PTEP

Prog. Theor. Exp. Phys. **2017**, 093F01 (15 pages)
DOI: 10.1093/ptep/ptx122

Deep searches for broadband extended gravitational-wave emission bursts by heterogeneous computing

Maurice H. P. M. van Putten*

Yeongsil-Gwan, Room 614, Physics and Astronomy, Sejong University, Seoul South Korea
*E-mail: mvp@sejong.ac.kr

Received June 13, 2017; Revised July 31, 2017; Accepted August 4, 2017; Published September 15, 2017

We present a heterogeneous search algorithm for broadband extended gravitational-wave emission, expected from gamma-ray bursts and energetic core-collapse supernovae. It searches the $(f,\dot{f})$-plane for long-duration bursts by inner engines slowly exhausting their energy reservoir by matched filtering on a *graphics processor unit* (GPU) over a template bank of millions of 1 s duration chirps. Parseval's theorem is used to predict the standard deviation σ of the filter output, taking advantage of the near-Gaussian noise in the LIGO S6 data over 350–2000 Hz. Tails exceeding a multiple of σ are communicated back to a *central processing unit*. This algorithm attains about 65% efficiency overall, normalized to the fast Fourier transform. At about one million correlations per second over data segments of 16 s duration ($N = 2^{16}$ samples), better than real-time analysis is achieved on a cluster of about a dozen GPUs. We demonstrate its application to the capture of high-frequency hardware LIGO injections. This algorithm serves as a starting point for deep all-sky searches in both archive data and real-time analysis in current observational runs.

Subject Index F30, F31

1. Introduction

Gravitational radiation offers a potentially powerful new channel to discover the physical nature and population statistics of core-collapse supernovae and their association with neutron stars and black holes. Recently, LIGO identified a black hole binary progenitor of GW150914 [1] with remarkably low spin. Stellar-mass black holes are believed to be remnants of extreme transient events such as gamma-ray bursts and core-collapse supernovae. Shortly after birth in the latter, possibly including the superluminous variety, black holes may encounter strong interactions with high-density matter. This outlook opens a window to release their angular momentum in gravitational radiation, leaving slowly spinning remnants with $a/M \simeq 0.3$ in dimensionless spin [2]. Future detection of similar events may reveal whether GW150915 is typical or the tail of a broad distribution in black hole mass and spin.

Neutron stars and black holes born in core collapse of massive stars are of great interest as candidate sources of gravitational waves, especially for their potential to also be visible in the electromagnetic spectrum. Searches for these events may be triggered in either radiation channel [3,4] and a combined detection would enable identification of the source and host environments, in the footsteps of multi-wavelength observations of gamma-ray bursts (GRBs) pioneered by BeppoSAX [5]. Electromagnetic (EM) triggers obtained from transient surveys allow off-line gravitational wave (GW) analysis of LIGO–Virgo archive data. On the other hand, GW triggers require relatively low latency in EM follow-up, which may be challenging because of the modest localization of LIGO–Virgo detections.

While core-collapse supernovae are relatively numerous, only a small fraction is known to be associated with extreme events. For instance, the true event rate (corrected for beaming) of long GRB is about 1 per year within a distance of 100 Mpc. Achieving sensitivity to tens of Mpc to emissions limited to $E_{\rm GW} = O\left(1M_\odot c^2\right)$, where c denotes the velocity of light, poses a challenge for *deep searches* in gravitational wave data.

Broadband extended gravitational-wave emission (BEGE) from the aforementioned extreme events may be produced with durations lasting up to tens of seconds. In chirp-based spectrograms, such events may appear as trajectories marked by frequencies slowly wandering in time, featuring ascending and descending chirps [6]. To search for these signatures in the $(f,\dot{f})$-plane, we recently devised a dedicated butterfly filtering using chirp templates of intermediate duration τ of, e.g., one second, targeting a time scale of phase coherence that may capture tens to hundreds of wave periods associated with nonaxisymmetric accretion flows. Using millions of chirp templates, it can detect complex signals such as Kolmogorov scaling in noisy time-series; recently, in BeppoSAX, gamma-ray light curves with an average photon have counted down to 1.26 photons per 0.5 ms bin [7]. This kind of sensitivity suggests exploration of its further applications to strain-amplitude gravitational wave data [4].

Deep searches covering a complete science run of LIGO require considerable computing resources in the application of butterfly filtering with a dense bank of templates. We here report on a novel algorithm by heterogeneous computing comprising both *graphics* and *central processing units* (GPUs and CPUs, respectively) using the *Open Compute Language* (OpenCL) [8,9].

A primary challenge in heterogeneous computing is circumventing GPU–CPU bottlenecks arising from potentially vast discrepancies in data throughput over the *peripheral component interface* (PCI). Our algorithm exploits near-Gaussian noise in the high-frequency bandwidth of 350–2000 Hz in the LIGO data, whereby the output of matched filtering is essentially Gaussian as well. Near-optimal efficiency is obtained by retaining only tails of relatively high signal-to-noise ratios back to the CPU from the GPU output, whose cut-off is predicted by Parseval's theorem. Including overhead in the latter, our algorithm achieves about 65% efficiency normalized to a GPU-accelerated fast Fourier transform (FFT) in *complex-to-complex* (C2C), *single precision* (SP), and *interleaved out-of-place* memory allocation.

Our choice of chirp templates is guided by inner engines involving black holes described by the Kerr metric, interacting with high-density matter, expected in core collapse of massive stars and mergers involving neutron stars, the latter envisioned in association with short GRBs with extended emission (SGRBEE) and long GRBs with no supernovae (LGRBN) such as GRB060614 [10].

In the application to LIGO S6, we give a detailed description of our database, which comprises bandpass filtering to the aforementioned 350–2000 Hz (over 64 s segments of data, $N = 2^{18}$ samples) and a restriction to simultaneous H1–L1 detector output (29.4% of total S6 data). On a modern GPU, we realize approximately 80 000 correlations per second over 16 s segments of data ($N = 2^{16}$ samples). On a cluster of about a dozen GPUs, about 1 million correlations per second realizes better than real time analysis. As such, the presented method is applicable to both archive analysis and low-latency searches in essentially real time, pioneered following GW150914 [11] and for current Advanced LIGO runs (see, e.g., Ref. [12]). For the present archive analysis of LIGO S6, however, our focus is on deep searches for an exhaustive search, all-sky and blind without triggers from electromagnetic data.

