

Multi-scale Peak and Trough Detection Optimised for Periodic and Quasi- Periodic Neuroscience Data

Steven M Bishop^{1*} Ari Ercole¹

Correspondence Address:

University of Cambridge Division of Anaesthesia,
Box 93, Cambridge University Hospitals NHS Foundation Trust,
Hills Road,
Cambridge.
CB2 0QQ.
UK.

Email: sbishop@doctors.org.uk

Telephone: +44 (0) 1223 217889

Keywords: Peak detection; trough detection; algorithm design; optimisation;
neuroinformatics; intracranial pressure waveform analysis

¹ Division of Anaesthesia, University of Cambridge, Cambridge, UK

* Corresponding author

Abstract

Objectives

The reliable detection of peaks and troughs in physiological signals is essential to many investigative techniques in medicine and computational biology. Analysis of the intracranial pressure (ICP) waveform is a particular challenge due to multi-scale features, a changing morphology over time and signal-to-noise limitations. Here we present an efficient peak and trough detection algorithm that extends the scalogram approach of *Scholkmann et al*, and results in greatly improved algorithm runtime performance.

Materials and Methods

Our improved algorithm (Modified-Scholkmann) was developed and analysed in Matlab R2015b (*MathWorks Inc., Massachusetts, USA*). Synthesised waveforms (periodic, quasi-periodic and chirp sinusoids) were degraded with white Gaussian noise to achieve signal-to-noise ratios down to 5dB and were used to compare the performance of the Original-Scholkmann and Modified-Scholkmann algorithms.

Results

Modified-Scholkmann has false positive (0%) and false negative (0%) detection rates identical to Original-Scholkmann when applied to our test suite. Actual compute time for a 200-run Monte Carlo simulation over a multicomponent noisy test signal was 40.96 ± 0.020 seconds (mean \pm 95%CI) for Original-Scholkmann and 1.81 ± 0.003 seconds (mean \pm 95%CI) for Modified-Scholkmann, demonstrating the expected improvement in run time complexity from $\mathbb{O}(n^2)$ to $\mathbb{O}(n)$.

Conclusions

The accurate interpretation of waveform data to identify peaks and troughs is crucial in signal parameterization, feature extraction and waveform identification tasks. Modification of a standard scalogram technique has produced a robust algorithm with linear computational complexity that is particularly suited to the challenges presented by large, noisy physiological datasets. The algorithm is optimised through a single parameter and can identify sub-waveform features with minimal additional overhead, and is easily adapted to run in real-time on commodity hardware.

Introduction

The reliable detection of peaks and troughs in physiological signals is essential to investigative techniques in medicine and computational biology and a prerequisite to many signal processing tasks. The challenge of accurate peak detection [1–3] is not unique to physiological signals and many solutions have been proposed in the literature ranging from simple window-thresholding [4] and wavelet transform techniques [5] to Hidden Markov Models [6], k-means clustering [7] and entropy-based techniques [8].

In neuroscience data the analysis of the intracranial pressure (ICP) waveform is a particular challenge and a number of algorithms [9,10] have been proposed. In particular, algorithms suitable for ICP peak detection must be suited to multi-scale features, changing waveform morphology with time and poor signal-to-noise.

There is a recognized trade-off between the generalizability of peak detection algorithms (degrees of freedom), accurate peak detection (false positive and false negative peak detection rates) and computational runtime performance [11]. For the most general algorithms to achieve a good domain-specific peak detection rate they usually require significant parameter optimisation and long computational runtimes. The converse is also generally true, with algorithms designed exclusively for a domain-specific problem producing a better peak detection performance.

Scholkmann et al [11] introduce an efficient and elegant algorithm for the automatic detection of peaks in noisy periodic and quasi-periodic signals, driven by their requirement to find peaks in near-infrared spectroscopy data. The Scholkmann algorithm (Original-Scholkmann) does not require any parameters, is fairly robust against high and low frequency noise and

accurately detects peaks in quasi-periodic signals (provided that the highest frequency of oscillation is less than or equal to four times the frequency of the lowest in the signal). The algorithm is well-suited to ICP waveform peak detection tasks, however it is hampered by an inefficient computational runtime and sizeable memory requirement.

