NON-UNIFORM TO UNIFORM GRID CONVERSION USING LEAST-SQUARES SPLINES

Arrate Muñoz, Thierry Blu and Michael Unser
Biomedical Imaging Group, IOA/DMT
Swiss Federal Institute of Technology Lausanne
CH–1015 Lausanne EPFL, Switzerland
Tel: +41(21)693.51.42; fax: +41(21)693.37.01
e-mail: arrate.munoz@epfl.ch

ABSTRACT

We propose a new technique to perform nonuniform to uniform grid conversion: first, interpolate using nonuniform splines, then project the resulting function onto a uniform spline space and finally, resample. We derive a closed form solution to the least-squares approximation problem. Our implementation is computationally exact and works for arbitrary sampling rates. We present examples that illustrate advantages of our projection technique over direct interpolation and resampling. The main benefit is the suppression of aliasing.

1 Introduction

Non-uniform to uniform grid conversion covers a wide range of applications. Examples of applications are format conversion for display or processing purposes and curve resampling in computer graphics [1]. Another promising application field is signal reconstruction from non-uniformly distributed samples coming from domains such as metrology, biomedical [2], [3], or robotics. Commonly used methods for that reconstruction inludes iterative algorithms, modified Fourier transforms and interpolation.

In this paper, we solve the problem of approximating a non-uniform spline $f(x)$ with a uniform spline of the form $s(x) = \sum_k c_k \beta^n(x-k)$, where $\beta^n(x)$ is the B-spline of degree-n.

The simplest approach would be to sample $f(x)$ uniformly and then compute the spline interpolant of these samples. Its main drawback is the introduction of aliasing when the sampling density is not high enough. A standard remedy is to apply some kind of analog prefiltering prior to sampling.

Here, we propose a more sophisticated approach that is optimal in the sense that the L_2 approximation error between the non-uniform spline and its uniform representation is minimized.

As we will be treating the problem in the continuous domain, we are avoiding the possible ill-posedness of the discrete matrix formulation [4].

We will restrict ourselves to 1-D signals only. It is however possible to extend the method to nonuniform N-D interpolation using thin-plate splines, for instance.

2 Principle of the Approach

From spline sampling theory [5], we know that, in order to approximate a function $f(x)$ on some uniform spline space $V^n = \text{span}\{\beta^n(x-k)\}_{k\in\mathbb{Z}}$, the best is to use the principle of minimum-error signal approximation, that is, to calculate the orthogonal projection of $f(x)$ onto V^n. We may also compute an oblique projection with a corresponding small loss of performance, provided the analysis function is well chosen [6], for instance a spline of degree n_1.

The corresponding projection formula is

$$s(x) \quad = \quad \sum_k \langle f(x), \tilde{\beta}^n(x-k)\rangle_{L_2} \beta^n(x-k), \tag{1}$$

where $\tilde{\beta}^n(x)$ is the analysis function and $\beta^n(x)$ the synthesis function. To have a projector, these functions must be biorthonormal: $\langle \tilde{\beta}^n(x), \beta^n(x-k)\rangle_{L_2} = \delta_k$ [7].

If $\tilde{\beta}(x) \in V^n$, we have an orthogonal projection; otherwise we have an oblique projection.

In practice, it is easier to compute L_2 inner-products with a B-spline β^{n_1} rather than with the analysis function $\tilde{\beta}^n$. Thus, if we define

$$c_k \quad = \quad \langle f(x), \beta^{n_1}(x-k)\rangle, \tag{2}$$

then we can obtain the projection as follows:

$$s(x) = \sum_k (c_k * b^{-(n+n_1+1)})\beta^n(x-k), \tag{3}$$

where $b^{-(n+n_1+1)}$ is an IIR spline filter that has a fast recursive implementation as described in [8, 9].

Thus, the only remaining difficulty is to compute (2); i.e., the inner products of $f(x)$ with B-splines of degree n_1.

3 Nonuniform Splines

Before explaining how to compute these inner products in an efficient and exact manner, we need to introduce some operators and definitions that are helpful to solve our problem.