Existing comprehensive, blind searches for bursts [13–20] cover various broadband emissions over 16–500 Hz [21], 40–1000 Hz [22], and, for short bursts, 32–4096 Hz [23]. Around a rotating

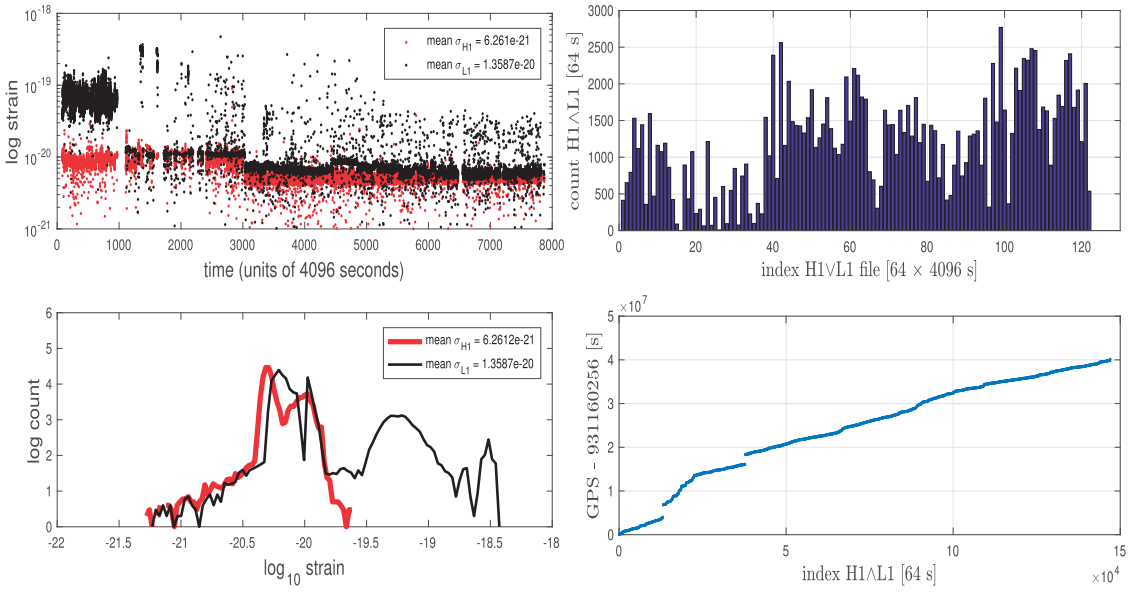

Fig. 1. Overview of LIGO S6, showing standard deviations over 64 s data segments ($N = 2^{18}$ samples) of H1∧L1. H1∧L1 ranged from 0–68% with an average yield of 29.4% of H1∨L1. Strain noise H1 and L1 was better than 10^{-20} over 89% and 64%, respectively, of the time. The performance of H1 and L1 became somewhat more consistent after the first 3000 hours.

black hole of mass M newly formed in core-collapse supernovae, the gravitational wave emission from nonaxisymmetric mass flow around the innermost stable circular orbit (ISCO) is expected to be potentially luminous [24], featuring a broadband descending chirp [4,25] with late-time frequency [26]:

$$f_{\rm GW} \simeq (595\text{–}704)\ {\rm Hz}\left(\frac{10M_\odot}{M}\right), \tag{1}$$

where the range in the frequency refers to dependence on initial black hole spin. This motivates our present focus on the high-frequency bandwidth 350–2000 Hz in LIGO S6. In this frequency bandwidth, the LIGO noise is essentially Gaussian, which shall be exploited in our GPU–CPU method of analysis.

In Sect. 2 we review chirp-based spectrograms by butterfly filtering. In Sect. 3, our heterogeneous computing algorithm is described with the use of *pre- and post-callback* functions. Benchmark results are given in Sect. 4. Section 5 reports on a detection of some illustrative LIGO S6 hardware burst and calibration injections. We summarize our findings and outlook in Sect. 6.

2. Bandpass-filtered H1 and L1 data in S6

LIGO S6 covers the period 7 July 2009 through 20 October 2010. In our analysis of LIGO S6, we focus on epochs when H1 and L1 are both taking data. These H1∧L1 data represent 29.4% of data when either H1 or L1 were taking data (H1∨L1), measured over 64 second segments (Fig. 1).

In our search for gravitational wave emission from core-collapse supernovae associated with stellar-mass black holes, we focus on a frequency bandwidth of 350–2000 Hz. With bandpass filtering (over 64 s segments of data, $N = 2^{18}$ samples), LIGO noise is essentially Gaussian (see, e.g., Ref. [4]). This bandwidth may contain gravitational wave emission from nonaxisymmetric mass motion about the innermost stable circular orbit (ISCO) around stellar-mass black holes [4,26].

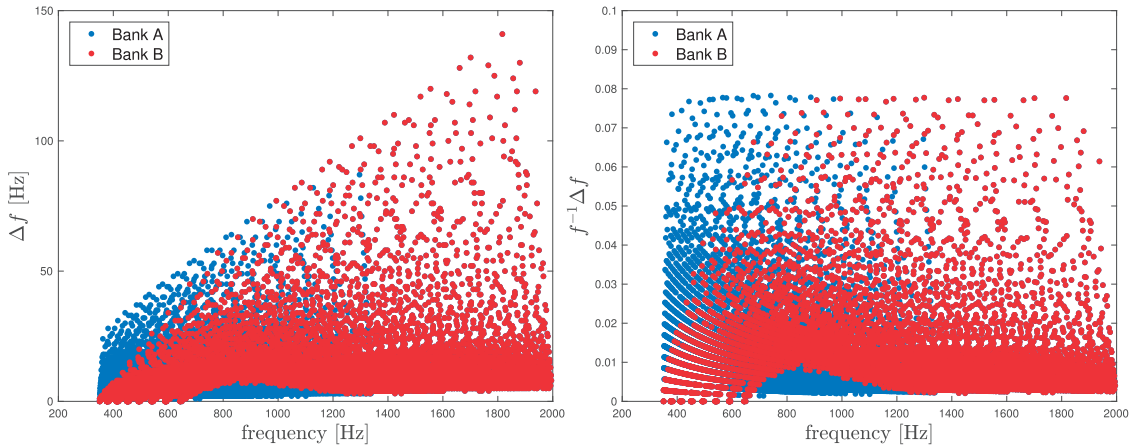

Fig. 2. Overview of template banks of 1 s duration chirps (dots) covering 350–2000 Hz, shown by frequency f and their change Δf in frequency, illustrated with a bank of small size. The chirps used in butterfly filtering are symmetric in time, obtained by superposition of chirps forward and backward in time, suitable in searches for both ascending and descending chirps. Banks A and B are similar, except Bank A is larger by including more pronounced chirps (larger Δf) at the lower bound of 350 Hz.

3. Butterfly filtering by heterogeneous computing

To search for slowly evolving trajectories in time–frequency space, we consider matched filtering over a large bank of chirp templates covering a range in f and time rate-of-change of frequency $\dot{f}$, i.e., a butterfly

$$0 < \delta_1 < \left|\dot{f}\right| < \delta_2 \tag{2}$$

for some $\delta_{1,2} > 0$. Over a finite bandwidth of frequencies, the resulting output is a *chirp-based spectrogram*. The chirps are generated from a long-duration template, produced by solving a pair of ordinary differential equations modeling black hole spindown against high-density matter at the ISCO [7], the results of which are illustrated in Fig. 2.

Matched filtering of a time series $y(t)$ against chirp templates $w(t)$ is defined by the correlations

$$\rho(t) = \int_{-\infty}^{\infty} w(s)y(t+s)dt. \tag{3}$$

In the present application to LIGO strain data, $y(t)$ and $w(t)$ have zero mean. This integral is conveniently evaluated in the Fourier domain as $\tilde{\rho}(k) = \tilde{w}^*(k)\tilde{y}(k)$, where

$$\tilde{f}(k) = \frac{1}{2\pi}\int_{-\infty}^{\infty} f(t)e^{-ikt}dt,\ f(t) = \int_{-\infty}^{\infty}\tilde{f}(k)e^{ikt}df. \tag{4}$$

Discretizing Eq. (3) to samples at equidistant instances t_n ($n = 0, 1, \ldots, N$), we evaluate Eq. (4) by FFT. This is more efficient compared to direct evaluation of Eq. (3) in the time domain, whenever the number of samples N exceeds a few hundred. This may be readily observed by comparing compute times, convolving two vectors **u** and **v** by FFT versus direct evaluation in, e.g., MatLab; see also Ref. [27].