In this work we adapt the algorithm of Scholkmann et al (Modified-Scholkmann) to dramatically improve runtime performance and memory storage requirements. We compare the peak detection rate and runtime performance of Modified-Scholkmann against Original-Scholkmann and provide further practical suggestions to improve performance when analysing ICP waveform data.

Materials and Methods

The Scholkmann Algorithm (Original-Scholkmann)

The Original-Scholkmann algorithm begins by calculating a local maxima scalogram (LMS) over a linearly detrended signal X of length N , where $X = \{x_t \mid 1 \leq t \leq N\}$. The LMS is a matrix of $S_{max} = \left\lceil \frac{N}{2} \right\rceil - 1$ scales (rows) against N columns. If the value at time $t \in \{1 \dots N\}$ and scale $s \in \{1 \dots S_{max}\}$ is locally maximal the matrix contains 0, otherwise it contains $r + \alpha$ (where $r \in \mathfrak{R}$ is a uniformly-distributed random variable and $\alpha \in \mathfrak{R}$ is a constant).

The LMC can be visualised more readily as an $S_{max} \times N$ matrix that marks the location of maxima at each scale (level of zoom). At scale $s = 1$ the matrix will encode a local maxima at time t if the signal value x_t is greater than the signal values at adjacent positions, i.e. $x_t > x_{t-1}$ and $x_t > x_{t+1}$. Likewise at scale $s = 2$ the matrix will encode a local maxima at

time t if the magnitude of the signal at position, x_t is greater than the signals at times $t - 2$ and $t + 2$. This pattern continues up to scale S_{max} .

The LMS extends from the first scale ($s = 1$, highest or “finest” resolution) to scale $s = \left\lceil \frac{N}{2} \right\rceil - 1$ (a “low resolution” of approximately half the signal length). However, it is subsequently cropped to include only scales from 1 to $S_{cropped}$, where $S_{cropped}$ is the scale containing the greatest number of maxima. In the final step of the algorithm the column-wise standard deviation is calculated across scales, and time points with a standard deviation of zero identify the locations of maxima (peaks).

Optimizations necessary to produce Modified-Scholkmann

A number of observations are necessary to optimize Original-Scholkmann. Firstly, the problem of peak and trough finding are equivalent: Troughs are found by inverting the original signal and applying the peak-finding algorithm. Hence, trough-finding can occur simultaneously at minimal additional computational cost. Secondly, calculation of the LMS is costly; computational runtime and memory requirements have an $\mathcal{O}(n^2)$ upper complexity bound. Thirdly, under most circumstances calculation of the LMS using scales up to $S_{max} = \left\lceil \frac{N}{2} \right\rceil - 1$ is unnecessary. An appropriate upper scale bound can be parameterised in the algorithm and chosen using domain-specific knowledge. For ICP waveform data the empirical maximum scale is equivalent to around one quarter signal wavelength, dramatically reducing the LMS search space.

Fourthly, calculation of uniform random numbers to populate the LMS is computationally expensive – the random numbers are only used during the final stage of the algorithm in the calculation of column-wise standard deviations. At the location of peaks the corresponding

column of the LMS will be a zero vector and can be found through linear search, rendering the calculation of column-wise standard deviations and pseudo-random numbers unnecessary. Finally, the LMS should only be calculated once per signal and cached to allow subsequent runs of the algorithm to complete in $\mathcal{O}(n)$ time. This is extremely useful when working with ICP waveforms since recursive application of the algorithm can be employed to identify the ICP waveform sub-peaks P_1 to P_3 in linear time.

Comparing Original-Scholkmann to Modified-Scholkmann

The Modified-Scholkmann algorithm was developed and analysed in Matlab R2015b (*MathWorks Inc., Massachusetts, USA*) on a sixteen core 3.3GHz Intel Xeon PC with 32GB RAM running Ubuntu Linux v12.04LTS. The algorithm code is found in Appendix 1.

