

Article

Fast Spectral Search Using Improved Preprocessing and Limited Axis Check

YoungJae Son ¹, Tiejun Chen ², Guangyong Shang ², Myeongjin Kim ^{1,3,*} and Sung-June Baek ^{1,*}¹ Department of Intelligent Electronics and Computer Engineering, Chonnam National University, Gwangju 61186, Republic of Korea; ya_ho@jnu.ac.kr² Inspur Yunzhou Industrial Internet Co., Ltd., Jinan 250101, China; chentiejun@inspur.com (T.C.); shangguangyong@inspur.com (G.S.)³ Research Center for Biological Cybernetics, Chonnam National University, Gwangju 61186, Republic of Korea

* Correspondence: myeongjin@jnu.ac.kr (M.K.); tozero@jnu.ac.kr (S.-J.B.)

Abstract

Efficient and accurate identification of spectra from large databases remains a critical challenge in spectroscopic analysis. Previous coarse-to-fine frameworks, typically combining Principal Component Analysis (PCA)-based preprocessing and k - d tree search, have shown that structured search can reduce computational cost without sacrificing accuracy. Building on this foundation, we propose an enhanced algorithm that integrates an improved preprocessing and a novel limited axis check (LAC) method. The preprocessing stage applies running average filtering, downsampling, and threshold-based noise-cutting, followed by PCA to construct a compact, noise-suppressed spectral representation. In the search stage, the proposed LAC algorithm replaces conventional tree-based structures by performing an axis-wise limited-range search and voting strategy to efficiently locate the candidate spectrum closest to the query within the reduced PCA domain. A subsequent refined search determines the closest spectrum by computing distances to the shortlisted candidates. Experimental results demonstrate that the proposed approach attains accuracy equivalent to that of the full search while markedly reducing computational complexity. These results confirm that the integration of enhanced preprocessing and LAC substantially accelerates the spectral search process.



Academic Editor: Leonid Piterbarg

Received: 25 November 2025

Revised: 8 December 2025

Accepted: 12 December 2025

Published: 14 December 2025

Citation: Son, Y.; Chen, T.; Shang, G.; Kim, M.; Baek, S.-J. Fast Spectral Search Using Improved Preprocessing and Limited Axis Check. *Mathematics* **2025**, *13*, 3983. <https://doi.org/10.3390/math13243983>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: limited axis check; fast spectral search; spectral identification; Fine Search**MSC:** 68P10

1. Introduction

Spectroscopic analysis serves as a cornerstone technique in material characterization, chemical identification, and biomedical diagnostics. Among various spectroscopic modalities, Raman spectroscopy provides distinct molecular fingerprints that allow highly specific identification of substances [1,2]. However, the rapid expansion of spectral libraries in both scale and chemical diversity, resulting from increasingly large and comprehensive spectral datasets, imposes growing computational demands on real-time and high-accuracy matching [3–5]. These challenges require efficient search strategies capable of handling large-scale data without compromising identification performance.

Recent advances in machine learning have achieved remarkable success in image and signal classification; however, their direct application to spectral identification remains limited [6]. Unlike typical image datasets containing numerous labeled samples per class,

spectral datasets often provide only one reference spectrum per material, making deep-learning-based approaches difficult to generalize [7,8]. Moreover, the demand for low-power, lightweight, and real-time spectral analysis, especially in portable or embedded spectroscopic devices, calls for search algorithms that are both computationally efficient and memory-conscious [9–11].

To address these challenges, research in this field has consistently advanced toward fast spectral search through dimensionality reduction and structured data representation techniques [12,13]. Among these, we proposed a coarse-to-fine framework that combines Principal Component Analysis (PCA), k - d tree search, and Fine Search (FS), along with a simple running average filter for spectral smoothing [14]. The framework first locates approximate matches in the PCA-reduced domain and subsequently refines them through precomputed distance comparisons. This approach effectively reduced search time compared to the full search. However, its tree-based structure remains sensitive to the underlying data distribution and requires recursive traversal with irregular memory access, which leads to additional computational overhead when applied to large spectral libraries [15,16].

In this study, we propose an improved spectral search algorithm designed to enhance both preprocessing and search efficiency. The proposed method incorporates a refined preprocessing pipeline that includes running average filtering, downsampling, and threshold-based noise removal, followed by Principal Component Analysis (PCA) for compact feature representation. In the search stage, we introduce a Limited Axis Check (LAC) method that replaces the traditional k - d tree search with an axis-wise limited-range voting mechanism. This strategy enables faster and more robust identification within the reduced-dimensional PCA space. The Fine Search (FS) stage from the previous framework is retained for candidate verification, ensuring results equivalent to those obtained through a full search.