3.1 Continuous Differential Operators

The conventional derivative operator is

$$Df(x) = \frac{df(x)}{dx}.$$

Its inverse is the integral operator

$$D^{-1}f(x) = \int_{-\infty}^{x} f(t)dt.$$

3.2 Discrete Differential Operators

3.2.1 Non-Uniform Grid

Given the integers i and $n > 0$, we define the divided differences of the polynomial f of order n as

$$[\lambda_i, \ldots, \lambda_{n+i}]f(\lambda) = \sum_{k=i}^{n+i} \frac{f(\lambda_k)}{\left(\prod_{l=i,l\neq k}^{n+i}(\lambda_l - \lambda_k)\right)}$$

3.2.2 Uniform Grid

In the uniform case, we get

$$\begin{aligned} [0, 1, \ldots, n]f(x) &= \frac{1}{n!}\sum_{k=0}^{n}(-1)^k \binom{n}{k} f(x-k) \\ &= \frac{\Delta^n}{n!} * f(x), \end{aligned}$$

were Δ^n is the nth order backward finite difference operator.

3.3 B-Splines

3.3.1 Non-Uniform B-Splines

We call $B_i^n(x)$ the normalized non-uniform B-spline of degree $n > 0$ associated with the knots $\lambda_i, \ldots, \lambda_{i+n+1}$; it is given by

$$B_i^n(x) = (n+1)[\lambda_i, \ldots, \lambda_{i+n+1}](x-\lambda)_+^n$$

with $x_+^n = \max(x, 0)^n$ (one-sided power function).

The support of B_i^n is finite. More precisely, $B_i^n(x) = 0$ if $x \notin [\lambda_i, \lambda_{i+n+1}]$. This is due to the localization properties of the divided differences with respect to polynomials.

We can get the B-spline of degree n from the one of degree $(n-1)$ by applying De Boor's recursion formula [10],

$$B_i^n = \left(\frac{n+1}{n}\right)\left[\frac{x-\lambda_i}{\lambda_{i+n+1}-\lambda_i}B_i^{n-1}(x) + \frac{\lambda_{i+n+1}-x}{\lambda_{i+n+1}-\lambda_i}B_{i+1}^{n-1}(x)\right]$$

3.3.2 Uniform B-Splines

The uniform B-splines are usually defined using finite differences instead of divided ones. The centered B-splines, which appear in (1), are given by

$$\beta^n(x) = \Delta^{n+1} * \frac{x_+^n}{n!} * \delta\left(x + \frac{n+1}{2}\right).$$

Using the equivalence

$$\frac{x_+^n}{n!} * f(x) = D^{-(n+1)}f(x),$$

the uniform B-splines can be expressed as

$$\beta^n(x) = \Delta^{n+1} * D^{-(n+1)} * \delta\left(x + \frac{n+1}{2}\right).$$

4 Computing Inner Products

As we said before, the difficulty of the method is to compute the inner product of the non-uniform spline with a uniform one. The key formula is derived easily from the uniform B-spline definition

$$\begin{aligned} c_k &= \langle f, \beta^{n_1}(x-k)\rangle \\ &= \Delta^{n_1+1} * D^{-(n_1+1)}f\left(k + \frac{n_1+1}{2}\right). \end{aligned}$$

In other words, provided that we have the uniform samples of $D^{-(n_1+1)}f\left(k + \frac{n_1+1}{2}\right)$, we can compute the B-spline coefficients c_k by simple digital filtering. So, $c_k = (h * I)(k)$, where $I(k) = D^{-(n_1+1)}f\left(k + \frac{n_1+1}{2}\right)$, and $H(z) = (1-z)^{n_1+1}$ ((n_1+1)th finite difference filter).

Thus, it is of critical importance to be able to compute the (n_1+1)-fold integral of a non-uniform spline. The recursive formula we use for the calculation is derived from recursion (4) and integration formulas of a non-uniform spline [11]. Specifically,

$$D^{-1}B_i^{n-1}(x) = \sum_{j\geq i} \frac{\lambda_{j+n+1}-\lambda_j}{n+1}B_j^n(x). \tag{4}$$

5 Practical Implementation of the Algorithm

The main steps in the algorithm are summarized below and shown in Fig. 1.

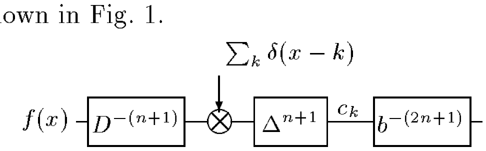

Figure 1: General scheme for the method (orthogonal projection)

- **Interpolation**

  Either the spline $f(x) = \sum_i a_i B_i^n(x)$ is known, or it is specified by non-uniform samples of $f(x)$ at knot positions λ_k. In this latter case, the coefficients a_i of the interpolation are the solution of a band-diagonal system of equations.