Table 1. Overview of the database of H1∧L1 when both H1 and L1 were taking data (measured over 64 s data segments), extracted from a total of 12 726 LIGO S6 frames. Frames on the LIGO Open Science Center (LOSC) comprise 4096 s ($N = 2^{24}$ samples) of H1 or L1 data, here bandpass-filtered to 350–2000 Hz over 64 s data segments ($N = 2^{18}$ samples). The H1∧L1 data for analysis are in 36 files of 4096 × 64 s segments (Table 2).

Data	64 s segments	LOSC frames (4096 s)	Memory	Source, target
H1	422 912	6608	–	LOSC
L1	391 552	6118	–	LOSC
H1∨L1	499 712	7867	1.05 TB	Disk
H1∧L1	147 000	–	305 GB	Disk
File	4096	–	8.59 GB	Compute node

For reference, recall that correlating vectors **y** and **w** comprises three steps: twice forward FFT, pointwise products $\tilde{\rho}$ involving complex conjugation, and one inverse FFT:

$$\{\tilde{\mathbf{w}}, \tilde{\mathbf{y}}\} = \mathrm{FFT}\{\mathbf{w}, \mathbf{y}\},\ \tilde{\rho} = \tilde{\mathbf{w}}^* \cdot \tilde{\mathbf{y}},\ \rho = \mathrm{FFT}^{-1}\{\tilde{\rho}\}. \tag{5}$$

For LIGO S6, the (downsampled) sampling rate is 4096 s^{-1}, whence $N = 2^{16}$ for 16 s data segments.

With vanishing mean values, the standard deviation of $\rho(t_n)$,

$$\sigma = \frac{1}{\sqrt{N}} \sqrt{\sum_{n=0}^{N-1} \rho(t_n)^2}, \tag{6}$$

satisfies Parseval's theorem

$$\sigma = \frac{\sqrt{2}}{N} \sqrt{\sum_{n=1}^{N/2-1} |c_n|^2}, \tag{7}$$

where c_n denote the Fourier coefficients of $\rho(t)$ according to the FFT pair

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} f_n e^{-ikt_n},\ f_n = \sum_{k=0}^{N-1} c_k e^{ikt_n}. \tag{8}$$

Bandpass-filtered to 350–2000 Hz, H1∧L1 (Table 1) has noise that is essentially Gaussian (see, e.g., Ref. [4]). This property is inherited by $\rho(t)$ in Eq. (3). Hence, $\rho(t)$ is effectively described by σ in Eq. (7) for a given pair of data segment and template. Therefore, Eq. (7) provides a *predictive step* to the output of Eq. (5). In processing Eq. (5) on a GPU, a threshold in a *post-callback function* can be used to retain only tails (Fig. 3),

$$\rho(t_n) > \kappa\sigma, \tag{9}$$

for feedback to the CPU over the PCI. In Eq. (9), we implicitly apply the inequality to the absolute value of $\rho_n = \rho(t_n)$. Thus, Eq. (9) circumvents vast discrepancies in the throughput of GPUs and CPUs whenever κ is on the order of a few. This step is essential for an optimal heterogeneous computing algorithm, to be benchmarked further below.

It should be mentioned that, below 350 Hz, LIGO data is non-Gaussian, giving rise to distributions of $\rho(t)$ that occasionally show multiple peaks. (This depends on the pair of data segment and template.) In this event, σ inadequately describes $\rho(t_n)$, whereby tails defined by Eq. (9) become less meaningful in defining candidate detections.

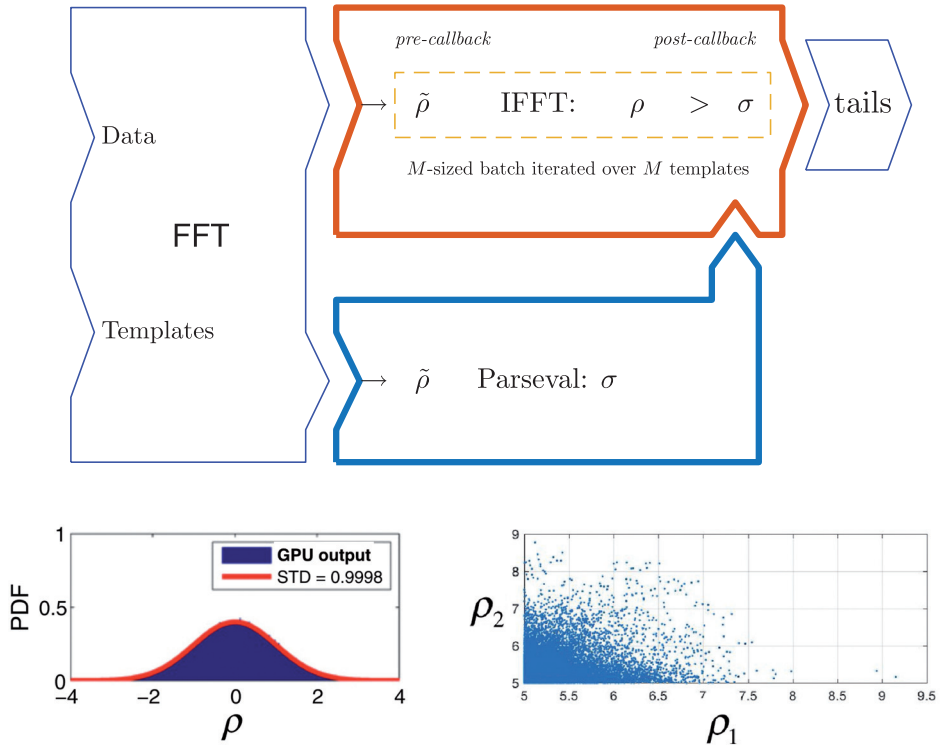

Fig. 3. Butterfly filtering by heterogeneous computing applied to M H1∧L1 data 16 s segments ($N = 2^{16}$ samples) by CPU (thin lines) and GPU (thick lines). Parseval's theorem computes M^2 standard deviations σ of essentially Gaussian correlations $\rho(t)$ obtained by matched filtering on the GPU. Potentially relevant results are contained in the tails of $\rho(t)$. Retaining tails $\rho(t) > \kappa\sigma$ for a threshold κ realizes near-optimal heterogeneous GPU–CPU computing, effectively circumventing PCI bandwidth limitations when κ is small.

Processing is applied to batches of $M = 2048$ of H1∧L1 16 s data. Such a *block* of about 9 hours of data comprises about 1 GB, suitable for allocation in the *global memory* of a typical GPU. Chirp templates are extracted by time slicing from a model of black hole spindown [7]. While these emissions are of relatively high frequency when the black hole spins rapidly, late time emission following spindown reaches an asymptotic frequency satisfying Eq. (1). Analysis is performed in groups of M such templates by FFT in *batch mode*. Batch mode operation is essential to reaching optimal FFT performance on a GPU.