Using a technique similar to *Scholkmann et al* [11], synthesised waveforms (periodic, quasi-periodic and chirp sinusoids) were degraded with white Gaussian noise to achieve a range of test waveforms with signal-to-noise ratios as low as 5dB. The synthesised waveforms were used to determine algorithmic performance and false positive and false negative peak detection rates were compared. The Multicomponent Simulated Noisy Signal defined by *Scholkmann et al* [11] was used in a 200-run Monte Carlo simulation to quantify the mean compute time with 95% confidence intervals for both algorithms. Further verification was performed using high-resolution electrocardiogram, arterial blood pressure and intracranial pressure waveforms from a local neurointensive care waveform database.

Results

The Modified-Scholkmann algorithm applied to the test suite has false positive (0%) and false negative (0%) detection rates that are comparable to Original-Scholkmann (*Figures 1 - 2*) provided a suitable maximum scale parameter is chosen (see *Discussion* for details).

Actual compute time for a 200-run Monte Carlo simulation using the Multicomponent Noisy Test Signal was 40.96 ± 0.020 seconds (mean \pm 95% CI) for Original-Scholkmann and 1.81 ± 0.003 seconds (mean \pm 95% CI) for Modified-Scholkmann, showing the expected improvement in runtime complexity.

Discussion

In the test suites a substantial improvement in compute time is seen from 40.96s for Original-Scholkmann to 1.81s for Modified-Scholkmann. The Modified-Scholkmann algorithm introduces a single parameter S_{max} , the maximum scale at which the LMS is computed, rather than deriving S_{max} from the signal data length. The LMS calculation in Original-Scholkmann is the single most expensive computation and efficiency gains in the LMS calculation can lead to substantial gains in runtime performance. The improved performance in Modified-Scholkmann relies on the periodic or quasi-periodic nature of physiological waveform signals – due to the periodicity of the signal it is unnecessary to search the scalogram for peaks at a scale greater than one cycle length, allowing the limitation of S_{max} to a range much less than the signal length N . In this work an empirical range for S_{max} for ICP waveform data was found to be equivalent to one quarter the waveform period. This reduces the algorithm's maxima search space, reducing space and time complexity from $\mathcal{O}(n^2)$ in Original-Scholkmann to $\mathcal{O}(kn) \equiv \mathcal{O}(n)$ in Modified-Scholkmann. An improvement in efficiency of this magnitude is necessary to support real-time or near real-time analysis of neurological waveform data.

A widening edge artefact is seen in the LMS at increasing scales because there is insufficient data to identify the location of 'local' maxima. This is a limitation of both algorithms (Original-Scholkmann and Modified-Scholkmann); they both usually fail to identify the first and last peak or trough in the signal. The use of larger signal lengths and overlapping multiple windows can substantially mitigate this effect.

The original and modified algorithms are resistant to noise and operate at SNRs as low as 5dB. At low SNRs both algorithms identify the expected peaks but with reduced location accuracy. This results from the tendency of both algorithms to identify the most prominent local maxima in the expected sub-region of interest (which is a necessary constraint for quasi-periodic signals), allowing the algorithm to be misled by occasional high-magnitude noise.

The calculation of LMS is deterministic and results in a unique matrix for each signal that is amenable to memory or disk caching for later reuse. This acts as a simple method to improve runtime performance. Peak finding over a previously analysed signal can occur in linear time using the cached LMS.

An important benefit of the Modified-Scholkmann algorithm is the ability to identify sub-features within a signal. An example is the recursive application of the algorithm to identify ICP sub-peaks (*Figure 3*) where Modified-Scholkmann is used firstly to identify all troughs allowing individual waveforms to be spliced, and Modified-Scholkmann is reapplied to each spliced waveform (which is efficient because of LMS caching) to identify the three largest

sub-peaks P_1 to P_3 . Here, the deliberate choice of a small maximum scale increases the false negative detection rate, increasing the algorithm's sub-peak detection sensitivity.

