

A Fast and Accurate Method for Numerical Integration of Bandwidth-Limited Signals

Mary S. Wei

mary.sj.wei@gmail.com

Quantica Computing LLC <https://orcid.org/0009-0002-0168-972X>

Research Article

Keywords: Numerical integration, Nyquist-Shannon sampling, Newton-Cotes formulas, fast Fourier transform

Posted Date: September 22nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-7801844/v2>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

A Fast and Accurate Method for Numerical Integration of Bandwidth-Limited Signals

Mary S. Wei*

Quantica Computing LLC, San Jose, California

* *mary.sj.wei@gmail.com*

Abstract: Fast and accurate integration of images or signals is important in many applications. This paper reports a novel split-band integrator (SBI) that is especially suitable for integrating discrete images representing a bandwidth-limited signal sampled above its Nyquist frequency. The SBI combines the advantages of the classical Newton-Cotes formulas (NCFs) and a fast Fourier transform (FFT) based integrator while avoiding their drawbacks. It works by splitting an input into two images and processing them separately then combining the results, where one image containing high-frequency components is integrated using FFT, whereas the other collecting the remaining low-frequency portion is integrated accurately by a classical NCF. Both theoretical derivations and numerical tests are presented to demonstrate the SBI.

1 Introduction

Herein the terms “image” and “signal” are used interchangeably to refer to a physical quantity distributed in an original domain of space, time, or spacetime. The concept and methods of an integral image, also known as a summed-area table, were introduced to the fields of computer graphics [1] and computer vision [2] for fast computation of the integral of an image $i(x, y)$ in any rectangle region $ABCD$ with $A = (x_1, y_1)$, $B = (x_2, y_1)$, $C = (x_2, y_2)$, $D = (x_1, y_2)$ as shown in Fig. 1. The basic idea is to have an integral image

$$I(x, y) \stackrel{\text{def}}{=} \int_0^y \int_0^x i(x', y') dx' dy' \quad (1)$$

precomputed and stored in an array for lookup, so that later on the regional integral in any rectangle $ABCD$ can be quickly computed via just four array look-ups and three additions as

$$\int_{y_1}^{y_2} \int_{x_1}^{x_2} i(x, y) dx dy = I(x_2, y_2) - I(x_1, y_2) + I(x_1, y_1) - I(x_2, y_1). \quad (2)$$

Fast and accurate computation of integral images is also useful in many other applications such as reconstructing a 3D height profile of a surface from a measured 2D gradient field [3], and rasterizing a grayscale image by integrating a gradient field of edges created from a vector graphics [4]. Another important application of numerical integration is in inertial navigation using accelerometers and gyroscopes, where accelerations are doubly integrated into positions [5].

A computer image or digital signal is usually a numerical array representing discrete samples of a continuous signal or field in the physical world, where the discrete sampling might incur a loss of information or a distortion of the real signal. Fortunately, when the continuous signal is bandwidth-limited, it can be perfectly represented by and recovered from a discretely sampled image without any loss, so long as the rate of sampling exceeds the bandwidth [6], in which case, the discrete image is said to be a sufficiently sampled representation of the continuous image, or just called *sufficiently*

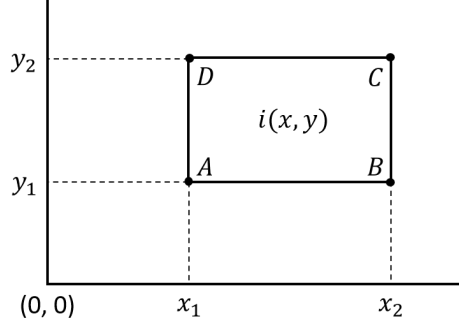


Fig. 1: A precomputed integral image enables fast evaluation of integration of an interested image $i(x, y)$ over a given rectangular region $ABCD$.

sampled in short. For any discretely sampled signal, the frequency commensurate to half of its sampling rate is called the Nyquist frequency. A sufficiently sampled image is called *critically sampled*, when the Nyquist frequency is barely sufficient, only slightly over half of the bandwidth of the continuous signal. Given a sufficiently sampled image representing an original bandwidth-limited continuous signal, it is desirable to compute an integral image that is sufficiently sampled and representing an antiderivative of the original continuous signal. However, straightforward integration of a sufficiently and critically sampled image via classical Newton-Cotes formulas (NCFs) [7] induces systematic errors, and worse yet, violates the symmetry of shift invariance. Upsampling the image before applying NCFs can improve the accuracy, but that entails a greatly increased computational cost and does not rid the problem entirely.

Then there have been proposals of using the Fourier transform to realize integration as an $(j\omega)^{-1}$ filter in the Fourier domain, with $\omega \stackrel{\text{def}}{=} 2\pi f$ being an angular frequency and $j \stackrel{\text{def}}{=} \sqrt{-1}$, which take advantage of the fast Fourier transform (FFT) and in principle can compute integral images exactly without a systematic error [8–10]. Unfortunately, in practice, such an FFT-based integrator suffers a severe numerical instability due to the problem of dividing by near-zero values at low frequencies, and requires an arithmetic processor to handle a wide dynamic range of signal amplitudes when the number of sample points is large. Such a numerical instability or wide dynamic range is problematic when the signal being integrated comes from a noisy measurement, the length of the data sequence being integrated is exceptionally long, or the computation is done with a limited arithmetic precision by an embedded system or a neural network operating at a reduced-precision [11, 12]. Besides, it is necessary to have an image satisfying the periodic boundary condition in order to apply any FFT-based method of signal processing.