3.1. Teraflop compute requirements

Sensitivity to arbitrary, slowly varying transients is realized by banks sufficiently large to densely cover the $(f, df/dt)$ parameter space. For matched filtering, a bank of chirps of 1 s duration covering $f = O(N)$ Hz with frequency changes $O(f)$ will be dense with step sizes on the order of $1/N$ Hz in f and df/dt, setting a minimum bank size of order $O(N^2)$. For f on the order of one kHz, the minimum bank size is $O(1\text{M})$, needed to ensure a reasonable probability to match a signal (a "hit" when $\rho > \kappa\sigma$).

For a better than real-time analysis by butterfly filtering of data segments of duration T over a template bank of size $K_1 M$, the required compute performance is

$$\dot{n} = 5N \log_2 N \times K_1 M T^{-1} = 2.75\,\text{teraflops}, \tag{10}$$

Table 2. Partitioning files of the H1∧L1 database on a heterogeneous compute node into blocks allocated in the global memory on a GPU for processing by $\mathrm{FFT}_{N,M}$ with transforms of size $N = 2^{16}$ in batch mode of size $M = 2048$.

Unit	Array length	Memory size	Target
File	8 blocks	8.59 GB	Disk storage, host
Block	NM	1.1 GB	Global memory/GPU
FFT batch size	$M = 2048$	1.1 GB	FFT/GPU
FFT data segment	$N = 2^{16}$	0.5 MB	Global and local memory/GPU

where the right-hand side refers to our choice of $T = 16$ s and a template bank of $\alpha = 1, 2, \ldots, K_1$ sets of size $M = 2048$ each.

The hardware requirements are considerably higher, since FFTs tend to be *memory limited* (not compute limited) on GPUs, especially when FFT array sizes exceed the size of *local memory* privy to individual *compute units* (CU). At typical efficiencies of $\eta \simeq 7\%$ in these cases, Eq. (10) points to a minimum requirement of about 50 teraflops at GPU maximal compute performance, assuming that Eq. (10) is realized at approximately optimal efficiency normalized to FFT.

In what follows, we consider partitioning the template bank by $K_1 M$ and the data in $\beta = 1, 2, \ldots, K_2$ blocks, respectively, in

$$W_\alpha = \{w_{\alpha k}\}_{k=1}^{M}, \quad Y_\beta = \{y_{\beta k}\}_{k=1}^{M}. \tag{11}$$

In our application, $K_1 M = 2^{23}$ (up to 8 million) $K_2 M = 288$ for LIGO S6. The total number of correlations for a full LIGO S6 analysis is

$$K_1 K_2 M^2 = 5 \times 10^{12}. \tag{12}$$

For our choice of 16 s segments ($N = 2^{16}$), Eq. (12) defines a compute requirement of 2.5×10^{19} floating-point operations for a complete LIGO S6 analysis over a bank of 8M templates.

3.2. Batch mode with pre- and post-callback functions

Figure 3 shows the butterfly filtering by our GPU–CPU heterogeneous computing algorithm, based on detailed partitioning of data and work listed in Table 2. For Eq. (5), we choose FFT with C2C, SP, and with interleaved out-of-place memory allocation by 1D $\mathrm{FFT}_{N,M}$ of length $N = 2^{16}$ in batch mode of size $M = 2048$:

(i) $\mathrm{FFT}_{N,M}$ of M pairs of 16 s data segments of H1∧L1 comprising a block of MN complex single precision (CSP) in allocatable memory of size 1 GB. (FFT is applied to arrays of complex numbers, merging pairs of real H1 and L1 data.) Transforms $\tilde{Z} = (\tilde{H}_1, \tilde{L}_1)$ comprise M subarrays $\tilde{\mathbf{Z}}_k$ $(k = 1, 2, \ldots, M)$, each of length N;

(ii) A chirp template $\mathbf{w}$ of duration $\tau = 1$ s is extended by zeros to length N and its transform $\tilde{\mathbf{w}}$ is loaded into the global memory. A pre-callback function computes M transforms $\tilde{\rho}$ from M pointwise array multiplications $\tilde{\rho}_{(k)} = \tilde{\mathbf{Z}}_{(k)} \cdot \tilde{\mathbf{w}}^*$ $(k = 1, 2, \ldots, M)$;

(iii) Inverse $\mathrm{FFT}_{N,M}$ applied to $\tilde{\mathbf{Z}}_{(k)}$ produce M corrections $\rho_k(t_n)$ over N samples, representing the most computationally (but memory limited) intensive step on the GPU;

(iv) (ii) and (iii) are repeated M times, once for each of the M chirp templates $\mathbf{w}$ at a total computational effort of M^2 inverse-FFT_N.

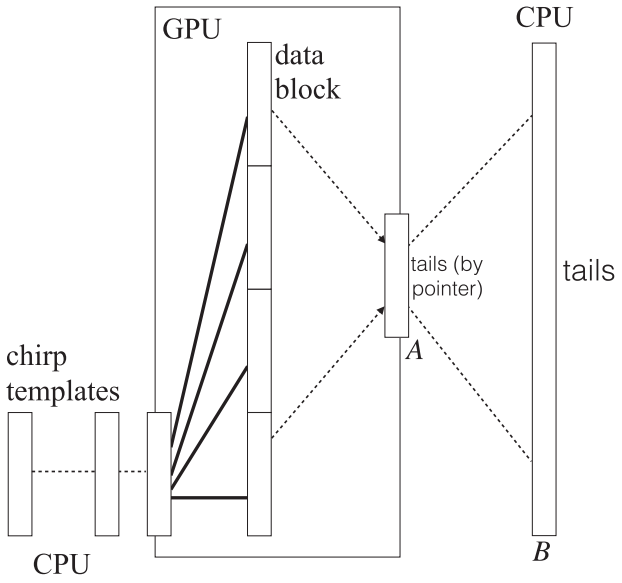

Fig. 4. Tails of correlations between chirps with a block of NM H1$\wedge$L1 (thick lines) projected by pointer into an array A of size N in the global memory, that will be sparse whenever κ is on the order of a few. Gathered over the PCI, $*A$ are stored in an array B of size NM on the host. B is stored to disk after completing correlations with a complete bank of chirp templates.

At $5N\log_2 N$ flops per FFT_N, these combined steps for the N and M^2 mentioned above involve $\sim$20 teraflop, producing 2 TB output. The latter shows the need to retain only tails of $M\times M$ convolutions $\tilde{\rho}_{(k)}$ $(k=1,2,\ldots,M)$, i.e., candidate events exceeding a multiple of σ_{km}, one for each 16 s segment k of data and chirp template l $(k,m=1,2,\ldots,M)$.

The σ_{km} are pre-computed by Parseval's theorem (7). As norms of complex Fourier coefficients, Eq. (7) is computationally demanding, requiring off-loading to the GPU as well (Fig. 3). For $\kappa=5.5$, for instance, tails are limited on the order of 10^4 byte s^{-1}, well below the PCI bandwidth of several GB s^{-1}, allowing near-optimal computing at about 65% efficiency overall (including Parseval's step), normalized to FFT alone. Retaining tails over the PCI by the CPU is realized as follows.