Conclusions

The accurate interpretation of neuroscience waveform data to identify peaks and troughs is crucial in signal parameterization, feature extraction and waveform identification tasks.

Modification of a standard scalogram technique has produced a robust algorithm with linear computational complexity that is particularly suited to the challenges presented by large, noisy physiological datasets. The algorithm can be tuned for specific applications by optimising the optional parameter, can identify sub-waveform features with minimal additional overhead, and is easily adapted to run in real-time on commodity hardware.

References

1. Sezan MI. A peak detection algorithm and its application to histogram-based image data reduction. *Comput Vision, Graph Image Process* [Internet]. 1990;49(1):36–51. Available from: <http://www.sciencedirect.com/science/article/pii/0734189X9090161N>
2. Wilbanks EG, Facciotti MT. Evaluation of Algorithm Performance in ChIP-Seq Peak Detection. Veenstra GJC, editor. *PLoS One* [Internet]. 2010 Jul 8 [cited 2016 Oct 30];5(7):e11471. Available from: <http://dx.plos.org/10.1371/journal.pone.0011471>
3. Du P, Kibbe WA, Lin SM. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics* [Internet]. 2006 Sep 1 [cited 2016 Oct 30];22(17):2059–65. Available from: <http://www.ncbi.nlm.nih.gov/pubmed/16820428>
4. Pan J, Tompkins WJ. A real-time QRS detection algorithm. *IEEE Trans Biomed Eng* [Internet]. 1985 Mar [cited 2016 Oct 30];32(3):230–6. Available from:

- <http://www.ncbi.nlm.nih.gov/pubmed/3997178>
5. Wee A, Grayden DB, Zhu Y, Petkovic-Duran K, Smith D. A continuous wavelet transform algorithm for peak detection. *Electrophoresis* [Internet]. 2008 Nov [cited 2016 Oct 30];29(20):4215–25. Available from:
<http://www.ncbi.nlm.nih.gov/pubmed/18924102>
 6. Coast DA, Stern RM, Cano GG, Briller SA. An approach to cardiac arrhythmia analysis using hidden Markov models. *IEEE Trans Biomed Eng* [Internet]. 1990 Sep [cited 2016 Oct 30];37(9):826–36. Available from:
<http://www.ncbi.nlm.nih.gov/pubmed/2227969>
 7. Mehta SS, Shete DA, Lingayat NS, Chouhan VS. K-means algorithm for the detection and delineation of QRS-complexes in Electrocardiogram. *Irbm*. 2010;31(1):48–54.
 8. Palshikar GK. Simple algorithms for peak detection in time-series. In: *First Int Conf Advanced Data Analysis, Business Analytics and Intelligence (ICADABAI2009)*, Ahmedabad, India, 6-7 June 2009. 2009.
 9. Scalzo F, Asgari S, Kim S, Bergsneider M, Hu X. Robust peak recognition in intracranial pressure signals. *Biomed Eng Online*. 2010;9(1):61.
 10. Scalzo F, Hamilton R, Hu X. Real-Time Analysis of Intracranial Pressure Waveform Morphology. *CdnIntechopenCom* [Internet]. Available from:
http://cdn.intechopen.com/pdfs/32480/InTech-Real_time_analysis_of_intracranial_pressure_waveform_morphology.pdf
 11. Scholkmann F, Boss J, Wolf M. An efficient algorithm for automatic peak detection in noisy periodic and quasi-periodic signals. *Algorithms*. 2012;5(4):588–603.

