inverted. Visualization with $\alpha = 1$ and $\lambda = -.5$

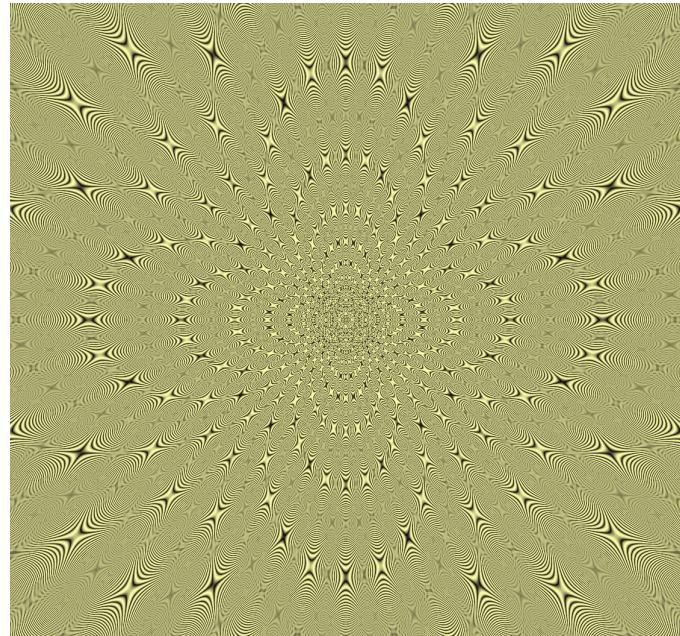


Figure 10: Concave/Inverted. Visualization with $\alpha = 2$ and $\lambda = .35$

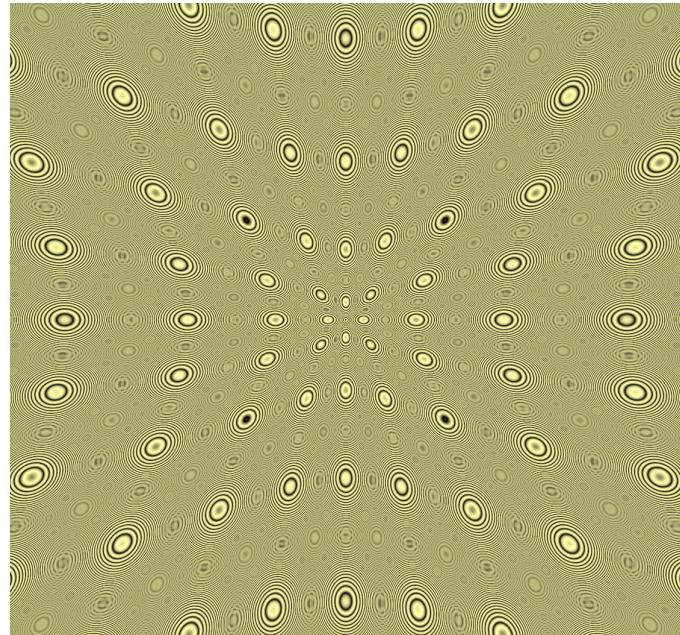


Figure 11: Concave. Visualization with $\alpha = 1$ and $\lambda = .75$

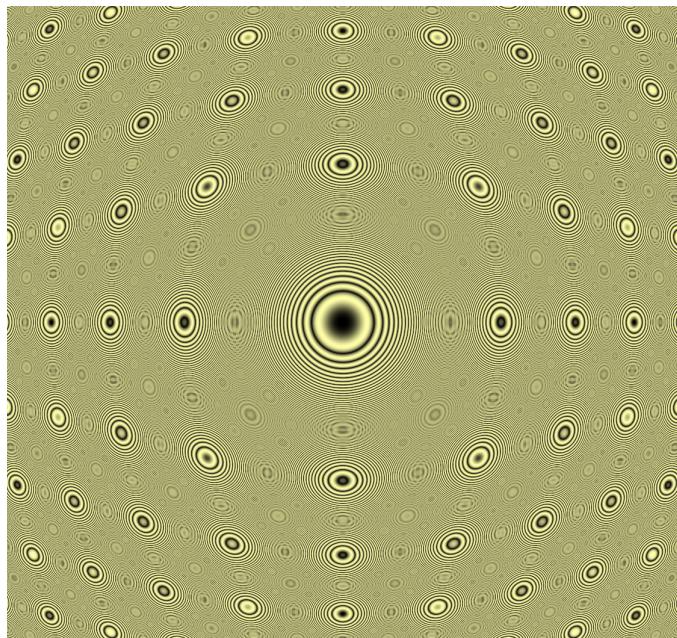


Figure 12: Convex. Visualization with $\alpha = 1$ and $\lambda = 1.4$

3 Mathematical Basis

How can we produce such a variable pattern from such a simple function? How do we get this recursive nature? The answer lies in modular arithmetic, and the way that the function is displayed on the screen using pixels.