3.3. Gathering GPU tails over the PCI

The tails of correlations satisfying Eq. (9) are gathered in two steps (Figs. 3 and 4). In correlations of a template $\mathbf{w}_k\epsilon W_\alpha$ and a segment $\mathbf{y}_m\epsilon Y_\beta$ $(1\le\alpha\le K_1,\ 1\le\beta\le K_2,\ k,m=1,2,\ldots,M)$, Eq. (9) is obtained for each σ_{km}. Thus, $\mathbf{w}_k$ gives M tails in correlation with Y_β referenced by time

$$t_m=t_{n_m+mN} \tag{13}$$

of maximal correlation satisfying

$$\rho_m\ge\kappa\sigma_{km}, \tag{14}$$

where $\rho_m=\rho(t_m)$. To circumvent limited PCI bandwidth, Eqs. (13 and 14) are converted to pointers projected into an array $A_{\alpha k}$ of size N,

$$A_{\alpha k}=\{(t_m,\rho_m)|\,\rho_m\ge\rho_{m'}\ge\kappa\sigma_{km'}\ (\text{all } m')\}. \tag{15}$$

We evaluate Eq. (15) by a post-callback function on the GPU by updating (t_m,ρ_m) with $(t_{m'},\rho_{m'})$ whenever $\rho_{m'}>\rho_m$ and $\rho_{m'}>\sigma_{km'}$. As an asynchronous read/write by pointwise index on the global

memory, this may lead to indeterministic behavior when two processors operate concurrently on the same index. When κ is appreciable, $A_{\alpha k}$ is sparse, and this anomalous behavior is exceedingly rare.

Repeating Eq. (15) for all $\mathbf{w}_k \epsilon W_\alpha$ obtains M^2 tails by pointers

$$A_\alpha = \bigcup_{k=1}^{M} A_k. \tag{16}$$

Collecting all pointers in A_α is evaluated by the CPU.

Gathering results over the complete template bank is obtained by repeating Eq. (16) for all $1 \leq \alpha \leq K_1$, each time dereferencing A_α into an array B of block size NM on the host,

$$B = \bigcup_{\alpha=1}^{K_1} * A_\alpha, \tag{17}$$

evaluated by the CPU. In collecting B, we select data with maximal ρ values at t_{n+mN} from the $*A_\alpha$.

Gathering *all* hits by removing a selection of maximal ρ in collecting B in Eq. (17) produces extended output with up to two orders of magnitude more output in the case of a signal. For the burst injection discussed below (Fig. 7), for instance, this increases the output to tens of GB for a bank of 8M templates. Such extended output may be of interest to second runs, following up on selected data segments covering candidate events, but less so to first runs through all data such as LIGO S6.

4. Benchmarks under OpenCL and filter output

The algorithm shown in Figs. 3 and 4 is implemented in Fortran90 and C++ using AMD's clFFT (in C99) under OpenCL. Following Table 2, clFFT operates on blocks of filtered H1$\wedge$L1 data in 1 GB blocks allocatable in global memory for $\text{clFFT}_{N,M}$ (C2C, SP) with interleaved out-of-place memory storage.

Under OpenCL, a GPU is partitioned in CUs with fast but privy local memory and registers. Only global memory is shared across all CUs. Performance hereby critically depends on efficient use of local memory and minimal use of global memory, since access to the latter is relatively slow. With a local memory size of typically 32 kB, clFFT performance for C2C SP will be essentially maximal $N \leq 2^{12}$. In our application, $N = 2^{16}$, whereby clFFT performance is practically memory limited.

Figure 5 shows clFFT performance on GPUs with varying numbers of CUs (each comprising a number of *stream processors*) and the global memory bus bandwidth (GB s^{-1}), namely the R9 nano (4096, 64, 512), the R9 390 (2560, 40, 384), and the D700 (2048, 32, 264). For the first, performance is over 600 Gflop s^{-1} for $N > 2^{12}$ (about 1000 Gflop s^{-1} for $N \leq 2^{12}$). This is a direct result of the 32 kB local memory size and $8N$ bytes in complex single precision storage and the need to access the global memory when $N > 2^{12}$. For $N = 2^{16}$, the net result is an overall efficiency of about 7% of peak floating-point compute performance by stream processors alone.

We implement Parseval's theorem by partial sums off-loaded to the GPU, the results of which are summed by the CPU. At a few hundred Gflop s^{-1} performance thus achieved, the wall clock compute time is about 25% compared to that of clFFT on the GPU. Including overhead in steps (i)–(iv) of Sect. 3.2 and gathering tails (Sect. 3.3), the net result (including Parseval's step) is an overall efficiency of about 65%, normalized to clFFT alone as shown in Fig. 4, or about $\sim 8 \times 10^4$ correlations per second per GPU. On a cluster of about a dozen GPUs, we hereby realize about 1 million correlations per second, sufficient for a real-time analysis by up to 16 million templates according to Eq. (10).

The filter output stored to disk is listed by block in files Bn, $n = 1, 2, \ldots, 288$, illustrated in Table 3.

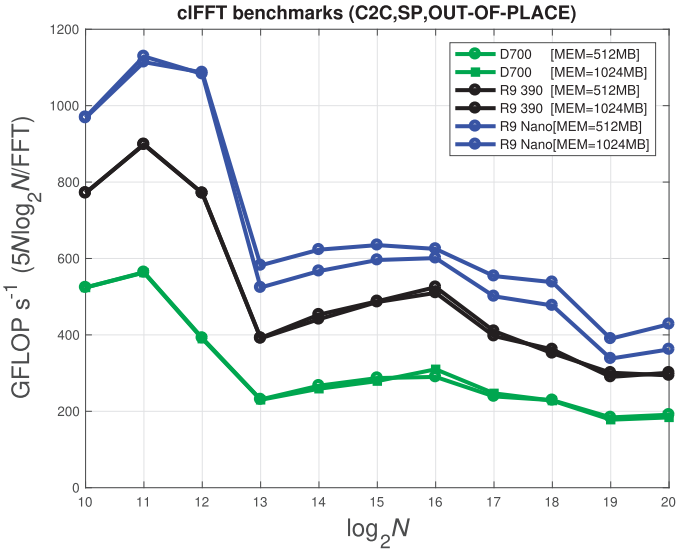

Fig. 5. Performance of $\mathrm{FFT}_{N,M}$ by clFFT under OpenCL on the AMD D700, R9 390, and R9 Nano GPUs, expressed in GFLOP s^{-1} as a function of transform array size N in C2C SP with interleaved out-of-place data storage and no output back to the CPU. Results are shown for two different batch sizes with correspondingly different allocations in the global memory. These results define a practical limit on performance in FFT-based correlations that involve additional communications to a CPU over a PCI.

Table 3. The butterfly filtering output Bn of a block n ($n = 1, 2, \ldots, 288$) of hits $\rho_i > \kappa\sigma$ ($i = 1, 2$) lists the data sample offset $i \epsilon \{1, 2, \ldots, 2^{27}\}$, ρ_i, and f_i, the latter the initial frequency of the associated chirp template. Multiplication of ρ_i by 1000 allows storage of all entries in 4 byte integers. The sample shown of B161 (6 388 647 rows produced by a bank of 4M templates) highlights some simultaneous hits. Zeros represent no hit.