Appendix 1

The Matlab implementation of the Modified-Scholkmann algorithm (`new_peak_trough.m`):

```
% -----  
% Physiology Feature Extraction Toolkit  
% Dr Steven Bishop, 2015-16  
% Division of Anaesthesia, University of Cambridge, UK  
% Email: sbishop {AT} doctors.org.uk  
% -----  
% PEAK_TROUGH_FINDER  
%  
% Based upon the algorithm by (with updates and optimisations):  
% Scholkmann F, Boss J, Wolk M. An Efficient Algorithm for Automatic Peak  
% Detection in Noisy Periodic and Quasi-Periodic Signals. Algorithms 2012  
% (5), p588-603; doi:10.3390/a5040588  
% -----  
% [peaks,troughs,maximagram,minimagram] = PEAK_TROUGH_FINDER(data, {max-interval})  
% data: input data as vector  
% sampling_frequency (optional): sampling frequency of input  
% Returns: vectors [peaks, troughs, maximagram, minimagram] containing  
% indices of the peaks and troughs and the maxima/minima scalograms  
  
function [peaks,troughs,maximagram,minimagram] = new_peak_trough(data, varargin)  
  
    N = length(data);  
  
    if nargin == 2  
        L = ceil(varargin{1}/2) - 1;  
    else  
        L = ceil(N/2) - 1;  
    end  
  
    %Detrend the data  
    meanval = nanmean(data);  
    data(isnan(data)) = meanval;  
    data = detrend(data, 'linear');  
  
    Mx = zeros(N, L);  
    Mn = zeros(N, L);  
  
    %Produce the local maxima scalogram  
    for j=1:L
```

```

k = j;
for i=k+2:N-k+1
    if data(i-1) > data(i-k-1) && data(i-1) > data(i+k-1)
        Mx(i-1,j) = true;
    end
    if data(i-1) < data(i-k-1) && data(i-1) < data(i+k-1)
        Mn(i-1,j) = true;
    end
end
end

maxigram = Mx;
minigram = Mn;

%Form Y the column-wise count of where Mx is 0, a scale-dependent distribution of
%local maxima. Find d, the scale with the most maxima (== most number
%of zeros in row). Redimension Mx to contain only the first d scales
Y = sum(Mx==true, 1);
[~, d] = max(Y);
Mx = Mx(:,1:d);

%Form Y the column-wise count of where Mn is 0, a scale-dependent distribution of
%local minima. Find d, the scale with the most minima (== most number
%of zeros in row). Redimension Mn to contain only the first d scales
Y = sum(Mn==true, 1);
[~, d] = max(Y);
Mn = Mn(:,1:d);

%Form Zx and Zn the row-wise counts of Mx and Mn's non-zero elements.
%Any row with a zero count contains entirely zeros, thus indicating
%the presence of a peak or trough
Zx = sum(Mx==false, 2);
Zn = sum(Mn==false, 2);

%Find all the zeros in Zx and Zn. The indices of the zero counts
%correspond to the position of peaks and troughs respectively
peaks = find(~Zx);
troughs = find(~Zn);
end

```

Figure Captions

Figure 1: *Panels A–D (Modified-Scholkmann algorithm):* Detection of peaks (*red stars*) and troughs (*blue crosses*) in sinusoids degraded with white Gaussian noise to achieve signal-to-noise ratios (SNRs) down to 5dB. Note that the algorithm can miss the first and/or last features in a signal due to edge effects in the computed scalogram.

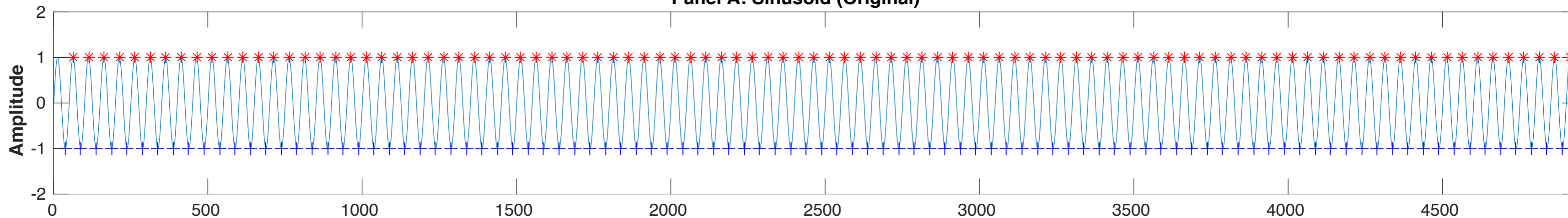
Figure 2: *Panels A–D (Modified-Scholkmann algorithm):* Detection of peaks (*red stars*) and troughs (*blue crosses*) in chirp sinusoids (frequency range 1Hz to 3.8Hz) degraded with white Gaussian noise to achieve signal-to-noise ratios (SNRs) down to 5dB. Note that the algorithm can miss the first and/or last features in a signal due to edge effects in the computed scalogram.