To validate the effectiveness of the proposed method, comparative experiments were conducted under two configurations. First, the k - d tree-based framework and the proposed LAC algorithm were evaluated under conventional preprocessing conditions to isolate the effect of the search mechanism. Second, the impact of the enhanced preprocessing including downsampling and noise-cut, was analyzed to quantify the improvement in efficiency.

The remainder of this paper is organized as follows. Section 2 reviews the previous k - d tree-based spectral search framework. Section 3 presents the proposed preprocessing and limited axis check (LAC) algorithm in detail. Section 4 describes the experimental setup and results. Finally, Section 5 provides a summary of the experimental results and conclusions.

2. k - d Tree-Based Search Method

A coarse-to-fine spectral identification framework based on k - d tree [14] was proposed to achieve fast and accurate Raman spectral matching in large databases. The framework consisted of three main stages: preprocessing, k - d tree-based coarse search, and Fine Search (FS).

2.1. Preprocessing Stage

The preprocessing consisted of two primary steps: running average filtering of the input spectra and PCA for dimensionality reduction. The running average filter was applied to suppress high-frequency detector noise in the raw spectra. This filter smooths the spectral signal without distorting major spectral peak positions, as demonstrated in the previous study. Assuming that W is the width of the smoothing kernel, the smoothed signal $\mathbf{x}'$ can be expressed as:

$$x'_i = \frac{1}{W} \sum_{k=-W/2}^{W/2} x_{i+k}. \quad (1)$$

After smoothing, both query and library spectra were projected into a low-dimensional subspace via PCA to reduce computational complexity [17]. Given a spectral library matrix $\mathbf{Y} \in \mathbb{R}^{N \times M}$, the correlation matrix $\mathbf{R} = \mathbf{Y}\mathbf{Y}^T$ can be decomposed into eigenvectors and eigenvalues. The principal components $\mathbf{P}$ are obtained by projecting the original data onto the eigenvector space:

$$\mathbf{P} = \mathbf{V}^T \mathbf{Y}. \quad (2)$$

In the previous study, 16 principal components were found sufficient to preserve the major spectral variance while substantially reducing computation from the original 3300-dimensional spectra.

2.2. Coarse Search with k -d Tree

The PCA-transformed library spectra were organized into a k -d tree structure, where each node represented one reference spectrum. The tree recursively partitioned the data along alternating PCA axes, forming balanced subdivisions for efficient nearest-neighbor searches.

Given an input spectrum $\mathbf{x}$, the algorithm traverses the tree by comparing coordinate values along the current splitting axis to locate the nearest node [18,19]. This coarse search yields a compact set of approximately similar candidates, while minimizing unnecessary distance evaluations. However, because the partitioning order depends on the variance along PCA axes, the k -d tree can result in nonuniform traversal depths and redundant node visits in high-dimensional spaces. Although the average search time is often quoted as $O(\log N)$, this behavior can degrade substantially for high-dimensional spectral data.

In this implementation, one tree node is created for each reference spectrum. Each node stores a feature vector consisting of principal components (PCs), an index value, and references to two child nodes. Thus, for a library of size N , the indexing structure requires approximately $(3 + \text{PCs})N$ scalar-equivalent parameters.

2.3. Fine Search

To ensure that the final identification result exactly matches that of a full search, a Fine Search (FS) stage is performed following the coarse search. Let the input spectrum be $\mathbf{x}$ and the spectrum identified by the coarse search as the closest candidate be $\mathbf{y}_{\min}$. The FS procedure refines this result by iteratively evaluating neighboring spectra in order of increasing precomputed distance from $\mathbf{y}_{\min}$. Each neighboring spectrum $\mathbf{y}_{\min,n}$ corresponds to the n -th nearest spectrum to $\mathbf{y}_{\min}$, with distances $d(\mathbf{y}_{\min}, \mathbf{y}_{\min,n})$ precomputed and stored for rapid retrieval.

$$d(\mathbf{y}_{\min}, \mathbf{y}_{\min,n}) > 2d(\mathbf{x}, \mathbf{y}_{\min}), \quad n = 1, 2, \dots \quad (3)$$

If Equation (3) holds, no subsequent spectrum can be closer to $\mathbf{x}$ than $\mathbf{y}_{\min}$, and the search terminates. Otherwise, the actual distance between $\mathbf{x}$ and the neighboring spectrum is computed. If the following Equation (4) is satisfied, $\mathbf{y}_{\min,n}$ is updated as the new nearest spectrum, and the refinement process continues accordingly.

$$d(\mathbf{x}, \mathbf{y}_{\min,n}) < d(\mathbf{x}, \mathbf{y}_{\min}). \quad (4)$$

Through this iterative procedure, FS guarantees results identical to a full search while avoiding unnecessary distance computations. The precomputed distance table requires $N(N-1)/2$ stored distances for a library of size N , which introduces additional memory usage; however, this memory–performance trade-off enables fast verification and remains practical for embedded or computationally constrained systems requiring real-time identification.