Sample offset i	$1000 \times \rho_i$(H1)	$1000 \times \rho_i$(L1)	f_i(H1) [Hz]	f_i(L1) [Hz]
...				
17 712 959	0	5522	0	1988
17 713 193	5747	0	486	0
17 713 194	5516	0	623	0
17 713 195	6424	0	632	0
17 713 196	6578	6660	497	489
17 713 197	5769	7491	488	489
17 713 198	7315	6671	490	489
17 713 199	8530	7111	563	565
...				

5. Tails and LIGO burst injections

To illustrate a full analysis, Fig. 6 shows a pseudo-spectrum of the tails (9) of H1$\wedge$L1 LIGO S6, obtained by averaging results of blocks using a template bank of intermediate size of 0.5 M chirps. The detailed structure shown represents the *non*-Gaussian features that carry any potentially relevant information, visible only by zooming in on tails in an otherwise overall near-Gaussian PDF of the internal GPU output ρ (Fig. 3). This has been verified numerically, in obtaining completely smooth spectra of tails of ρ following time-randomization of H1 or L1 data (Fig. 6).

Figure 6 shows various pronounced features, some of which are probably associated with unsteady behavior in various instrumental lines familiar from conventional Fourier spectra of S6 strain noise [31]. The details of these remain to be understood in more detail, especially so given the

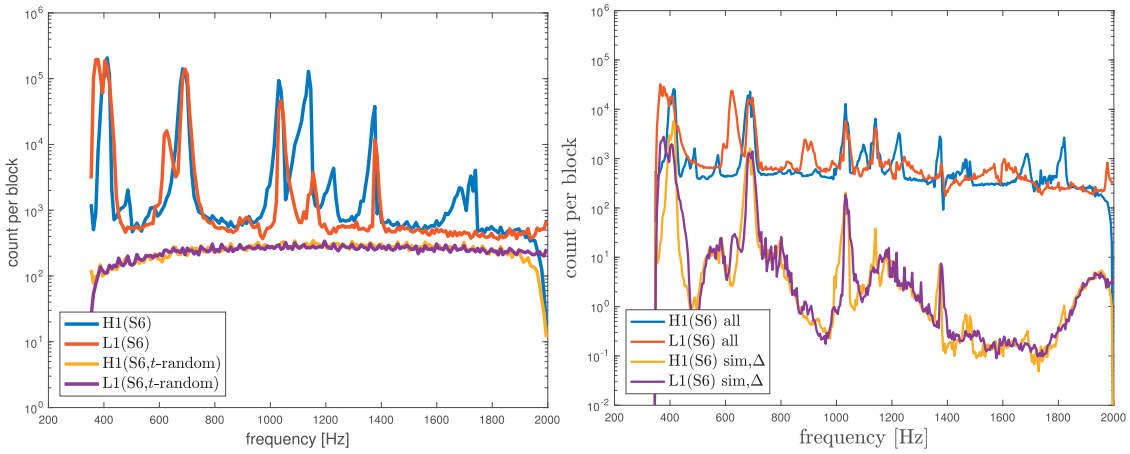

Fig. 6. (Left) Pseudo-spectra of simultaneous hits in tails $> \kappa\sigma$ with $\kappa = 5.5$ of butterfly-filtered output of H1 (red) and L1 (blue), shown as an average over four blocks (161–2, 177–8) of S6 H1∧L1, using a bank of type A of 8M chirp templates, along with baseline results following time-randomized data. (Right) Pseudo-spectra as an average over *all* 288 blocks of S6 H1∧L1, using a bank of type A of 0.5M chirp templates, of H1 and L1 by independent counts and by simultaneous counts with frequency pairs within $\Delta = 50$ Hz.

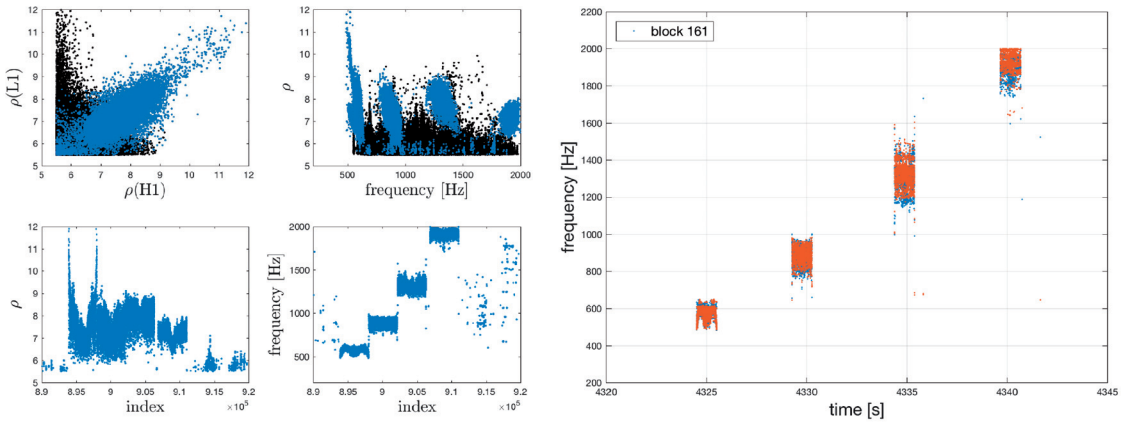

Fig. 7. Detection of a high-frequency LIGO injection in block 161 by butterfly filtering with a bank of type B of 4M chirp templates at large injection SNR (Table 4), seen in simultaneous hits in H1 and L1 (left panels; ρ and frequency refer to geometric means of those of H1 and L1). Hits in H1 (red) and L1 (blue) practically overlap (right panel), shown as a function of time based on the data sample offset in output file B161 (Table 3).

nontrivial residual spectrum of simultaneous hits with frequency pairs (f_1, f_2) of H1 and L1 that are relatively close, here shown with $|f_1 - f_2| < \Delta$, $\Delta = 50$ Hz. (A similar spectrum is obtained for $\Delta = 100$ Hz.) For the analysis with a bank of 0.5M templates shown, the total counts per block for H1 and L1 are $(2.1 \times 10^6, 3.2 \times 10^6)$ with simultaneous counts (19.73%, 13.01%), reduced to (6.82%, 4.49%) for frequency pairs within $\Delta = 50$ Hz ((7.55%, 4.98%) for frequency pairs within $\Delta = 100$ Hz).

LIGO detectors are routinely given a variety of hardware injections to test the detectors and various signal detection pipelines. Of interest to the present analysis are burst injections that cover the relatively high-frequency range 350–2000 Hz. The following uses some LIGO injections for a formal test and validation of software implementation.

Figure 7 shows an injection to both H1 and L1 captured by our algorithm at large injection SNR (LOSC), detected using a bank of 4M templates in a partial analysis of LIGO S6.

Table 4. Sample of a LIGO S6 injection in H1 and L1 [28,29], comprising a sequence of sine-Gaussian signals [30] stepwise covering 50–2000 Hz with injection signal-to-noise ratio (SNR) listed by the LOSC.