Figure 3:

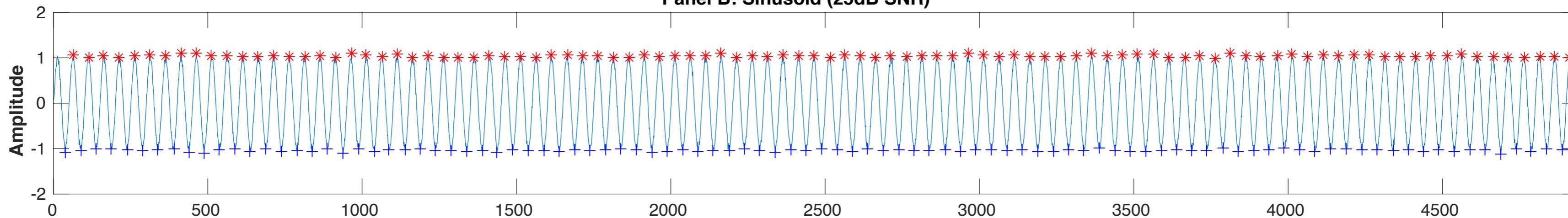
Panel A (Modified-Scholkmann algorithm): The detection of systole (*red stars*) and diastole (*blue stars*) in a continuous recording of intracranial pressure (ICP), sampled at 240Hz via a Codman ICP Express Monitoring System (*DePuySyntheses, Massachusetts, USA*) and pre-processed with an unweighted 25ms moving-average filter. A sliding window technique is used to mitigate scalogram edge effects and ensure that all peaks and troughs are accurately detected.

Panel B (Modified-Scholkmann algorithm): Recursive application of the algorithm to individual ICP waveforms delineated by trough-trough interval (*blue stars*) demonstrates the correct identification of individual ICP sub-peaks P1 to P3 (*red stars*).

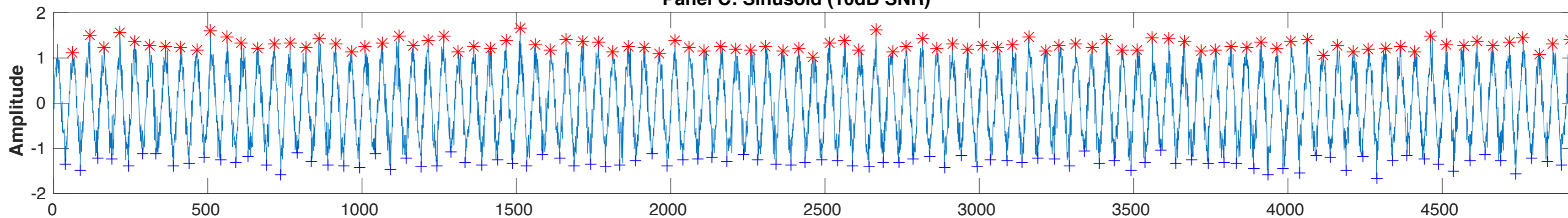
Panel A: Sinusoid (Original)