3. Proposed Method

3.1. Preprocessing

The proposed method employs an enhanced preprocessing (EP) pipeline to construct a compact and noise-robust spectral representation. The EP integrates smoothing, down-sampling, threshold-based noise removal, and dimensionality reduction via PCA. This produces consistent low-dimensional features for both query and library spectra, forming a reliable foundation for the subsequent search stage.

Let $S(\cdot)$ denote the running average smoothing filter, $D(\cdot)$ represent uniform down-sampling, and $N(\cdot)$ be the threshold-based noise-cut operator. Let $PCT(\cdot)$ denote the principal component transformation operator that projects the processed data into a low-dimensional subspace. Given a raw input spectrum $\mathbf{x}$, the EP workflow can be expressed as

$$\mathbf{y} = PCT(N(D(S(\mathbf{x}))). \quad (5)$$

The smoothing filter, $S(\cdot)$, as defined in Equation (1), suppresses stochastic noise without altering spectral peak positions. The downsampling filter, $D(\cdot)$, reduces each spectrum from 3300 points to a predetermined number of points, eliminating redundant information while preserving the overall peak structures. In this work, the spectrum is divided into 100 bins, and the data within each bin are averaged.

$$D_i(\mathbf{x}) = \frac{1}{33} \sum_{k=i*33}^{i*33+32} x_k, \quad i = 0, 1, \dots, 99, \quad k = 0, 1, \dots, 3299. \quad (6)$$

Finally, the noise-cut operator, $N(\cdot)$, removes weak signals below a threshold τ . In this work, τ was set to 5% of the global maximum intensity, taking into account the maximum expected noise amplitude in the spectroscopic acquisition system. $N(\cdot)$ is defined as

$$N(x_i) = \begin{cases} x_i, & \text{if } x_i \geq \tau, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

The EP sequence for an example spectrum is illustrated in Figure 1. Figure 1a shows the raw Raman spectrum containing high-frequency noise components. Figure 1b presents the smoothed spectrum obtained using the running-average filter. Figure 1c depicts the result of the proposed preprocessing pipeline, where downsampling and noise-cutting are additionally applied. Despite the dimensionality reduction from 3300 to 100 points, the essential peak structures are well preserved, ensuring that the spectral characteristics remain suitable for rapid identification.

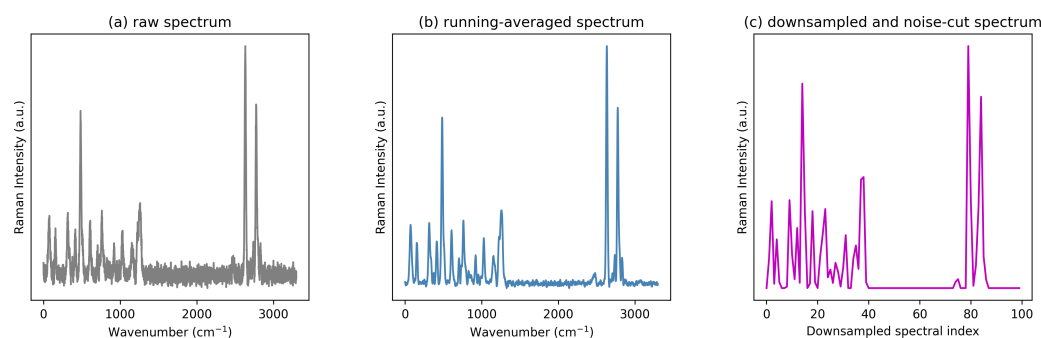


Figure 1. Raman spectrum of 1-Methylcyclohexanol (96%).

Finally, the noise-cut spectra are projected into a low-dimensional feature space using PCA, which preserves the essential variance structure of the data while substantially reducing the computational cost in subsequent search stages.