GPS time [s]	strain amplitude (h_{rss})	Waveform (f [Hz], Q)	SNR (LOSC)
H1			
958 413 408.20	3.57×10^{-21}	sine-Gaussian (393,9)	93.27
958 413 413.50	4.33×10^{-21}	sine-Gaussian (554,9)	92.37
958 413 418.30	6.41×10^{-21}	sine-Gaussian (850,9)	98.23
958 413 423.40	9.84×10^{-21}	sine-Gaussian (1304,9)	91.20
958 413 428.70	7.47×10^{-21}	sine-Gaussian (2000,9)	23.15
L1			
958 413 408.20	3.63×10^{-21}	sine-Gaussian (393,9)	65.37
958 413 413.50	4.74×10^{-21}	sine-Gaussian (554,9)	68.50
958 413 418.30	7.19×10^{-21}	sine-Gaussian (850,9)	72.22
958 413 423.40	1.09×10^{-20}	sine-Gaussian (1304,9)	67.98
958 413 428.70	8.61×10^{-21}	sine-Gaussian (2000,9)	25.85

For high-confidence detections, correlated H1–L1 output such as illustrated in Fig. 7 is essential. While signal injections are often injected at the same GPS time, astrophysical sources will impact H1 and L1 along some finite viewing angle. In the time domain, this is commonly identified by maximizing correlations over some finite time shift, here 0–10 ms given the distance between H1 and L1. Here, we make use of the fact that a difference in arrival time between H1 and L1 from a putative astrophysical source with finite time rate-of-change in $f(t)$ is equivalent to a frequency shift, allowing searches in simultaneous H1–L1 filter output such as plotted in Fig. 7.

Figures 8 and 9 shows a validation of sensitivity (see also the earlier analyses of Refs. [4,7]), here a priori limited to ρ exceeding $5.5\,\sigma$ by choice of κ in Eq. (9), obtained in a partial LIGO S6 analysis using 8M templates. Overall, it appears that sensitivity in H1 is slightly better than L1 when signals are small. Searches for signals fainter than those shown would require a rerun of the analysis with $\kappa < 5.5$ in Eq. (9). For such extremely deep searches, excess tail sizes can conceivably be curtailed by generalizing Eq. (9) to a finite band, $\kappa_1\sigma > \rho(t_n) > \kappa_2\sigma$ with $\kappa_2 - \kappa_1 \lesssim 1$.

Figure 9 quantifies the gain in using bank sizes beyond the minimal requirements (Sect. 3.1), showing an increase in hit counts and ρ in a detection of a sample of high-frequency burst injections.

6. Conclusions and outlook

Probing inner engines to gamma-ray bursts and core-collapse supernovae requires deep searches in LIGO data. Taking full advantage of modern GPU hardware, we present a GPU–CPU implementation of butterfly filtering to search for broadband extended emission in gravitational waves from accreting flows around black holes, potentially relevant to the most extreme transient events.

Our benchmarks demonstrate near-optimal performance using banks of up to millions of chirp templates at better than real-time analysis, facilitating deep searches in LIGO archive data such as S6, advanced LIGO O1, and the currently ongoing O2 run.

Specific applications of the proposed method include correlation analysis of the H1 and L1 detectors and identification of mysterious or peculiar events of interest to further analysis. A leading-order indication of correlations may be derived, for instance, from counting statistics of hits, comparing simultaneous hit counts with total hit counts in H1 and L1. Specific events of interest may be followed up by second runs, gathering all hits by removing a selection of maxima in collecting B in Eq. (17).

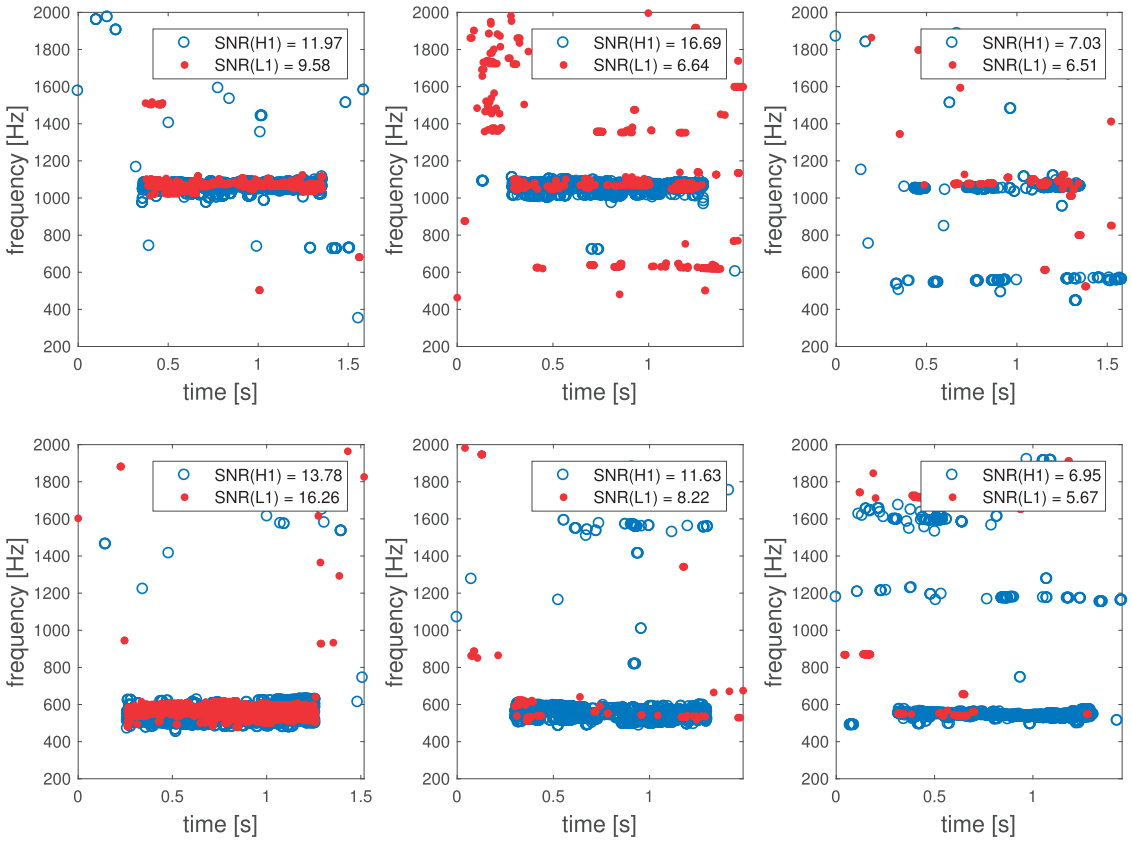

Fig. 8. Selected high-frequency injections at 1053 Hz (top panels) and 554 Hz (bottom panels) H1∧L1 LIGO S6 at high to low injection SNR (LOSC) (left to right), here detected by H1 (blue circles) and L1 (red dots) using a bank of type A of 8M chirp templates. The 1053 Hz (554 Hz) injections shown are at the respective GPS times 932 380 188.50, 935 143 367.60, 946 193 522.10 (959 322 411.40, 934 962 011.00, 946 205 393.30).