To overcome the problems associated with the existing integrators, this paper reports a novel split-band integrator (SBI) that should be fast and runs at a similar speed as the classical rectangle or trapezoid quadrature rule, while producing a highly accurate, sufficiently sampled integral image without incurring the problem of numerical instability plaguing the previous FFT-based integrators. The fundamental principle underlying my solution is a split-band processing methodology, which splits an input image into a high-frequency (HF) and a low-frequency (LF) bands, uses FFT-based integration for the HF band, and employs a classical NCF or an equivalent to process the LF band in an accurate and shift-invariant manner.

The rest of my report is organized as follows. Section 2 analyzes the classical NCFs applied to sufficiently sampled images to illustrate why they incur systematic errors and violate shift invariance, and section 3 introduces boundary treatments to render an input image suitable for FFT-based

signal processing. Then section 4 presents the novel SBI, and section 5 reports numerical tests and validations of the SBI. Finally, section 6 summarizes the paper.

2 Problems with classical NCFs

Given a discrete image $u(\mathbf{n})$, $\mathbf{n} \in \mathbb{Z}^d$ that is sufficiently sampled from a continuous signal $u_c(\mathbf{r})$, $\mathbf{r} \in \mathbb{R}^d$ over a d -dimensional bounded region of the original domain, $d \in \mathbb{N}$, the Nyquist-Shannon sampling theorem says that $u_c(\mathbf{r})$ can be perfectly reconstructed from $u(\mathbf{n})$ through the Whittaker-Shannon interpolation formula [6]. $u(\mathbf{n})$ being sufficiently sampled implies that $u_c(\mathbf{r})$ is bandwidth-limited. The Nyquist-Shannon sampling theorem enables many methods for discrete-time processing of continuous-time signals, including theoretically exact differentiation [6, 10] and integration [8–10] of bandwidth-limited signals. Exact and stable differentiation of a signal in one dimension is realized straightforwardly by multiplying a transfer function $j\omega$ in the Fourier domain.

By contrast, integration corresponds to dividing the transfer function $j\omega$ in the Fourier domain, which, although is exact in theory, suffers in practice a severe numerical instability at low frequencies, that is a serious issue when dealing with noisy measurement signals or using reduced-precision arithmetic. Thus, many applications prefer a classical NCF to a straightforward FFT-based integrator. However, as being demonstrated and analyzed before [9, 13], integration by the classical NCFs corresponds to finite impulse response (FIR) filters which are inherently unable to realize the $(j\omega)^{-1}$ behavior of an ideal integrator, thus perform poorly at high frequencies. To appreciate the difficulty from another angle, note that the classical NCFs are designed to deal with smooth and slowly changing signals, for which adjacent sampling points fit nicely into a low-order polynomial, whereas the assumption becomes increasingly ill-suited when the signal frequency turns higher and approaches the Nyquist frequency.

Therefore, the classical NCFs are mathematically inexact integrators for sampled signals even using exact or arbitrary-precision arithmetic. Such a loss of exactitude can be seen from an example of a simple signal with a known exact integral. Take a continuous signal $F(t) = \sin(k_0 t)$ with $k_0 \in \mathbb{N}$, $t \in [0, 2\pi)$ for instance, whose analytic, exact, definite integral is $G(t) \stackrel{\text{def}}{=} \int_0^t \sin(k_0 u) du = k_0^{-1}[1 - \cos(k_0 t)]$. Numerically, the function $F(\cdot)$ may be sufficiently sampled into a sequence of $N \geq 2k_0 + 1$ points $\{F_n \stackrel{\text{def}}{=} \sin(2\pi k_0 n/N)\}_{n=0}^{N-1}$, whose discrete Fourier transform is $\tilde{F}_k = N[\delta(k - k_0) - \delta(k + k_0)]/2j$, $k \in [0, N-1]$, which becomes $\tilde{G}_k = (-N/k_0)[\delta(k - k_0) + \delta(k + k_0)]/2$ after divided by a frequency sequence $\{j\omega_k \stackrel{\text{def}}{=} j2\pi f_k\}_{k=1}^{N-1}$ with $\omega_k \stackrel{\text{def}}{=} k$, $\forall k \in [0, N/2)$ and $\omega_k \stackrel{\text{def}}{=} k - N$, $\forall k \in [N/2, N-1]$, so to implement a numerical integration exactly. It is easily seen that the inverse Fourier transform of $\{\tilde{G}_k\}_{k=0}^{N-1}$ produces a sequence $\{G_n = C - k_0^{-1} \cos(2\pi k_0 n/N)\}_{n=0}^{N-1}$ with C being an arbitrary constant, which matches exactly the discrete samples of the exact integral $G(t)$, $t \in [0, 2\pi)$ with $C = k_0^{-1}$. By contrast, the classical NCFs could not yield a discretely sampled version of the exact integral in general.

