# Color Interpolation inside Wave machines

## Introduction

Wave compiled show presentation files provides information about the LED colors during a timeframe. It indicates the colors in a RGB format (from 0.000 to 1.000) for each specific time in seconds. Nevertheless it does not indicate the values of the colors among those times.

A timeframe could contain the next entry

{

"time":0,

"rgb":[0,0,0]

}, {

"time":100,

"rgb":[1,1,1]

},

A initial approach will be using a linear interpolation, nevertheless the results are not visually pleasing. A better approach is to generate the values using Bezier interpolation.

The Bezier curve is defined by

Where P0 and P3 are the desired initial and final values and P1 and P2 are anchor points that smooth the curve and t is the % of the generated path.

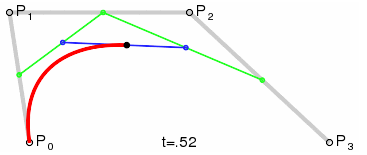


Figure The red line is the path created

The Bezier curve has to be calculated for each component (R, G and B) of the color. All values of Px have a time component (x) that will be defined as Tx and a intensity component (y) which will be defined as Ix. The variable t indicates the % of the paths being generated.

The calculation could be highly time consuming because

* Each component interpolation could take up to 26 float equation. As each color has 3 components there will be required 78 float equations per color interpolation.
* As each Wave has 16 LEDs, there will be 1248 required float equations per Wave interpolation.
* As there will be 24 Wave interpolations (frames) per second, there will be 29952 required floats equations per second.
* The variable t doesn't match the time, but the % of the path being generated, in fact a general Bezier curve might have up to 3 values for each time.

Given the huge amount of required float equations, though optimizations are required, even if the optimized approximation of the Bezier curve is not completely correct.

## Interpolation optimization

The first optimization is to make the intensity of I1 and I2 a multiple of I0 and I3 respectively.

In the case of having I1 = I0 and I2 =I3, the equation can be defined as

Which can be shrink to

Another optimization is to correlate the time component (x) of the generated Bezier value with t variable. This can be done by

* Define T0 as 0 and match it to the frame of the initial interpolation time (s0).
* Define T3 as 1 and match the frame of the final interpolation time (s1).
* Define T1,x as 0.4 and T2,x as 0.6 generates and smooth curve

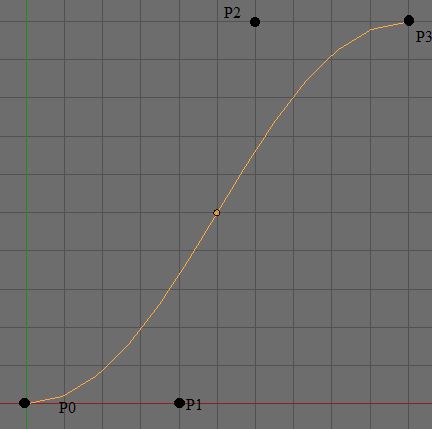


Figure Default interpolation generates smooth curve

* The value of t for each required time component Bt can be found by solving the equation
* By reorganizing the elements, and applying the values the new equation looks like
* By simplifying the equation
* Then a new equation can be defined as

Then solving the information becomes finding the desired t from a specific time component(equation **[2]**), Bt, then applying that t value to equation **[1]** to find the interpolation of the intensity.

* By defining BI0t and BI3t as

The equation in **[3]** can be rewritten as

Finally, because the values of BI0t and BI3t can be pre-calculated and stored in a table. A 10000 entries table will allow all required values visible interpolation of a RGB component values. The table will indicate the Bt and its respective BI0t and BI3t.

Table Recalculated intensity interpolation values

|  |  |  |
| --- | --- | --- |
|  | Bp0t | Bp3t |
| 0.0 | 1.0 | 0.0 |
| ... | ... | ... |
| 1.0 | 0.0 | 1.0 |

The next section will define an algorithm in pseudo code to find any color intensity with the pre-calculated table.

## Interpolation algorithm

The algorithm to generate the interpolation will be

* Given a initial Intensity I0 for a specific time so, a final one, I3 for a second time s1, and the current time st the value for Bt can be easily calculated by
* Because of the linear relation between frames and seconds, the value Bt can be found for any to frames (f1 and f2) and a current frame ft by the equation
* The value Bt will be used to find BI0t and BI3t in the table. With those the intensity calculation becomes

## Implementation

By defining a 0.001 increments in the values of Bt the table can be easy created in a JSON array without explicitly defining Bt, then the JSON Array will follow the next format

[ [ 1.000,0.000] , // BI0t and BI3t for Bt = 0.000

[ 0.999,0.001] , // BI0t and BI3t for Bt = 0.001

...

[ 0.001,0. 999] , // BI0t and BI3t for Bt = 0.999

[ 0.000,1.000] , // BI0t and BI3t for Bt = 1.000

]

## Bibliography

http://en.wikipedia.org/wiki/B%C3%A9zier\_curve