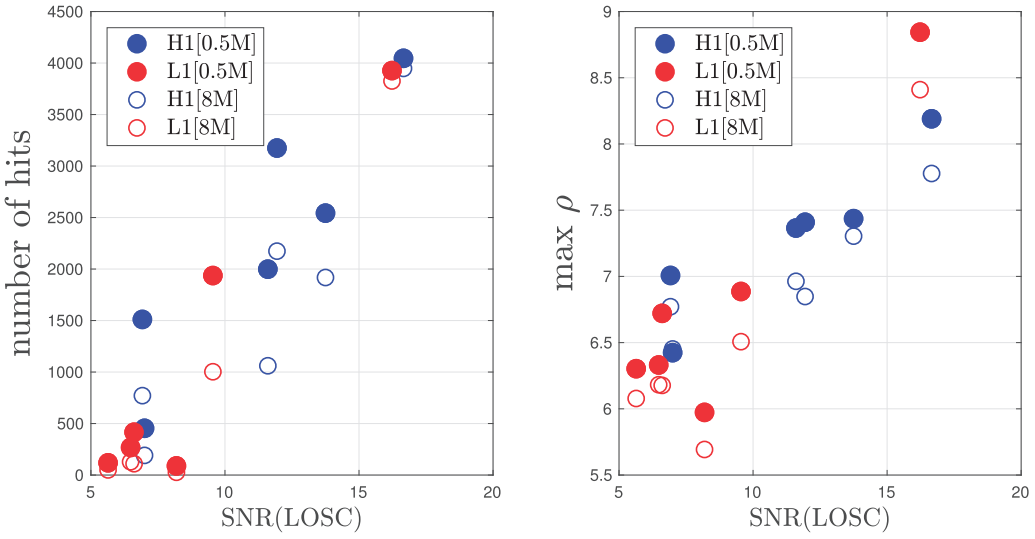

Fig. 9. The number of hits $\rho(t) > \kappa\sigma$ and maximal values of ρ in Fig. 8 about injections at 554 Hz and 1053 Hz in H1 and L1 shows a generic trend with SNR in the injection process. On average, counts improve by a factor of 1.69 and ρ increments by 0.29 with the template bank of 8M compared to 0.5M chirps. Hits are counted with a frequency margin of ± 50 Hz about these injection frequencies.

In butterfly filtering, signal detection typically comprises a large number of hits, representing approximate matches with no single template providing a perfect match to the full signal at hand, as illustrated in Fig. 8. This combined output can in principle recover essentially maximal sensitivity [4]. For automated searches of candidate events, clustering algorithms might apply (see, e.g., Ref. [32]), which may also facilitate quantifying the level of confidence for such complex detection output.

Acknowledgements

The author gratefully acknowledges detailed constructive comments from the referee and J. B. Kanner. This work was partially supported by the National Research Foundation of Korea under grants 2015R1D1A1A01059793 and 2016R1A5A1013277 and made use of LIGO S6 data from the LIGO Open Science Center (losc.ligo.org), provided by the LIGO Laboratory and LIGO Scientific Collaboration. LIGO is funded by the US National Science Foundation. Additional support is acknowledged from MEXT, the JSPS Leading-edge Research Infrastructure Program, a JSPS Grant-in-Aid for Specially Promoted Research 26000005, a MEXT Grant-in-Aid for Scientific Research on Innovative Areas 24103005, the JSPS Core-to-Core Program, A. Advanced Research Networks, and the joint research program of the Institute for Cosmic Ray Research.

References

[1] B. P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), Phys. Rev. Lett. **116**, 241102 (2016).
[2] M. H. P. M. van Putten and M. Della Valle, Mon. Not. R. Astron. Soc. **464**, 3219 (2017).
[3] B. S. Sathyaprakash and B. F. Shutz, Living Rev. Relat. **12**, 2 (2009).
[4] M. H. P. M. van Putten, Astrophys. J. **819**, 169 (2016).
[5] F. Frontera et al., Astrophys. J. Suppl. **180**, 192 (2009).
[6] A. Levinson, M. H. P. M. van Putten, and G. Pick, Astrophys. J. **812**, 124 (2015).
[7] M. H. P. M. van Putten, C. Guidorzi, and F. Frontera, Astrophys. J. **786**, 146 (2014).
[8] B. Gaster, D. R. Kaeli, L. Howes, P. Mistry, and D. Schaa, *Heterogeneous Computing with OpenCL* (Elsevier, Amsterdam, 2011).
[9] Khronos Group, 2015, *OpenclCL 2.1 and SPIR-V 1.0 Launch*, https://www.khronos.org/assets/uploads/developers/library/overview/opencl_overview.pdf.
[10] M. H. P. M. van Putten, G. M. Lee, M. Della Valle, L. Amati, and A. Levinson, Mon. Not. R. Astron. Soc. **444**, L58 (2014).
[11] B. Abbot et al., Astrophys. J. Lett. **826**, 13A (2016).
[12] X. Guo, Q. Chu, S. K. Chung, and L. Wen, submitted (2017).
[13] B. P. Abbott et al., Phys. Rev. D **69**, 102001 (2004).
[14] B. P. Abbott et al., Classical Quantum Gravity **24**, 5343 (2007).
[15] B. P. Abbott et al., Phys. Rev. D **80**, 102002 (2009).
[16] B. P. Abbott et al., Phys. Rev. D **80**, 102001 (2009).
[17] J. Abadie et al., Phys. Rev. D **81**, 102001 (2010).
[18] J. Abadie et al., Phys. Rev. D **85**, 122007 (2012).
[19] M. Ando et al., Classical Quantum Gravity **22**, S1283 (2005).
[20] J. Aasi et al., Phys. Rev. D **88**, 122004 (2013).
[21] B. P. Abbott et al., Astrophys. J. **841**, 89 (2017).
[22] B. Abbot et al., Phys. Rev. D **93**, 042005 (2016).
[23] B. P. Abbott et al., Phys. Rev. D **95**, 042003 (2017).
[24] M. H. P. M. van Putten, Phys. Rev. Lett. **87**, 091101 (2001).
[25] M. H. P. M. van Putten, Astrophys. J. **684**, L91 (2008).
[26] M. H. P. M. van Putten et al., Phys. Rev. D **83**, 044046 (2011).
[27] J. O. Smith, 2016, *Review of the Discrete Fourier Transform* (DFT), https://ccrma.stanford.edu/ jos/ReviewFourier/ReviewFourier.pdf.
[28] LIGO Open Science Center, S6 Burst Injections, https://losc.ligo.org/s/injections/s6/burst/H1_s6burst_simple.txt, date last accessed August 2017.

[29] LIGO Open Science Center, S6 Burst Injections, https://losc.ligo.org/s/injections/s6/burst/L1_s6burst_simple.txt, date last accessed August 2017.
[30] É. C. Mottin, M. Miele, S. Mohapatra, and L. Cadonati, Classical Quantum Gravity **27**, 194017 (2010).
[31] LIGO Open Science Center, S6 Instrumental Lines, http://losc.ligo.org/s6speclines, date last accessed August 2017.
[32] D. George, H. Shen, and E. A. Huerta, arXiv:1706.07446 [gr-qc] [Search INSPIRE].