International Journal on Emerging Trends in Electronics & Communication Engineering
Vol. 1, Issue 1 - 2017

COMPARISON OF INTERPOLATION ALGORITHMS FOR PERFORMING SAMPLE RATE CONVERSION OF AN AUDIO RECORDING

SAGAR CHAUDHARY[*], SAURABH SINGH[*]

ABSTRACT

There are many algorithms for performing sample rate conversion of an audio recording. These vary widely in both complexity and in audio quality. The simplest (e.g. 'polynomial interpolation algorithms' commonly used in synthesizers) are very CPU-efficient but do not perform any filtering of the sound before downsampling (converting from a higher to a lower sample rate). Frequencies left from the original recording above the frequency limit of the down-sampled recording will result in very undesirable 'aliasing noise' and so it is therefore very important to filter out these high frequencies before downsampling. The ideal filter would be a 'brick filter' that cuts off everything above the new frequency limit and leaves everything below it intact.

KEYWORDS: Interpolation Algorithms, Sample Rate Conversion, Linear Interpolation, Piecewise Constant Interpolation, Spline Interpolation.

INTRODUCTION TO LINEAR INTERPOLATION

Linear interpolation is simple interpolation technique, where the continuous function is approximated as piecewise-linear by drawing lines between the successive samples. Generally, linear interpolation takes two data points, say (x_a,y_a) and (x_b,y_b), and the interpolant is given by:

$$y = y_a + (y_b - y_a)\frac{(x - x_a)}{(x_b - x_a)}$$

at the point (x,y)

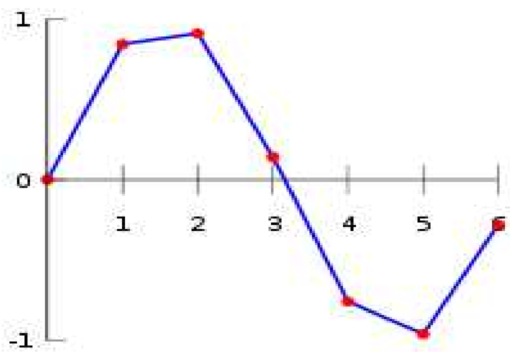

Figure 1.Linear interpolation imposed

[*]Department of Electronics & Communication Engineering, PES Institute of Technology, Banashankari, Bangalore, Karnataka-560085. ***Correspondence E-mail Id:*** editor@eurekajournals.com

Any new samples to be generated between any two given points lies on the line (blue colored) joining the two given points. Linear interpolation is quick and easy, but it is not very precise and its interpolant is not differentiable at the point x_k. The following error estimate shows that linear interpolation is not very precise.

If the function to be interpolated is given by g, and supposes that x lies between x_a andx_b and that g is twice continuously differentiable, then the linear interpolation error is given by

$$|f(x)-g(x)| \le C(x_b-x_a)^2 \quad \text{where} \quad C = \frac{1}{8}\max_{y\in[x_a,x_b]}|g''(y)|.$$

i.e. the error is directly proportional to the square of the distance between the two given points.

Linear interpolation techniques are quick and simple.

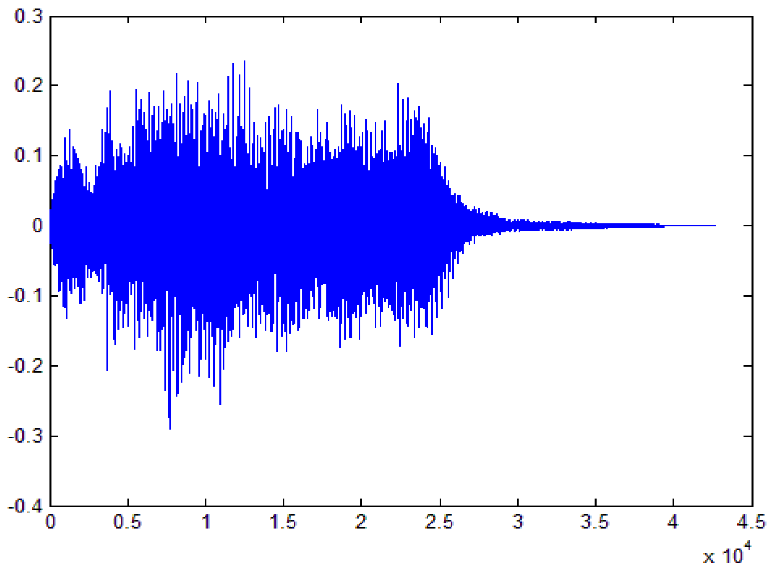

Figure 2.Input wave sample

Linear interpolation of the above input wave is given as below:

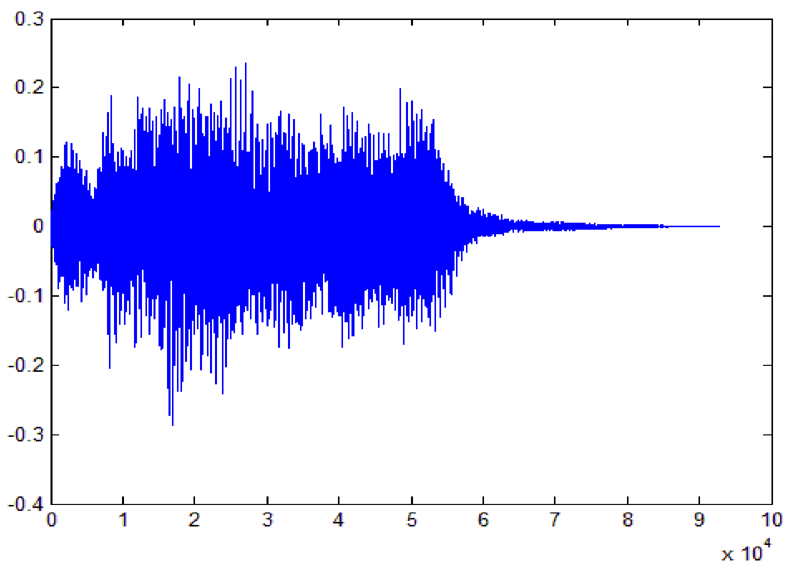

Figure 3.linear interpolation output

PIECEWISE CONSTANT INTERPOLATION

Piecewise constant interpolation is the most straightforward technique, which finds the closest data value and assigns the same value. In one dimensional linear interpolation is preferred to piecewise interpolation though in higher dimensional multivariate interpolation, piecewise is favorable choice because of its simplicity and speed.

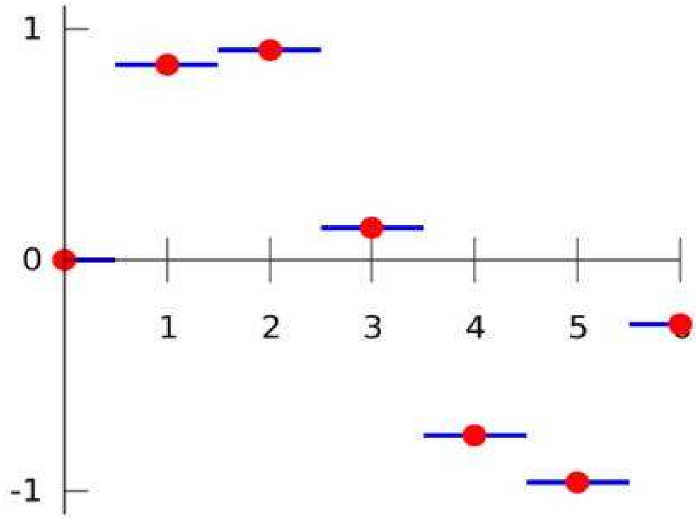

Figure 4.Piecewise interpolation

All the new samples to be generated at the positions on the blue line are given with the red colored value (nearest sample) on the line. Piece-wise interpolation output for the considered input wave is given as:

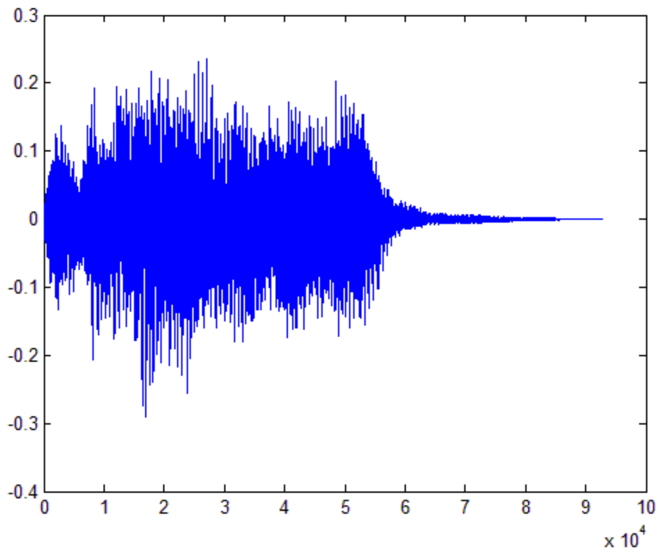

Figure 5.Piece-wise interpolation output

SPLINE INTERPOLATION

Spline interpolation uses the lower degree polynomials in each of the intervals and chooses the polynomial pieces such that they fit smoothly together. The resulting function is called as a spline. Error generates using the high degree polynomials is reduced in this interpolation technique because of the usage of low degree polynomials instead of the high degree polynomials. In this technique the coefficients are found by selecting an initial value and then generating the rest of the coefficients using recurrence relation.