Specifically, if using the right hand rectangle rule, one sums the sequence $\{F_n\}_{n=0}^{N-1}$ iteratively and gets a discrete sequence $\{R'_n\}_{n=0}^{N-1}$, with $R'_0 \stackrel{\text{def}}{=} 0$, $R'_n \stackrel{\text{def}}{=} R'_{n-1} + F_n$, $\forall n \in [1, N-1]$, namely,

$$\begin{aligned} R'_n &= (2\pi/N) \sum_{m=1}^n \sin(2\pi k_0 m/N) \\ &= (\pi/N) [\sin(\pi k_0/N)]^{-1} \{ \cos[2\pi k_0(1/2)/N] - \cos[2\pi k_0(n + 1/2)/N] \}, \quad \forall n \in [0, N-1], \end{aligned} \quad (3)$$

which obviously suffers a $t = 1/2$ time shift with respect to the exact integral. Such a timing offset can be removed rigorously by an FFT-based operation of fractional time shift [6, 10], followed by

an adjustment of integral constant to make the first element 0, yielding a time-aligned sequence

$$R_n = (\pi/N) [\sin(\pi k_0/N)]^{-1} [1 - \cos(2\pi k_0 n/N)], \quad \forall n \in [0, N-1]. \quad (4)$$

Still though, the resulted $\{R_n\}_{n=0}^{N-1}$ differs from the exact integral $\{G_n\}_{n=0}^{N-1}$ by a frequency-dependent prefactor $(\pi/N) [\sin(\pi k_0/N)]^{-1} \neq k_0^{-1}$, although the relative difference is small for a small $k_0 \in \mathbb{N}$ of frequency harmonic. Alternatively, one might use the trapezoid rule and get $T_0 \stackrel{\text{def}}{=} 0$, $T_n \stackrel{\text{def}}{=} T_{n-1} + (F_{n-1} + F_n)/2$, $\forall n \in [1, N-1]$, that is,

$$\begin{aligned} T_n &= (2\pi/N) \sum_{m=1}^n \frac{1}{2} \{ \sin[2\pi k_0(m-1)/N] + \sin(2\pi k_0 m/N) \} \\ &= (\pi/N) [\tan(\pi k_0/N)]^{-1} [1 - \cos(2\pi k_0 n/N)], \quad \forall n \in [0, N-1], \end{aligned} \quad (5)$$

which is time-aligned with the exact integral $\{G_n\}_{n=0}^{N-1}$, but still not quite the same, due to a different frequency-dependent prefactor $(\pi/N) [\tan(\pi k_0/N)]^{-1} \neq k_0^{-1}$. Now it is clear that the classical NCFs will not produce an accurate integral for a general input signal containing HF components, because the numerical integrals of different sinusoidal harmonics indexed by a $k_0 \in \mathbb{N}$ suffer different deviations from the ideal k_0^{-1} prefactor.

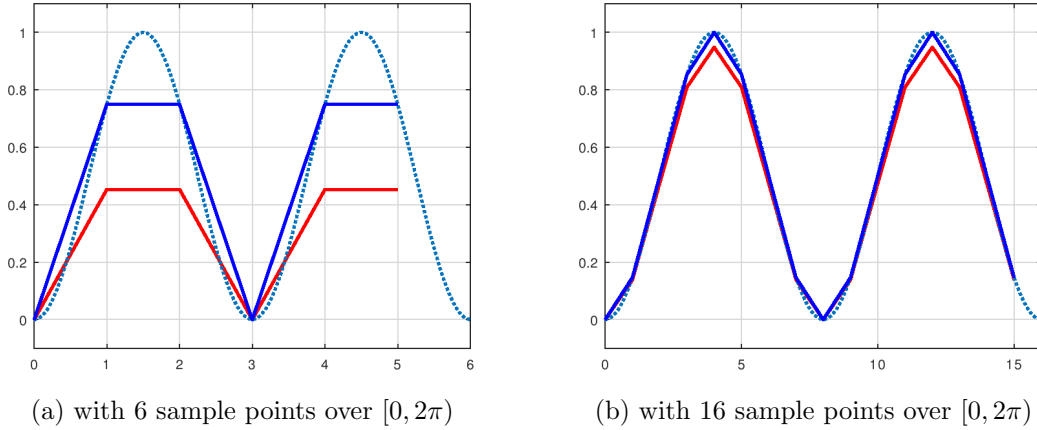


Fig. 2: Classical trapezoid (red) and exact (blue) integrals of $\sin(2t)$, $t \in [0, 2\pi]$. A continuous curve of the exact integral $[1 - \cos(2\pi k_0 n/N)]/2$ is also shown in dotted lines for eye guidance.

Fig. 2 plots and compares the analytically exact integral with results from a classical trapezoid rule for a test signal $\sin(k_0 t)$ with $k_0 = 2$ and $t \in [0, 2\pi]$, which requires at least 5 points to be sufficiently sampled in accordance with the Nyquist-Shannon theory. Fig. 2(a) shows that the classical trapezoid rule suffers a large error when the signal is close to critically sampled with 6 points only, while Fig. 2(b) illustrates how the error is significantly reduced, although still clearly visible, when the signal is upsampled with 16 points.