3.2. Limited Axis Check

The proposed limited axis check (LAC) efficiently identifies candidate spectra in the principal component (PC) space by restricting the search range along each principal axis. Unlike tree-based structures such as the k - d tree, which rely on recursive node traversal, LAC performs a direct axis-wise range search over pre-sorted feature data. This approach eliminates the overhead associated with recursive branching and provides consistent computational complexity, making it particularly well-suited for high-dimensional spectral datasets.

For an input spectrum $\mathbf{x}$ after PCA, LAC examines each component x_j independently. For the j -th component, a search range centered at x_j is defined with width Δ_j , predetermined as a fraction α of the total span of the library data along that axis:

$$[x_j - \Delta_j, x_j + \Delta_j], \quad \Delta_j = \frac{\alpha}{2} (\max(s_{:,j}) - \min(s_{:,j})), \quad (8)$$

where $s_{:,j}$ denotes the j -th component of all library spectra, and α is the predefined range ratio. Library spectra whose j -th component values fall within this range are collected as candidates for that axis, and their indices are appended to the candidate set $\mathcal{C}$.

In this work, α is set to $1/16$. If α is too large, the axis-wise filtering becomes loose and the number of surviving candidates increases, which raises the computational cost of the LAC stage. Conversely, an excessively small α narrows the valid range, reducing the likelihood of retaining the true neighbor and consequently increasing the workload of the Fine Search stage. Experimental evaluation confirmed that $\alpha = 1/16$ produced the minimum total arithmetic operations across both stages, and this value was therefore adopted in all experiments.

After the candidate set $\mathcal{C}$ is obtained, LAC finds the most frequently occurring spectra in $\mathcal{C}$, forming the set *near_set*. LAC then computes the squared Euclidean distance between the PCA transformed d -dimensional input spectrum $\mathbf{x}$ and each candidate $\mathbf{s}_i \in \text{near_set}$:

$$D(\mathbf{x}, \mathbf{s}_i) = \sum_{j=1}^d (x_j - s_{i,j})^2. \quad (9)$$

Since the only candidates within *near_set* are evaluated, distance computation is limited to a very small subset of the entire library, avoiding exhaustive comparison with all spectra. The spectrum with the smallest distance is selected as the nearest neighbor candidate:

$$i^* = \arg \min_{i \in \mathcal{C}} D(\mathbf{x}, \mathbf{s}_i). \quad (10)$$

The LAC implementation stores sorted PCA feature values and their corresponding indices along each axis, enabling efficient axis-wise candidate filtering. For a library of N spectra and principal components (PCs), approximately $2 \times \text{PCs} \times N$ scalar-equivalent parameters are required. Additionally, each axis range is evaluated via binary search on the sorted coordinates, so the average search time is $O(\log N)$. While the coarse-grained search index size can be larger than a k - d tree, the non-recursive search significantly reduces the computational overhead of candidate selection.

The corresponding pseudocode is presented in Algorithm 1, summarizing the limited axis check.

Algorithm 1: Limited Axis Check: LAC.**Input:** test spectrum $\mathbf{x}$, library spectra $\mathbf{S}$, search range Δ **Output:** the index of the nearest candidate i^*

```

for all axis  $j$  do
  for all  $s_i$  in  $\mathbf{S}$  do
    if  $s_{i,j} \in [x_j - \Delta_j, x_j + \Delta_j]$  then
       $\mathcal{C} = \mathcal{C} \cup i$ 
    end
  end
end

 $\text{max\_count} = \max(\text{count}(i) \text{ for } i \in \mathcal{C})$ 
 $\text{near\_set} = \{i : \text{max\_count} == \text{count}(i) \text{ for } i \in \mathcal{C}\}$ 
compute Euclidean distance  $D(\mathbf{x}, \mathbf{s}_i)$  for all  $i \in \text{near\_set}$ 

return  $i^* = \arg \min_i D(\mathbf{x}, \mathbf{s}_i)$ 

```

3.3. Fine Search

The Fine Search (FS) stage refines the coarse candidate retrieved by LAC to ensure exact identification. Starting from the selected candidate, FS performs distance-based refinement using the precomputed index table described in our previous study [14].

While the internal algorithm remains identical to the previous implementation, its integration with LAC significantly reduces the number of distance evaluations by limiting the initial search space. Figure 2 illustrates the overall workflow of the proposed framework, including the EP pipeline and LAC + FS integration.