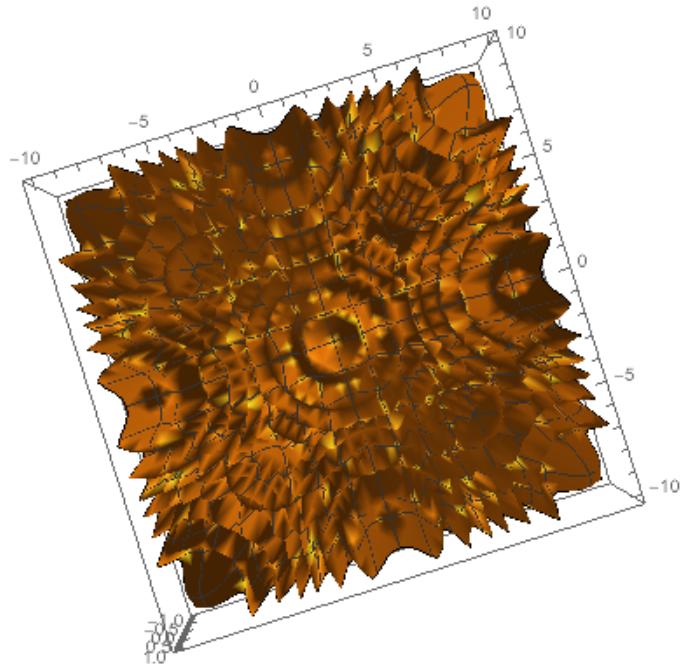


Figure 13: 3D-Plot from $(-10, -10)$ to $(10, 10)$. z values are $f(x, y)$

We created a three dimensional plot of what the output should be with $\alpha, \lambda = 1$. [Figure 13] from $(-10, -10)$ to $(10, 10)$. Then, we have the same graph from $(8, -1)$ to $(10, 1)$ [Figure 14]. If we look closely at the area, they appear completely different in the two plots. This is due to the sampling of the pixels – eventually $\alpha(x^2 + y^2)^\lambda$ becomes so large it has a slope greater than 255 for every pixel. At certain frequencies, the slope is great enough for the function to create another parabola.

Suppose we take the cross-section $f(x, 0) = x^2 \bmod 255$ with $\alpha, \lambda = 1$. Obviously, for $x^2 < 255$ forms a parabola. However, when do we form a new

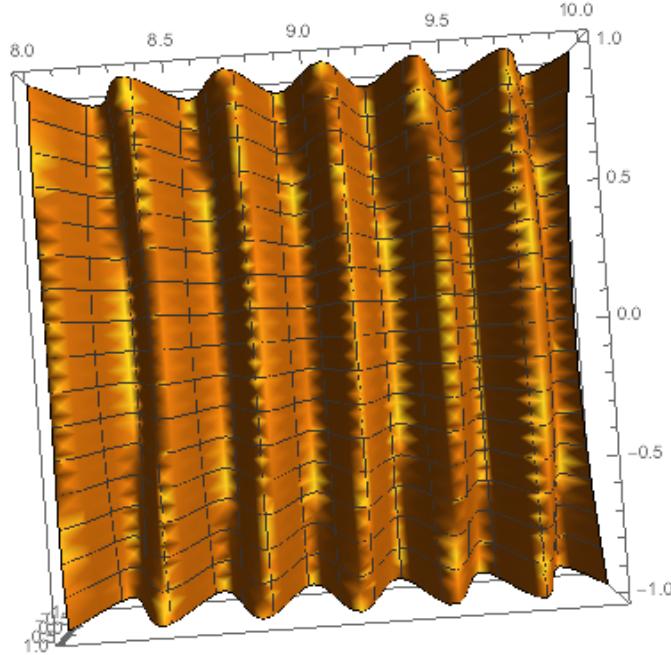


Figure 14: 3D-Plot from $(8 - 1)$ to $(10, 1)$. z values are $f(x, y)$

parabola? Note that the new values only have to line up for $x \in \mathbb{Z}$

$$\begin{aligned} (x - k \cdot 255)^2 \mod 255 &\equiv x^2 - 255k \cdot x + (255k)^2 \mod 255 \\ &\equiv x^2 \mod 255 \end{aligned}$$

Note that this works for $k \in \mathbb{N}$ and integer values of x – otherwise the modulo would not go to zero. However, this implies that every $255k$ pixels we get a new parabola that is exactly the same as the original in the center. This is shown in [Figure 15]. The position of the pixel is represented by the dotted line, the blue line represents our function $f(x, y)$, and the orange line is the parabola that is formed by sampling a position of $f(x, y)$ at every pixel.