The above analysis and example not only reveal the reason why the classical NCFs become inaccurate for integrating HF signals, but also inspire methods to combine such rules and achieve higher accuracy for integrating LF signals, where the differences of both $\{R_n\}_{n=0}^{N-1}$ and $\{T_n\}_{n=0}^{N-1}$ from $\{G_n\}_{n=0}^{N-1}$ are as small as $O((k_0/N)^3)$ when k_0/N is small. In view of $\sin(x) = x - x^3/6 + O(x^5)$ and $\tan(x) = x + x^3/3 + O(x^5)$ for small $x \in \mathbb{R}$, it would make an excellent integrator by weighting and combining $\{R_n\}_{n=0}^{N-1}$ and $\{T_n\}_{n=0}^{N-1}$ into a new integral sequence $\{RRT_n\}_{n=0}^{N-1}$ with

$RRT_n \stackrel{\text{def}}{=} (R_n + R_n + T_n)/3, \forall n \in [0, N-1]$. The new integrator is called the rectangle-rectangle-trapezoid (RRT) rule, whose discretization error is $O((k_0/N)^5)$ for sinusoidal harmonics with a small $k_0 \in \mathbb{N}$. The error scaling of the RRT rule parallels that of Simpson's rule [7], and the RRT is roughly 16 times more accurate than Simpson's, due to the FFT-based 1/2 grid time shift, which essentially does exact signal interpolation at the midpoints of the original grid. Indeed, one implementation of the RRT integrator is to $2\times$ upsample the input signal, apply Simpson's rule on the upsampled signal, then $2\times$ downsample the integrated signal. An alternative implementation is to multiply the spectrum of the signal by the transfer function of the FIR filter in the original domain corresponding to the RRT rule.

3 Boundary treatments

FFT will be used to perform various operations of signal processing, which requires a discrete signal of a finite length satisfying a periodic boundary condition. It is especially convenient to have a signal starting and ending with 0 signal values. Otherwise, for a one-dimensional signal $\{s_n\}_{n=0}^N$ with either s_0 or s_N being nonzero, it is advantageous to extract an analytical component such as $u(t) = s_0 + (s_N - s_0)t/N$ or $u(t) = s_0 + (s_N - s_0)\sin^2(\pi t/2N)$, $t \in [0, N]$ out of the sequence $\{s_n\}_{n=0}^N$, so to obtain a modified signal $\{v_n \stackrel{\text{def}}{=} s_n - u(n)\}_{n=0}^N$ satisfying a periodic and zero boundary condition, while the integral of $u(t)$ can be computed analytically and added back to the numerical integral of $\{v_n\}_{n=0}^N$. The same technique of extracting out an analytical component can be generalized to multi-dimensional images to render a periodic and zero boundary condition.

An alternative technique to enforce a periodic boundary condition concatenates a given sequence $\{s_n\}_{n=0}^N$ with a mirror-reflected (also called time-reversed when the signal is one-dimensional and time-indexed) copy [10] and FFT-processes a length-doubled sequence $\{u_n\}_{n=0}^{2N+1}$ with $u_n \stackrel{\text{def}}{=} s_n$ for $n \in [0, N]$ and $u_n \stackrel{\text{def}}{=} s_{2N-n}$ for $n \in [N+1, 2N]$. From an engineering perspective, enforcing a periodic boundary condition helps to mitigate spurious boundary effects [10] due to abrupt truncation of a continuous signal varying smoothly in the original domain. However, previous methods of boundary treatment are only able to secure continuity for signal values, leaving spacetime derivatives in jeopardy of discontinuity. To overcome this limitation, a new method of boundary treatment firstly extracts out an analytical component and gets a signal $\{s_n\}_{n=0}^N$ with the periodic and zero boundary condition, then concatenates $\{s_n\}_{n=0}^N$ with a time-reversed and sign-flipped copy of itself to yield $\{u_n\}_{n=0}^{2N+1}$ with $u_n \stackrel{\text{def}}{=} s_n$ for $n \in [0, N]$ and $u_n \stackrel{\text{def}}{=} -s_{2N-n}$ for $n \in [N+1, 2N]$, which ensures periodicity and continuity in both signal values and time derivatives.

Some analysis helps to see the improvement. A periodic sequence $\{s_n\}_{n=0}^N$ can faithfully sample and represent all linear combinations of half-integer harmonics of the form $\cos(kt)$ or $\sin(kt)$, $k \in [0, N] \subseteq \mathbb{N}$, $t \in [0, \pi]$, among which, some harmonics satisfy the zero boundary condition individually and are called boundary vanishing (BV), while others do not vanish at the boundary and are called boundary non-vanishing (BNV) accordingly. Still, the BNV harmonics necessarily sum to zero at the end points to honor the overall zero boundary condition. By concatenating $\{s_n\}_{n=0}^N$ with a time-reversed and sign-flipped copy of itself, it is clear that the BV harmonics and their time derivatives are continuous at the boundaries, while the BNV harmonics have both their sum and individual derivatives vanish at the boundaries. Thus, the new method of boundary treatment guarantees continuous signal values and first-order time derivatives. Again, the method generalizes to multi-dimensional signals or images straightforwardly.

In many practical applications, an original signal of interest often extends in the original domain

well beyond the finite data volume capacity of a digital processor. Then it is necessary to apply a periodic sequence repeating a window function with a finite support to clip chunks of windowed data out of the original signal, with each chunk fitting into the digital processor. So long as the sequence of window functions sums to a constant and the signal processing executes a linear operation (as numerical integration is), summing up processed results from the windowed data recovers the desired result corresponding to the original signal of interest [8, 9]. In particular, if the window function is continuous and reduces to 0 at the boundary of its support in the original domain, then each chunk of windowed data automatically satisfies a periodic and zero boundary condition, thus is ready for FFT-based processing. As explained above, it is further advantageous to concatenate such a chunk of windowed data with a time-reversed copy of itself, to ensure continuity for both the signal values and the first-order derivatives at the boundaries, which helps to further reduce spurious numerical artifacts due to signal truncation at the boundaries.