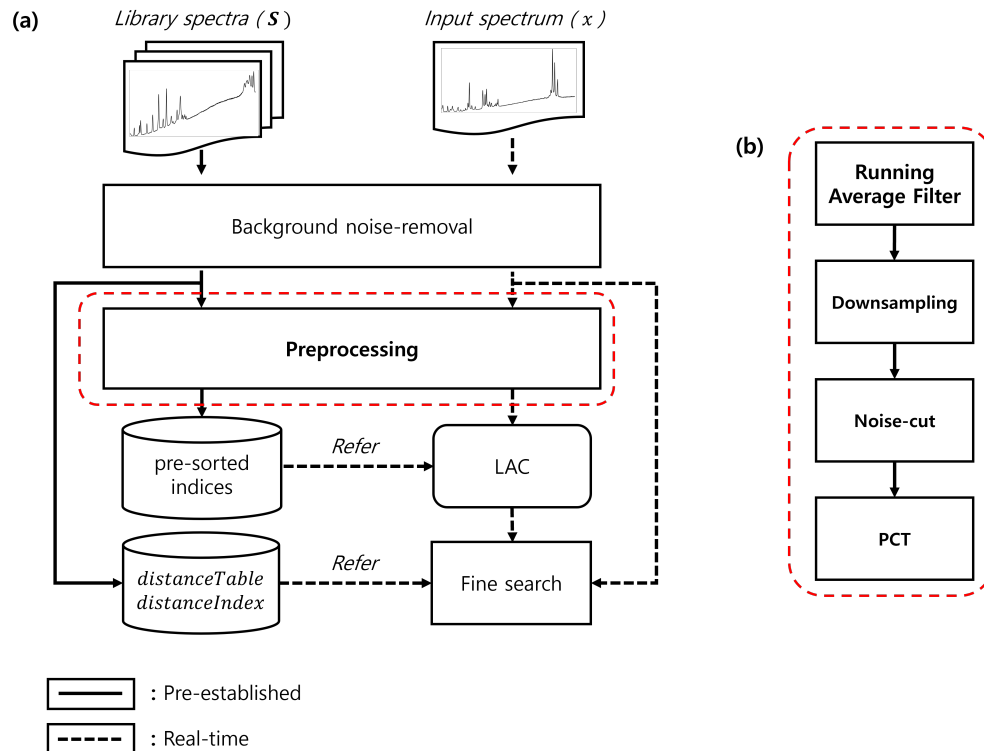


Figure 2. Overview of the proposed algorithm: (a) Overall flowchart of the proposed algorithm. (b) Enhanced preprocessing pipeline.

4. Experiment

All experiments were conducted using the same Raman spectrum library described in our previous study [14]. The database comprises 14,085 chemical substances, each represented by a Raman spectrum of size 3300. To ensure fair comparison, the same pre-processing and baseline removal procedures as in the previous work were applied [20]. For performance evaluation, 2817 spectra (approximately 20% of the database) were randomly selected as test queries. Additive white Gaussian noise (15, 20, and 24 dB) was introduced to simulate different measurement conditions. The primary evaluation metrics were the average numbers of multiplication and addition operations per input spectrum.

4.1. Comparison Under Conventional Preprocessing

To isolate the effect of the proposed search framework itself, the first experiment was conducted without applying the newly proposed preprocessing pipeline. Instead, both the conventional k - d tree-based method and the proposed limited axis check (LAC) were tested under the same preprocessing condition, in which only a running-average filter was applied to the spectra to suppress high-frequency noise. The number of principal components was fixed at 16 for both methods.

Table 1 summarizes the number of arithmetic operations measured during the coarse and FS stages. Since both methods ultimately employ the same FS stage, which exhaustively verifies candidates, their identification results are identical to those of a full search; however, the proposed LAC + FS requires fewer operations.

Table 1. Operation counts under identical preprocessing (running average only) with 16 PCs.

Method	Coarse Search		Fine Search (FS)		Total Ops	Reduction
	Mul	Add	Mul	Add		
KD + FS	67,760	60,752	11,872	23,739	164,123	—
LAC + FS	52,819	58,837	8824	17,644	138,124	15%

Even without the proposed preprocessing, LAC + FS achieved a 15% reduction in total arithmetic operations compared with the k - d tree-based approach. In addition, k - d tree rely on PCA-space distances, which can be distorted by noise and lead to suboptimal candidates being forwarded to the Fine Search stage. The simple axis-wise filtering of the LAC method tends to retain candidates with spectra that retain a more uniform overall shape, reducing the Fine Search overhead. The impact of the proposed preprocessing pipeline is analyzed in the following section.

4.2. Evaluation of LAC with EP

The next experiment evaluated the performance of the proposed LAC + FS framework when combined with the EP pipeline. Since previous results have demonstrated that LAC outperforms the k - d tree approach, this experiment focuses exclusively on the proposed method. First, we determine the optimal number of principal components and investigate how the EP affects overall search efficiency.