Given (X0,Y0), (X1,Y1),…, (Xn,Yn), fit splines through the data, the splines are given by:

f(x) = a1(x)^2 + b1(x)+c1, *x0≤x≤x1*

a2(x)^2 + b2(x)+c2, *x1≤x≤x2*

an(x)^2 + bn(x)+cn, *xn-1≤x≤xn*

Solving the coefficients of the above equations under different conditions gives the solution.

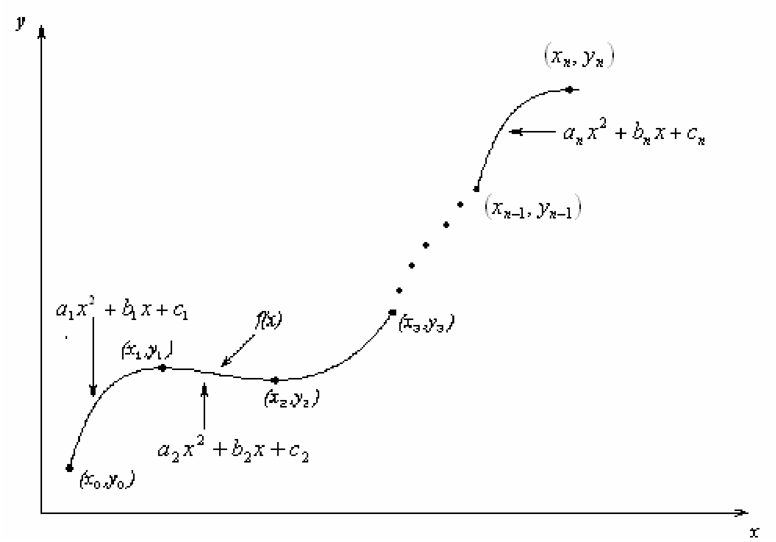

Figure 6.Spline interpolation

Solutions using spline interpolation are given as below:

S(t) = a1(t)^2 + b1(t)+c1, *t0 ≤ t≤t1*

a2(t)^2 + b2(t)+c2, *t1 ≤t ≤ t2*

an(t)^2 + bn(t)+cn, *tn-1 <= t <= tn*

Coefficients of the above equations are obtained by substituting the limit values of't', then differentiating and finally equating them.

Spline interpolation technique incurs less error when compared to that generated by the linear interpolation technique. The output for the considered input wave is given as

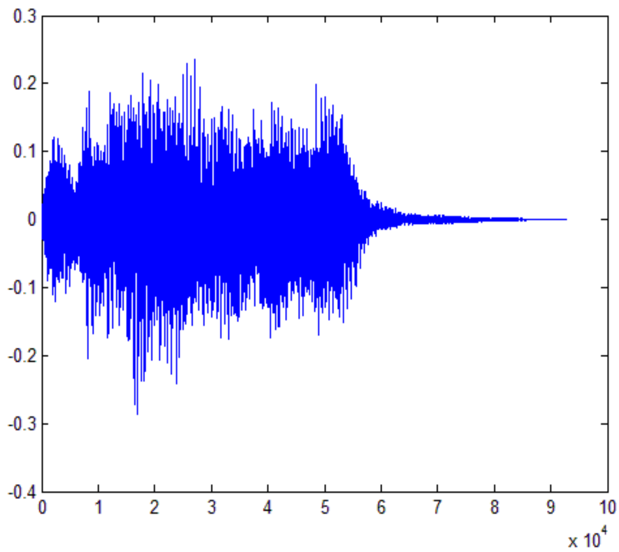

Figure 7.Spline interpolation output

REFERENCES

[1]. R. Crochiere and L. R. Rabiner, Multirate Digital Signal Processing, Englewood Cliffs, NJ:Prentice-Hall, 1983.

[2]. Rothacher, F.M.: “Sample-Rate Conversion: Algorithms and VLSI Implementation”, PhD-thesis, Swiss Federal Institute of Technology, Zurich.

[3]. Smith, O.J.: “The Digital Audio Resampling Home-Page”, http://wwwccrma.stanford.edu/~jos/resample/.