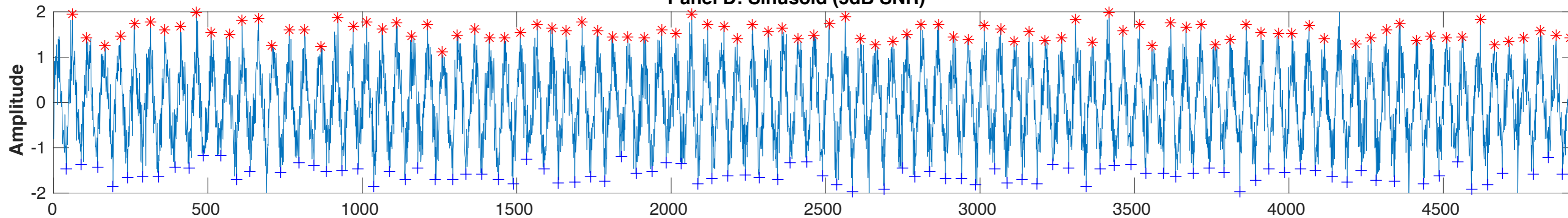
Panel B: Sinusoid (25dB SNR)



Panel C: Sinusoid (10dB SNR)

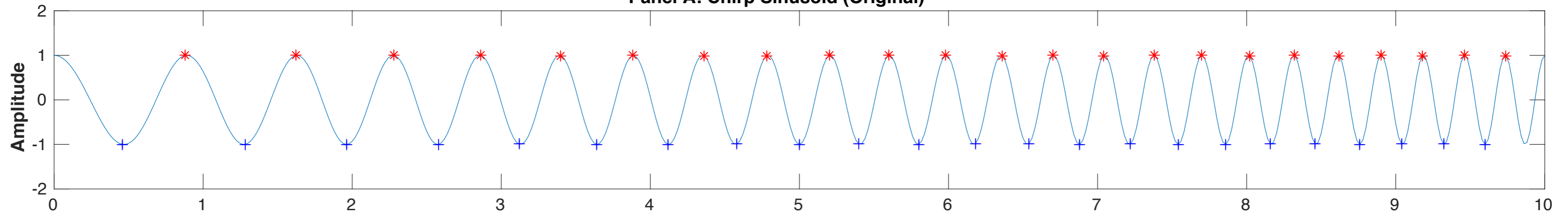


Panel D: Sinusoid (5dB SNR)

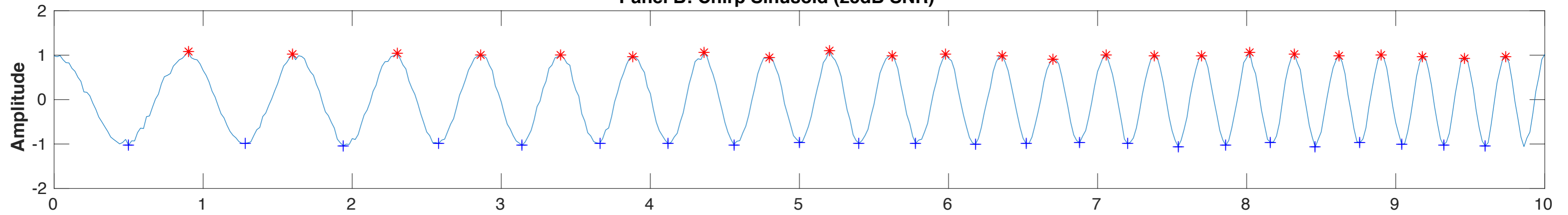


Time [seconds]

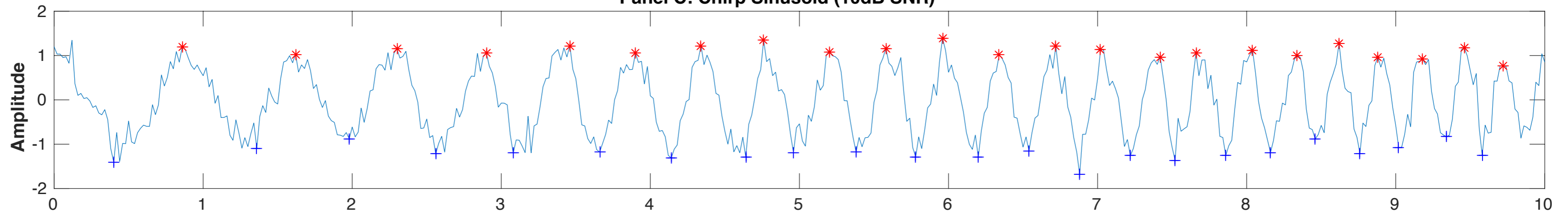
Panel A: Chirp Sinusoid (Original)