Table 2 summarizes the number of arithmetic operations for different numbers of principal components (PCs). As expected, the number of LAC operations increases almost linearly with PCA dimensionality, whereas FS operations decrease sharply and converge beyond 16 PCs. This trend suggests that the EP enhances feature separability in the PCA space, enabling LAC to exclude irrelevant candidates early and thereby reducing the computational load of FS.

Table 2. Arithmetic operation counts of the proposed LAC + FS method with the EP pipeline under varying PCA dimensionality.

Number of PCs	LAC		Fine Search (FS)		LAC + FS (Total)	
	Multiplication	Addition	Multiplication	Addition	Multiplication	Addition
4	447	6483	920,590	1,840,882	921,037	1,847,365
8	817	6833	199,076	398,088	199,893	404,921
16	1618	7635	8010	16,016	9628	23,651
32	3234	9268	7603	15,203	10,837	24,471
64	6468	12,536	7603	15,203	14,071	27,739

The results demonstrate that the enhanced preprocessing (EP) pipeline contributes significantly to improving search efficiency. Without these steps, the spectral features remain high-dimensional and noise-contaminated, forcing LAC to examine more candidates and consequently increasing FS operations. In contrast, the combined use of downsampling and noise-cut removes redundant local fluctuations while emphasizing dominant spectral peaks, enabling LAC to identify the correct neighborhood with minimal distance computations.

A substantial reduction in Fine Search operations is observed at 16 PCs compared with 8 PCs, indicating that most incorrect candidates are already removed during coarse filtering. Increasing the number of PCs to 32 or 64 yields negligible improvement, confirming that 16 PCs represent an efficient operating point where the proposed LAC + FS framework maintains exact identification results while minimizing computational cost.

The computational cost of the EP stage was also evaluated to assess the overall efficiency of the proposed method. The EP phase consists of a fixed number of arithmetic operations including running-average smoothing, downsampling, and threshold-based noise-cutting. The corresponding arithmetic operation counts for each stage are summarized in Table 3.

Table 3. Total arithmetic operation counts per query for the proposed framework at 16 PCs.

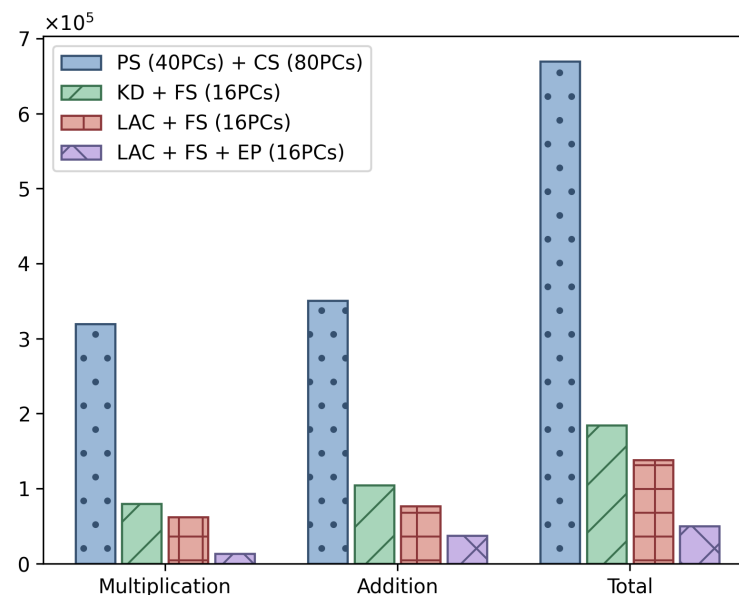
Stage	Mul	Add	Description
LAC	1618	7635	Candidate identification
Fine Search (FS)	8010	16,016	Exact matching using precomputed distance table
Enhanced Preprocessing (EP)	3400	13,200	Running average, downsampling, noise-cut
Total (Proposed)	13,028	36,851	—

Although the preprocessing stage introduces additional arithmetic operations, the resulting reduction in data dimensionality and effective noise suppression substantially accelerate both the LAC and FS stages, leading to a significantly lower overall computational load even when the preprocessing cost is included.

As summarized in Table 4, the proposed LAC + FS + EP framework achieves a substantial improvement in computational efficiency over previous methods. Earlier PCA-based and Partial Distortion Search (PDS) approaches offered only limited reductions in computational complexity. Although the KD + FS framework was previously the most efficient, the integration of the proposed preprocessing and LAC further reduced the total arithmetic operations by more than 72%. The overall average results of the main algorithms are shown in Figure 3.