Notice that we have two ‘bands’ for every pixel – we call this a 2nd Harmonic. We, in fact, have all integer harmonics (with a high enough α), and we even have half harmonics that are between the two. If you look at [Figure 2] the 0th harmonic is in the center, 1st harmonics to the left and right, and half-harmonics in between. If you zoom up, you will notice that this half harmonic is a combination of the 0th harmonic and the 1st that alternates

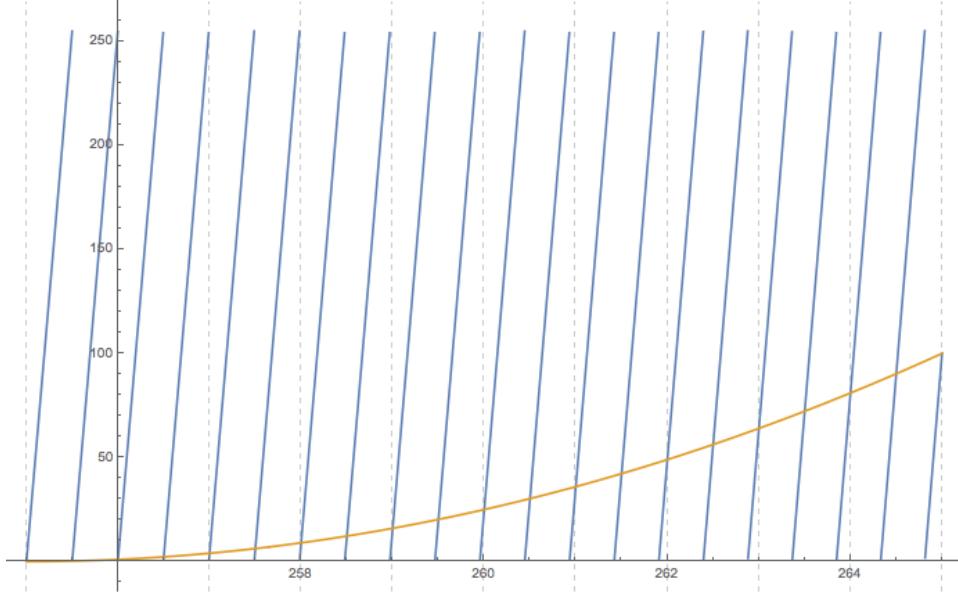


Figure 15: Cross-Section $f(x, 0)$. Note every integer value of the blue line is equal to the integer value of the orange one. The blue line is $f(x, 0)$, and the orange is the new parabola we see on screen.

every pixel.

When α is increased, the non-zero harmonics oscillate as multiple bands move through the pixels. This is important for creating an animation for the visualization. We notice that given an α , there's a minimum in the parabola at $x = 255 - (\alpha - 1)255$. So, as we're increasing α , we reach a new zero minimum zero at $\alpha = k/255 + 1$ when $0 \leq k < 255$ for $x = 255 + k$ and $(\alpha - 1)255$. This implies that if we change alpha to increase by $1/255$ over one second, the first harmonic will make a complete oscillation in one second. This fact is very useful to us in the music generating process, since it will allow us to make the visualization in tune with the music.

4 Controlling

Suppose we are increasing α by a constant rate of $1/255$. As we found before, the n th harmonic will oscillate completely over $1/n$ seconds. This is useful for matching the tempo and beats of the underlying music. For example,

let's suppose our song has a certain BPM , the change in alpha over time for the first harmonic would be an ideal

$$\Delta\alpha = \frac{1/255BPM}{60}$$

Currently our controller only modifies λ to be in the range $.5 \leq \lambda \leq 1.5$, which makes it bulge inwards or outwards. We modify λ based on a floating distribution of the current pitch of a note being played. We have α increase according to the tempo, however, we transition the value of α to a very high, low, or medium value depending on the tempo, placing different tempos in bands with predetermined values of α that we will use. The method for mapping this is discussed in our general project file.