4 Split-band integration

Fig. 3 shows a block diagram of an exemplary implementation of our split-band integrator, where a general input image/signal firstly undergoes a step of preparation for FFT-based processing, including, for example, clipping out a chunk of windowed data, extraction of an analytical component, and concatenating the data with a mirror-reflected copy of itself, *etc.* Then, the prepared data is transformed to the Fourier domain via FFT, where the spectrum is split into an LF band and an HF band which go through two separated paths of integration called the LF and the HF paths respectively. The LF and HF bands are delimited by a frequency boundary at a fraction $\eta \in (0, 1)$ of the Nyquist frequency $f_N > 0$, with the LF band containing all of the frequencies within the range $(-\eta f_N, \eta f_N)$, and the HF band collecting the frequencies in the set $(-f_N, -\eta f_N] \cup [\eta f_N, f_N)$. A suitable frequency boundary has an $\eta \in [1/32, 1/4]$, which is small enough such that the LF path of integration achieves a good accuracy with a low-complexity low-order NCF, but not too small to cause a wide dynamic range for the $(j2\pi f)^{-1}$ transfer function applied in the HF path.

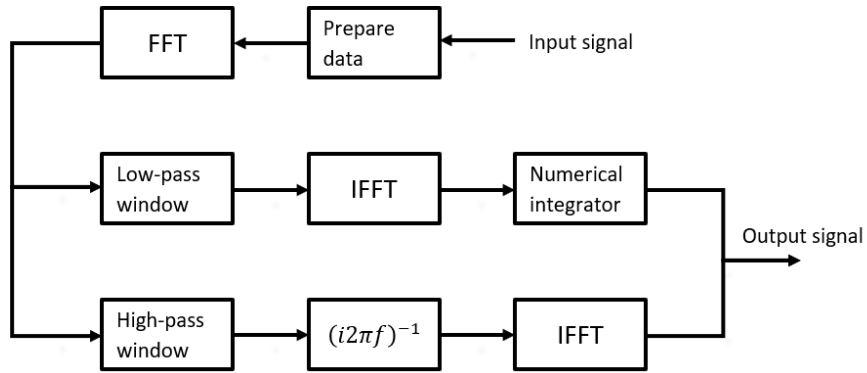


Fig. 3: Block diagram of a split-band integrator.

In one exemplary embodiment of the HF path, a high-pass window zeros out the Fourier amplitudes for lower frequencies within the interval $(-\eta f_N, \eta f_N)$ and passes the higher frequency components, then the passed Fourier amplitudes is multiplied by the transfer function $(j2\pi f)^{-1}$, finally the signal is transformed back to the original domain by inverse FFT (IFFT) and becomes an HF output. Theoretically, the combination of FFT/IFFT and the $(j2\pi f)^{-1}$ Fourier filter implements an integrator exactly without any approximation. Practically, the numerical computation is subject

to rounding errors due to a finite arithmetic precision, but steers clear of a numerical instability and an excessively wide signal dynamic range by choosing an η that is not too small.

An exemplary embodiment of the LF path uses a low-pass window to pass the Fourier amplitudes in the LF band $(-\eta f_N, \eta f_N)$, zero out the HF components, and double the number of Fourier samples by padding zero amplitudes for frequencies above f_N . It then converts the signal back to the original domain via IFFT, obtaining a $2\times$ upsampled signal having values interpolated at the half-grid points. Next, the upsampled signal is integrated into an integral image by a classical NCF such as Simpson’s rule, which is $2\times$ downsampled by outputting only half of the integral data at the original grid points before upsampling. Equivalently, using the RRT integrator, the rectangle rule is employed to integrate a subsequence consisting of the half-grid points only, while the trapezoid rule is used to process the other half of the sequence, then the results from the rectangle and the trapezoid rules are weighted by $2/3$ and $1/3$ respectively and combined into an LF output. Being able to integrate polynomials up to the third order exactly, the relative error of Simpson’s rule or the RRT integrator scales as $O(\eta^4)$, which decreases rapidly as η becomes smaller.

Furthermore, it is straightforward to employ a higher-order NCF to achieve a better accuracy that scales as a higher-order power of η . Either way, a classical NCF or a combination of NCFs corresponds to an FIR filter that makes a fairly accurate integrator for the LF band. An alternative implementation of the LF path does not sum a data sequence in the original domain. Instead, it multiplies the signal spectrum after the low-pass window by a transfer function corresponding the FIR filter, then applies an IFFT to produce an LF output. In this alternative, the LF and HF paths can have their signals combined in the Fourier domain immediately before IFFT, thus share one and the same step of IFFT.

Naturally, the last step of an SBI is to combine the LF output and the HF output into a single output signal, which has both the low and the high frequency components of the input signal integrated with a satisfying accuracy.

In terms of computational complexity, it is apparent that an SBI in accordance with the exact block diagram of Fig. 3, having the LF integration done in the original domain, would cost 2 or 3 times of memory and approximately $2 \sim 3$ times of runtime, comparing to a previous FFT-based integrator that applies a $(j2\pi f)^{-1}$ transfer function across the entire Fourier band of the input signal. Such increase of computational complexity is paid in exchange of numerical stability and operability on hardware with a limited arithmetic precision. An alternatively implemented SBI can certainly reduce the memory and runtime costs, when the LF path also does integration in the Fourier domain and shares an IFFT with the HF path. However, doing integration in the original domain affords an opportunity to use the Kahan summation algorithm and similar variants [14, 15] to achieve enhanced numerical accuracy under a limited hardware precision.