Table 4. Comparison of total operations of each algorithm.

Method	Multiplication	Addition	Total
Full Search	46,480,500	92,946,915	139,427,415
Full Search + PDS	8,846,894	17,679,704	26,526,598
PCT + PDS (150 PCs)	822,617	1,135,948	1,958,565
PS (40 PCs) + CS (80 PCs)	319,385	350,336	669,721
KD + FS (16 PCs)	79,633	104,475	184,108
LAC + FS + EP (16 PCs)	13,028	36,851	49,879

**Figure 3.** Comparison of arithmetic operations in major algorithms.

To demonstrate end-to-end runtime under the same PCA configuration (16 PCs), Table 5 summarizes the average results obtained from multiple executions using different combinations of 2817 query spectra.

Table 5. Runtime comparison under the same PCA configuration (16 PCs), averaged over 50 independent runs.

Method	Time (s)	Speedup	Environment
KD + FS	7.12	1.00×	Windows 11 (64-bit) AMD Ryzen 9 5900X
LAC + FS + EP	2.96	2.40×	64 GB RAM, Python 3.11.10

While absolute execution times may vary depending on the computational environment, the observed performance consistently demonstrates the superiority of the proposed LAC + FS + EP method. This improvement is primarily due to eliminating the recursive traversal and irregular memory access patterns inherent in the k - d tree search.

This result demonstrates that the proposed framework attains full search accuracy while dramatically reducing computational cost, thereby enabling real-time spectral identification even on low-power or embedded hardware platforms.

5. Conclusions

This study presented an efficient spectral identification framework that integrates enhanced preprocessing (EP) with a novel LAC algorithm. The EP pipeline, comprising running-average smoothing, uniform downsampling, and threshold-based noise-cutting,

effectively removes redundant and noisy spectral components, leading to a compact and discriminative input representation. Subsequent dimensionality reduction through PCA further simplifies the data while preserving critical spectral features.

Building on this preprocessed representation, the LAC algorithm efficiently identifies candidate spectra by axis-limited searching rather than recursively partitioned spaces as in conventional k - d tree search. A final fine step guarantees that the identification results are identical to those obtained from a full exhaustive search.

Experimental evaluations demonstrated that the EP and LAC framework achieved the same identification accuracy as the full search while reducing total arithmetic operations by approximately one-quarter compared to the previous k - d tree + FS approach. Although the Fine Search stage requires additional memory for storing the precomputed distance table and index structure, its low arithmetic complexity ensures high efficiency even in CPU-constrained environments. Overall, the proposed framework offers a promising solution for real-time and embedded spectral search applications.

Author Contributions: Conceptualization, Y.S., M.K. and S.-J.B.; methodology, Y.S., M.K. and S.-J.B.; validation, Y.S., T.C. and S.-J.B.; formal analysis, Y.S. and S.-J.B.; investigation, T.C. and G.S.; resources, Y.S. and S.-J.B.; data curation, T.C., G.S. and M.K.; writing—review and editing, Y.S., M.K. and S.-J.B.; visualization, Y.S. and T.C.; supervision, M.K. and S.-J.B.; project administration, M.K. and S.-J.B.; funding acquisition, M.K. and S.-J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by Innovative Human Resource Development for Local Intellectualization program (MSIT) (IITP-2025-RS-2022-00156287, 50%) and ICAN(ICT Challenge and Advanced Network of HRD) support program (IITP-2025-RS-2022-00156385, 30%) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation) grant funded by the Korea government(MSIT), and the Advanced R&D team of Digital Appliances Business by the Samsung Electronics (20%).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: Authors Tiejun Chen and Guangyong Shang were employed by the Inspur Yunzhou Industrial Internet Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Hu, Y.; Jiang, T.; Shen, A.; Li, W.; Wang, X.; Hu, J. A background elimination method based on wavelet transform for Raman spectra. *Chemom. Intell. Lab. Syst.* **2007**, *85*, 94–101. [[CrossRef](#)]
2. Sigle, M.; Rohlfing, A.K.; Kenny, M.; Scheuermann, S.; Sun, N.; Graefner, U.; Haug, U.; Sudmann, J.; Seitz, C.M.; Heinzmann, D.; et al. Translating genomic tools to Raman spectroscopy analysis enables high-dimensional tissue characterization on molecular resolution. *Nat. Commun.* **2023**, *14*, 5799. [[CrossRef](#)] [[PubMed](#)]
3. Li, J.; Chu, X.; Tian, S.; Lu, W. The identification of highly similar crude oils by infrared spectroscopy combined with pattern recognition method. *Spectrochim. Acta Part B* **2013**, *112*, 457–462. [[CrossRef](#)] [[PubMed](#)]
4. Iscen, A.; Avrithis, Y.; Tolia, G.; Furon, T.; Chum, O. Fast spectral ranking for similarity search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7632–7641. [[CrossRef](#)]
5. Wang, J.; Shen, J. Fast spectral analysis for approximate nearest neighbor search. *Mach. Learn.* **2022**, *111*, 2297–2322. [[CrossRef](#)]
6. Chen, T.; Son, Y.; Dong, C.; Baek, S.-J. Baseline correction of Raman spectral data using triangular deep convolutional networks. *Analyst* **2025**, *150*, 2653–2660. [[CrossRef](#)] [[PubMed](#)]
7. Liu, X.; An, H.; Cai, W.; Shao, X. Deep learning in spectral analysis: Modeling and imaging. *TrAC Trends Anal. Chem.* **2024**, *172*, 117612. [[CrossRef](#)]
8. Ho, C.S.; Jean, N.; Hogan, C.A.; Blackmon, L.; Jeffrey, S.S.; Holodniy, M.; Banaei, N.; Saleh, A.A.; Ermon, S.; Dionne, J. Rapid identification of pathogenic bacteria using Raman spectroscopy and deep learning. *Nat. Commun.* **2019**, *10*, 4927. [[CrossRef](#)] [[PubMed](#)]