- **Integration**

  In Sections 3 and 4, we have presented all the tools we need to calculate the $(n_1 + 1)$-fold integral of $f(x)$. We write

$$D^{-1} f(x) = \sum_i a_i D^{-1} B_i^n(x) = \sum_i a_i^{(-1)} B_i^{n+1}(x)$$

  with

$$a_i^{(-1)} = \frac{\lambda_{i+n+2} - \lambda_i}{n+2} \sum_{j \geq i} a_j \qquad (5)$$

  We then define

$$I(x) = D^{-(n_1+1)} f(x) = \sum_i a_i^{-(n_1+1)} B_i^{n+n_1+1}(x)$$

  The basic idea is that the $(n_1 + 1)$-fold integral of a non-uniform spline of degree n is a non-uniform spline of degree $(n + n_1 + 1)$.

  The coefficients $a_i^{-(n_1+1)}$ are pre-computed by recursive application of (5). Likewise, we may also update the B-spline basis functions by using De Boor's recursion formula (4).

- **Sampling**

  The next step is the resampling of the above integral at the points $\left(k + \frac{n_1+1}{2}\right)$

$$\begin{aligned} I(k) &= D^{-(n_1+1)} s(x) \,|_{x=k+\frac{n_1+1}{2}} \qquad (6) \\ &= \sum_i a_i^{-(n_1+1)} B^{n_1+n+1}\left(k + \frac{n_1+1}{2}\right). \end{aligned}$$

- **Digital Filtering**

  At this stage we apply (n_1+1) centered finite differences corresponding to the filter with z-transform $\Delta^{n_1+1} \leftrightarrow (1-z)^{n_1+1}$. Finally, we postfilter with the IIR filter $b^{-(n+n_1+1)}$ specified in (3).

6 Examples

The goal of this section is to point out the improvement we get in terms of aliasing suppression by using our projection method over standard interpolation and resampling.

We use as test a signal composed of two well differentiated components: a high frequency supperimposed onto a low frequency one. The signal is a cubic nonuniform spline (Fig. 2).

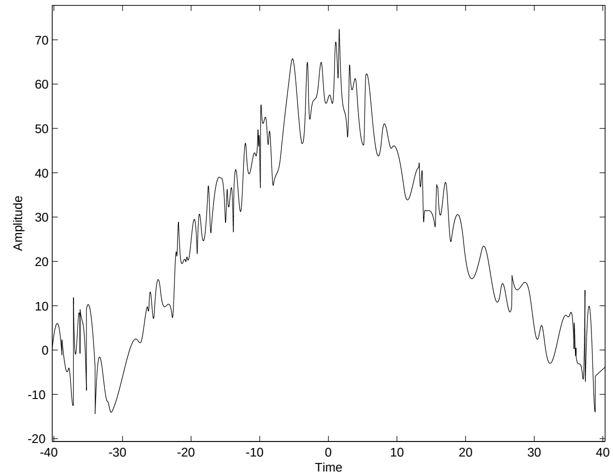

Figure 2: Test signal which is a non-uniform cubic spline.

When we just resample it at uniform sample locations and interpolate it with a uniform spline, we observe that the spline curve is constrained to pass through the sampling points (Fig. 3).

When the projection method is applied, the resulting uniform spline curve tries to adjust itself to minimize the difference with the nonuniform spline curve leading to a much better performance in terms of L_2-error as shown in Fig. 4.

The sampling points for the uniform interpolation are the same in both cases.

7 Conclusions

We have presented a refined tool to convert non-uniform splines into uniform ones. The approach is efficient computationally; its complexity per output point is constant and independent of the knot spacing of the input signal. It is easy to check that the cost of an oblique projection into V^n perpendicular to V^{n_1} is approximately equivalent to a (non-uniform) spline interpolation of degree $(n + n_1 + 1)$.

Experimental results show that our method outperforms standard interpolation as our solution is optimal in the least-squares sense. Moreover, the implicit analog prefiltering step in our approximation formula (1) reduces aliasing.

Acknowledgment:

This work was supported by the Swiss National Science Foundation under grant $2100-053540$.