Compared to a classical NCF integrator working over the whole frequency band $(-f_N, f_N)$, an SBI apparently involves a much larger count of arithmetic operations in theory due to the FFT/IFFT steps. However, when algorithms are executed on a hardware device in practice, the latency of data access and transfer has a large sway over the actual runtime, so much so that the runtime of an SBI is similar to that of a classical NCF with only a modest overhead factor. More importantly, as being analyzed previously, the classical NCFs cannot produce an accurate integral for the HF components close to the Nyquist frequency. In order to achieve an acceptable accuracy over the entire bandwidth of an original signal, an NCF-based classical integrator would have to upsample a critically sampled image by a large factor $N_{\text{usf}} > 1$, and incur a steep $N_{\text{usf}}^d \times$ increase of both memory usage and runtime for a d -dimensional image, $d \in \mathbb{N}$. The great advantage of an SBI is to

avoid this cost and deliver a better accuracy.

5 Tests and validations

There is a very effective method to test and validate the accuracy of our SBI using an FFT-based numerical differentiator [6, 10], which is analytically exact and numerically stable. A bandwidth-limited image is sufficiently sampled into a golden image, which is transformed to the Fourier domain via FFT, where a $j2\pi f$ transfer function is multiplied along each Fourier dimension, then the signal is transformed back into the original domain and becomes a highly accurate differential image. An SBI under test takes the differential image as input and produces an integral image, which is compared with the golden image to compute a *RelativeError* in order to assess the numerical accuracy of the SBI, with said *RelativeError* being defined as

$$RelativeError \stackrel{\text{def}}{=} \frac{\|IntegralImage - GoldenImage\|_p}{\|GoldenImage\|_p}, \quad (6)$$

where *GoldenImage* and *IntegralImage* are numerical arrays representing the golden image and the integral image respectively, $\|\cdot\|_p$, $p \geq 1$ is a p -norm measuring the length of a numerical array considered as a vector. Commonly used norms include the Manhattan norm $\|\cdot\|_1$, the Euclidean norm $\|\cdot\|_2$, and the infinity norm $\|\cdot\|_\infty$.

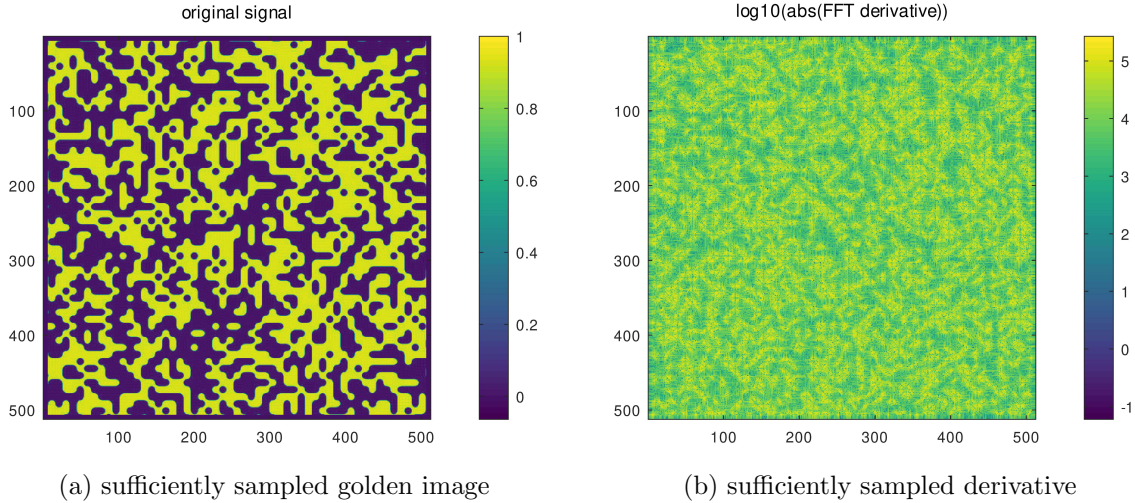


Fig. 4: A golden image and its derivative.

Numerical tests have been carried out using GNU Octave version 6.4.0 on a Dell PowerEdge T330 tower server running openSUSE 15.5. The machine has 8 Intel(R) Xeon(R) CPU cores clocked at 3.70GHz and sharing 16GB memory. To create a golden image for testing, an initially binary image I_0 of a randomly generated QR code is space- and bandwidth-limited by repeatedly applying a Landau-Widom space- and spectrum-limiting operator [16]

$$L \stackrel{\text{def}}{=} W_r \circ \mathcal{F}^{-1} \circ W_f \circ \mathcal{F} \circ W_r, \quad (7)$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the FFT and IFFT operators respectively, W_r and W_f represent windowing functions in the real space and the frequency space respectively, which zero out the border region of a spatial or spectral image while leaving its interior content intact. A relative image

change $\delta I_n \stackrel{\text{def}}{=} \|L^n I_0 - L^{n-1} I_0\|_\infty / \|L^n I_0\|_\infty$, $n \in \mathbb{N}$ due to the n -th application of the L operator is monitored as an indicator of convergence. Specifically, for the numerical tests, a 512×512 QR code was used as the initial I_0 , both W_r and W_f were set to zero out a border region of 6 pixels. It was observed that, upon 10 applications of the L operator, δI_{10} is already less than 10^{-6} , and after 15 iterations, δI_{25} falls below 10^{-14} to approach the level of double precision. The resulted golden image $L^{25} I_0$ is shown in Fig. 4(a), which sufficiently samples and reconstructs a two-dimensional continuous image $I(x, y)$, $(x, y) \in \mathbb{R}^2$. A sufficiently sampled image of the same size for the derivative $i(x, y) \stackrel{\text{def}}{=} \partial^2 I(x, y) / \partial x \partial y$ is computed via FFT/IFFT by applying a transfer function $(j2\pi f_x)(j2\pi f_y)$ in the Fourier domain, which is highly accurate up to the double-precision floating-point default of Octave. The result is displayed in Fig. 4(b).