9. Hajjou, M.; Qin, Y.; Bradby, S.; Bempong, D.; Lukulay, P. Assessment of the performance of a handheld Raman device for potential use as a screening tool in evaluating medicines quality. *J. Pharm. Biomed. Anal.* **2013**, *74*, 47–55. [[CrossRef](#)]
10. Sanchez, L.; Farber, C.; Lei, J.; Zhu-Salzman, K.; Kurouski, D. Noninvasive and nondestructive detection of cowpea bruchid within cowpea seeds with a hand-held Raman spectrometer. *Anal. Chem.* **2019**, *91*, 1733–1737. [[CrossRef](#)] [[PubMed](#)]
11. Koyun, O.C.; Keser, R.K.; Şahin, S.O.; Bulut, D.; Yorulmaz, M.; Yücesoy, V.; Töreyn, B.U. RamanFormer: A transformer-based quantification approach for Raman mixture components. *ACS Omega* **2024**, *9*, 23241–23251. [[CrossRef](#)]
12. Lackey, H.E.; Nelson, G.L.; Felmy, H.M.; Guo, X.; Bryan, S.A.; Lines, A.M. PCA and PLS analysis of lanthanides using absorbance and single-beam visible spectra. *ACS Omega* **2024**, *9*, 33662–33670. [[CrossRef](#)]
13. Tai, S.C.; Lai, C.C.; Lin, Y.C. Two fast nearest neighbor searching algorithms for image vector quantization. *IEEE Trans. Commun.* **1996**, *44*, 1623–1628. [[CrossRef](#)]
14. Son, Y.; Chen, T.; Baek, S.-J. Fast search using k-d trees with Fine Search for spectral data identification. *Mathematics* **2025**, *13*, 574. [[CrossRef](#)]
15. Huang, Z.; Laffan, S.W. Sensitivity analysis of a decision tree classification to input data errors using a general Monte Carlo error sensitivity model. *Int. J. Geogr. Inf. Sci.* **2009**, *23*, 1433–1452. [[CrossRef](#)]
16. Magniez, F.; Nayak, A.; Santha, M.; Sherman, J.; Tardos, G.; Xiao, D. Improved bounds for the randomized decision tree complexity of recursive majority. *Random Struct. Algorithms* **2016**, *48*, 612–638. [[CrossRef](#)]
17. Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1901**, *2*, 559–572. [[CrossRef](#)]
18. Bentley, J.L. Multidimensional binary search trees used for associative searching. *Commun. ACM* **1975**, *18*, 509–517. [[CrossRef](#)]
19. Tiwari, V.R. Developments in KD Tree and KNN Searches. *Int. J. Comput. Appl.* **2023**, *185*, 17–23. [[CrossRef](#)]
20. Press, W.H.; Flannery, B.P.; Teukolsky, S.A.; Vetterling, W.T. *Numerical Recipes in C*; Cambridge University Press: New York, NY, USA, 1988.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.