References

[1] Martin Roche and Sheri Beusan, "A nonuniform b-spline input interpolation procedure," *Intelligent Instruments & Computers*, vol. 7, pp. 9–17, January/February 1987.

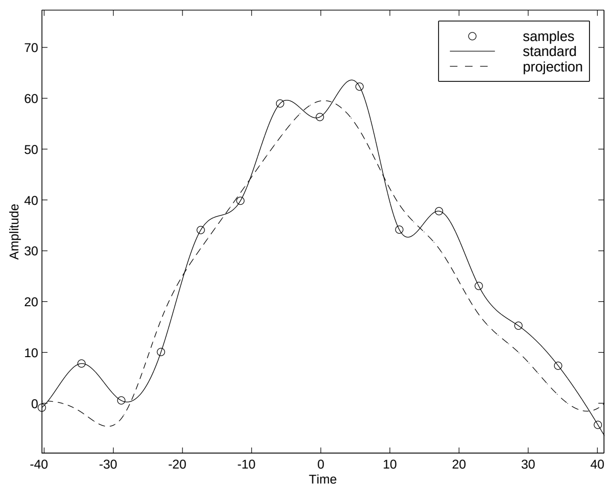

Figure 3: Approximation of test signal in Fig. 2 by the standard (L_2-error= 49.7) and projection method (L_2-error= 23.75.) using a uniform sampling step size of 5.7 s

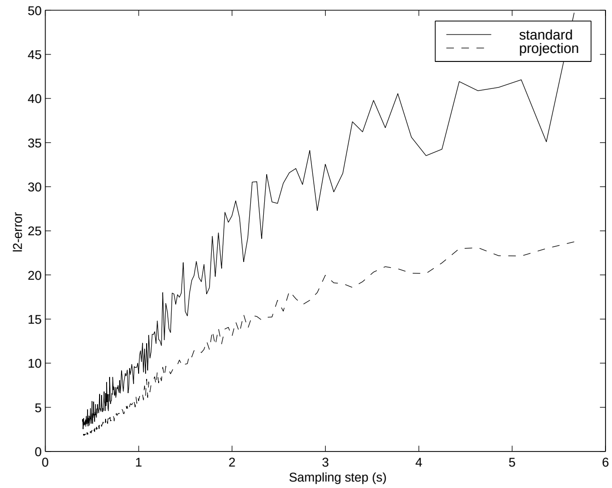

Figure 4: Comparison of the L_2 error for the standard and projection method for different uniform sampling steps.

[2] K. Schreibman, C. Thomas, and M. Levy, "Spectral analysis of cardiac cycle length variations: Resampling overcomes effects of nonuniform spacing," in *Engineering in Medicine and Biology Society 11th. Annual International Conference*, 1989.

[3] C. O'Bara and S. Reisman, "Evaluation of step interpolation of systolic arterial blood pressure signals," in *Proc. IEEE 22nd. Bioengineering Conference*, 1996, pp. 105–106.

[4] M.O. Ahmad, E.I. Plotkin, M.N.S. Swamy, and J.D. Wang, "An iterative procedure for uniform resampling of a 2-D signal from its nonequally-spaced samples," in *Proc. IEEE International Symposium on Circuits and Systems*, 1989, vol. 2, pp. 1087–1090.

[5] Michael Unser, Akram Aldroubi, and Murray Eden, "Polynomial spline signal approximations: Filter design and asymptotic equivalence with Shannon's sampling theorem," *IEEE Transactions on Information Theory*, vol. 38, pp. 95–103, February 1992.

[6] Michael Unser, "Approximation power of biorthogonal wavelet expansions," *IEEE Transactions on Signal Processing*, vol. 44, pp. 519–527, March 1996.

[7] Michael Unser, "Sampling—50 years after Shannon," *Proceedings of IEEE*, April 2000.

[8] Michael Unser, Akram Aldroubi, and Murray Eden, "B-spline signal processing: Part I—Theory," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 821–832, February 1993.

[9] Michael Unser, Akram Aldroubi, and Murray Eden, "B-spline signal processing: Part II—Efficient design and applications," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 834–848, February 1993.

[10] C. De Boor, "On calculating with B-splines," *Journal of Approximation Theory*, vol. 6, pp. 50–62, January 1972.

[11] C. De Boor, *The Spline Toolbox for Matlab*, 1997.