An SBI is coded in Octave, which splits the spectrum with $\eta = 1/16$ along each of the two dimensions and has integration done in the Fourier domain for both the LF and the HF bands, where the transfer function for the LF path corresponds to an FIR filter for the RRT integrator. For comparison, a classical integrator using the trapezoid rule is also implemented in Octave by summing signals in the original domain. The same derivative image of Fig. 4(b) and its upsampled versions are fed into the SBI and the classical trapezoid integrator, then the resulted integral images are compared to the golden image of Fig. 4(a) and its correspondingly upsampled versions, with the *RelativeError* of equation (6) computed using the infinity norm $\|\cdot\|_\infty$.

When operating on the 512×512 sized derivative image of Fig. 4(b) without upsampling, the classical trapezoid integrator takes 2.19 seconds to compute an integral image that suffers a large *RelativeError* = 0.22 comparing to the original golden image, while the SBI takes 5.32 seconds to produce a highly accurate integral image with a *RelativeError* = 6.66×10^{-7} comparing to the same golden image. The memory usage of the SBI is exactly $2\times$ that of the classical trapezoid integrator as expected theoretically. The already small *RelativeError* of the SBI is mostly systematic and due to the approximation made by Simpson's rule that truncates terms of the fourth-order and above in a Taylor-expanded signal, since the FFT-based integrator and other signal processing steps entail no other approximation and incur only numerical rounding errors, which is at the level of 10^{-15} for the double-precision floating-point that GNU Octave defaults to.

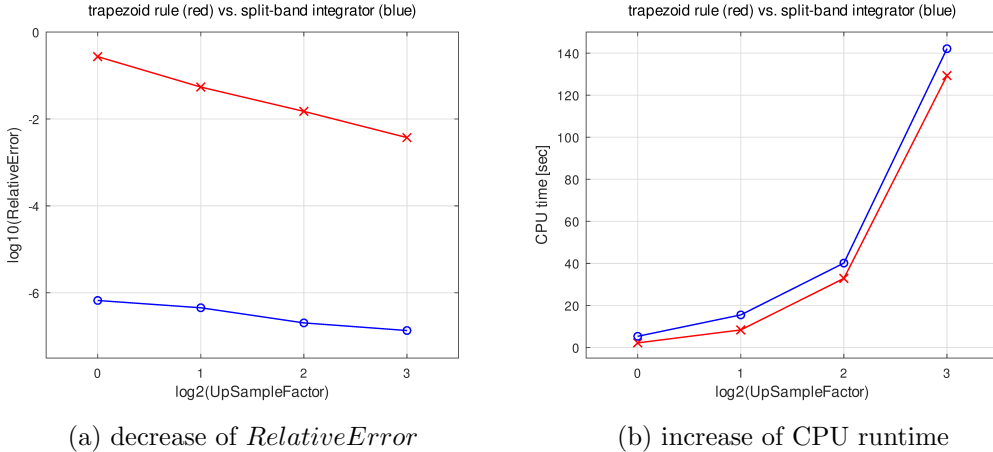


Fig. 5: The decrease of *RelativeError* values and increase of CPU runtimes for the SBI in blue and the classical trapezoid integrator in red respectively, corresponding to $N_{\text{usf}} = 1, 2, 4, 8$.

Therefore, the accuracy of an SBI using Simpson's rule or the RRT integrator for the LF path is

already satisfactory for many applications, and in practice the accuracy is more likely to be limited by the hardware precision on an embedded system or a neural network. By contrast, the accuracy of a classical integrator using an NCF over the entire bandwidth of a signal is far from satisfactory, when the signal is close to critically sampled. In order to reduce the approximation error associated with the NCF, said classical integrator would have to upsample the input image by an upsample factor $N_{\text{usf}} > 1$ and endure an N_{usf}^d times increase of computational costs in both memory usage and CPU time, with $d \in \mathbb{N}$ being the dimensionality of the image. The same GNU Octave scripts are used to test upsampled images with $N_{\text{usf}} = 2, 4, 8$ as inputs, in addition to the previous golden and derived images which correspond to an $N_{\text{usf}} = 1$.

As observed in the experiments and shown in Fig. 5(a), the *RelativeError* of the classical trapezoid integrator is reduced by a factor of $2^3 = 8$ for each $2\times$ increase of N_{usf} , in agreement with the trapezoid rule having an approximation error proportional to the cube of the grid size. However, as shown in Fig. 5(b), such an improvement of accuracy is obtained at a steep cost of $4\times$ increase in CPU runtime for upsampling a two-dimensional image. Even at the hefty cost of $64\times$ increase of memory and runtime at $N_{\text{usf}} = 8$, the classical trapezoid integrator still experiences a *RelativeError* of 3.72×10^{-3} , which is far too large even comparing to that of the SBI at $N_{\text{usf}} = 1$. Fig. 5(a) further shows that the *RelativeError* of the SBI also decreases monotonically as N_{usf} increases, indicating that the SBI has not hit a noise floor due to rounding errors. Naturally, as shown in Fig. 5(b), the CPU runtime of the SBI also increases steeply as N_{usf} increases, always within a factor ~ 2 times that of the classical trapezoid integrator. Fortunately, the SBI makes such expensive upsampling unnecessary, thanks to its split-band strategy.