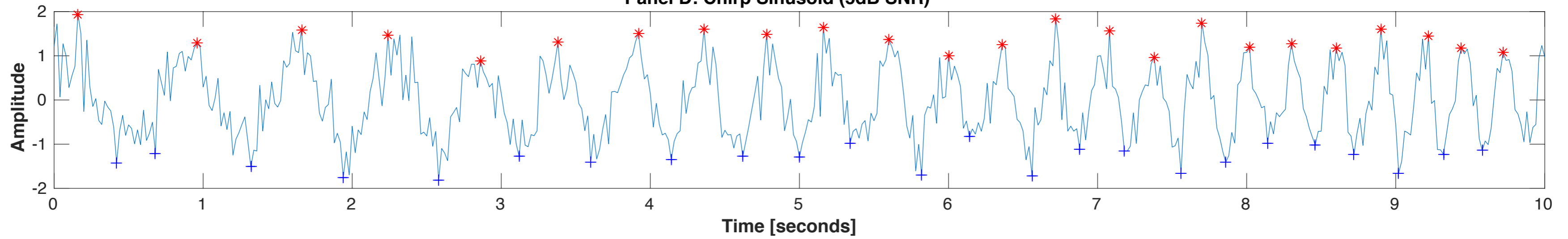
Panel B: Chirp Sinusoid (25dB SNR)



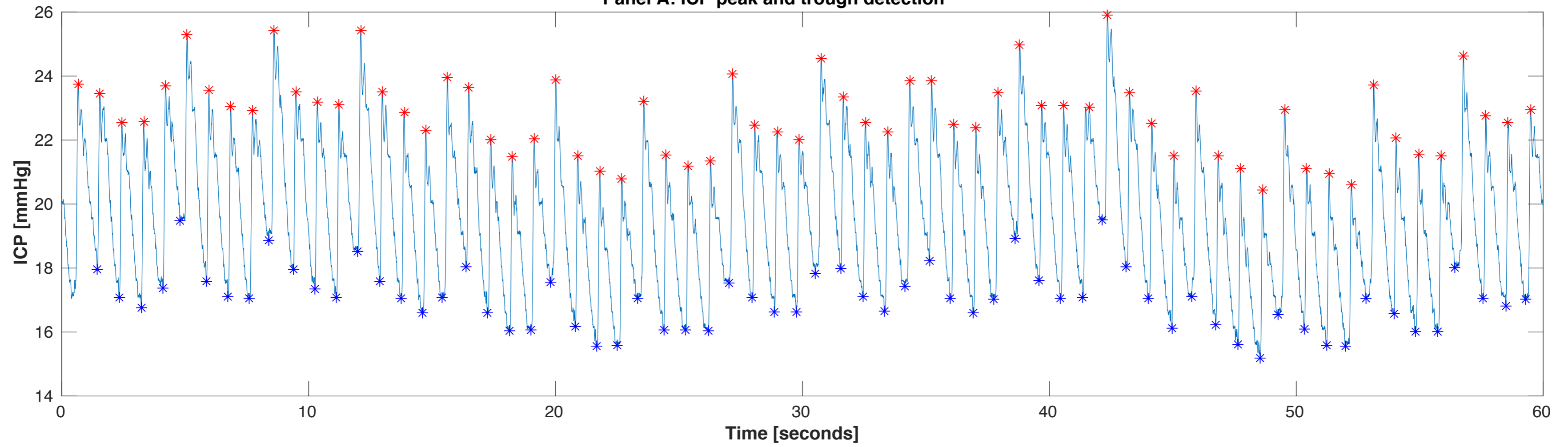
Panel C: Chirp Sinusoid (10dB SNR)



Panel D: Chirp Sinusoid (5dB SNR)



Panel A: ICP peak and trough detection



Panel B: ICP sub-peak detection