6 Summary

A novel method of split-band integration has been reported for numerically accurate and stable integration of bandwidth-limited images, which combines the advantages of both the classical NCFs and the FFT-based integrator while avoiding their drawbacks at the same time. The method splits an input into two images and passes them through two separated paths of integration, with one image containing LF components going through an LF path and getting integrated by a classical NCF or a combination of NCFs to a predetermined accuracy in a numerically stable fashion, while the other image collects the complementary HF portion and gets filtered by $(j2\pi f)^{-1}$ along each dimension in the Fourier domain. The results from the two separated integrators are combined to output a desired integral image.

Numerical tests using GNU Octave have validated the functionality of an SBI, which, with Simpson's rule or an equivalent integrator and an LF/HF splitting frequency boundary at $1/16$ of the Nyquist frequency, consistently achieves a low *RelativeError* at the level of 10^{-7} , at the cost of $2\times$ in memory usage and $\sim 2\times$ in CPU runtime in comparison to a classical NCF integrator. The observed *RelativeError* is mostly systematic and substantially all attributable to the approximation of Simpson's rule by truncating terms of the fourth-order and above in a Taylor-expanded signal. This corroborates suitability of an SBI to operate on an embedded system or a neural network with a limited arithmetic precision, where it helps to overcome the problem of numerical instability and a wide dynamic range. Besides, it is worth noting that an SBI having the LF integration done in the original domain provides an advantage of using the Kahan summation algorithm and similar variants [14, 15] to achieve enhanced numerical accuracy for integration via the NCFs, which is even $\log_2 N$ times better than the methods using FFT or pairwise summation in a parallel fashion [15, 17, 18], where N is the length of the data sequence being integrated.

References

1. F. C. Crow, "Summed-area tables for texture mapping," *Proc. 11th Conf. Comp. Graph. Inter. Techn. (SIG-GRAPH 1984)*, vol. 18, no. 3, pp. 207-212 (1984).
2. J. P. Lewis, "Fast template matching," *Proc. Vision Interface 1995*, Quebec City, Canada, May 15-19, 1995, pp. 120-123. Also see the expanded and corrected version "Fast normalized cross-correlation," available online at <https://scribblethink.org/Work/nvisionInterface/nip.pdf>
3. S. C. Irick, R. K. Kaza, and W. R. McKinney, "Obtaining three-dimensional height profiles from a two-dimensional slope measuring instrument," *Rev. Sci. Instrum.*, vol. 66, no. 2, pp. 2108-2111 (1995).
4. S. Cai and M.-G. Jeong, "Methods of rasterizing mask layout and methods of fabricating photomask using the same," United States Patent No.: US 10255397B2, Date of Patent: April 09, 2019.
5. O. J. Woodman, "An introduction to inertial navigation," Technical Report UCAM-CL-TR-696, University of Cambridge, Computer Laboratory (August 2007).
6. A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*, 3rd Ed. (Prentice Hall, 2010).
7. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes*, 3rd Ed. (Cambridge University Press, 2007).
8. R. Brincker and A. Brandt, "FFT Integration of time series using an overlap-add technique," in T. Proulx (ed.), *Structural Dynamics*, vol. 3, Conf. Proc. Society for Experimental Mechanics Series (Springer, New York, 2011), pp. 1467-1474, available online at https://doi.org/10.1007/978-1-4419-9834-7_130
9. A. Brandt and R. Brincker, "Integrating time signals in frequency domain - Comparison with time domain integration," *Measurement*, vol. 58, pp. 511-519 (2014).
10. L. P. Yaroslavsky, *Advances in Sampling Theory and Techniques* (SPIE, 2020).
11. S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep learning with limited numerical precision," *Proc. 32nd Int. Conf. Machine Learning*, vol. 35, pp. 1737-1746 (2015).
12. R. Banner, I. Hubara, E. Hoffer, and D. Soudry, "Scalable methods for 8-bit training of neural networks," *Proc. 32nd Int. Conf. Neural Information Processing Systems*, pp. 5151-5159 (2018).
13. R. Pintelon, J. Schoukens, "Real-time integration and differentiation of analog-signals by means of digital filtering," *IEEE Trans. Instrum. Meas.*, vol. 39, no. 6, pp. 923-927 (1990).
14. W. Kahan, "Further remarks on reducing truncation errors," *Comm. ACM*, vol. 8, no. 1, p. 40 (1965).
15. N. J. Higham, "The accuracy of floating point summation," *SIAM J. Sci. Comput.*, vol. 14, no. 4, pp. 783-799 (1993).
16. H. J. Landau and H. Widom, "Eigenvalue distribution of time and frequency limiting," *J. Math. Anal. Appl.*, vol. 77, no. 2, pp. 469-481 (1980).
17. B. Bilgic, B. K. P. Horn, and I. Masaki, "Efficient integral image computation on the GPU," *2010 IEEE Intellig. Veh. Symp.*, UC San Diego, June 21-24, 2010, pp. 528-533.
18. D. Puchala and K. Stokfiszewski, "Numerical accuracy of integral images computation algorithms," *IET Image Proc.*, vol. 12, no. 1, pp. 31-41 (2018).