Noise Source Detection Using BFS-DFS Graph Traversal and A* Algorithm Implementation

Bevinda Vivian - 13523120
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: bevindavivian@gmail.com, 13523120@std.stei.itb.ac.id

***Abstract*— Noise pollution is a growing concern in urban and campus environments, necessitating efficient and accurate detection of noise sources. This study presents the design and implementation of an enhanced noise source detection system utilizing classical graph traversal algorithms, such as Breadth-First Search (BFS), Depth-First Search (DFS), and the A* search algorithm. The system models the environment as a grid-based graph, where each node represents a spatial location with associated noise levels, and integrates real-time audio processing for dynamic updates. Through comprehensive experiments, the system demonstrates the strengths and trade-offs of each algorithm in terms of path optimality, computational efficiency, and adaptability to real-world scenarios. The results highlight the effectiveness of combining graph-based search strategies with audio analysis for robust noise source localization. This approach provides a scalable foundation for future developments in environmental monitoring and smart city applications.**

Keywords—noise source detection; graph traversal; BFS; DFS; A* algorithm; audio processing; environmental monitoring; smart city;

I. Introduction

In the contemporary era of rapid urbanization and technological advancement, environmental challenges have become increasingly complex and demanding of innovative solutions. Among these challenges, noise pollution stands as one of the most pervasive yet underaddressed issues affecting urban populations worldwide. The World Health Organization estimates that environmental noise pollution affects over 100 million people in Europe alone, with similar alarming trends observed globally, including in rapidly developing nations like Indonesia where urban centers continue to expand at unprecedented rates.

Modern urban environments are characterized by multiple simultaneous noise sources operating in complex spatial and temporal patterns. Vehicular traffic creates continuous low-frequency rumble along transportation corridors, industrial facilities generate high-amplitude periodic noise, construction activities produce variable-intensity disturbances, and aircraft movements contribute intermittent but significant acoustic events. These diverse noise sources create intricate acoustic landscapes that require sophisticated monitoring, analysis, and management strategies to protect public health and maintain quality of life for urban residents.

The health implications of prolonged exposure to environmental noise are well-documented and increasingly concerning. Research indicates that chronic noise exposure contributes to cardiovascular disease, sleep disorders, cognitive impairment, and psychological stress, particularly affecting vulnerable populations including children, elderly individuals, and those with pre-existing health conditions. In educational environments, excessive noise levels have been shown to significantly impact learning outcomes, concentration abilities, and academic performance, making noise management particularly critical for institutions of higher learning.

Traditional approaches to noise source identification and monitoring have relied heavily on manual surveys, static measurement stations, and reactive complaint-based systems. These conventional methods suffer from significant limitations including high operational costs, limited spatial coverage, delayed response times, and inability to provide real-time actionable information. Static monitoring stations, while providing accurate point measurements, cannot effectively track mobile noise sources or provide comprehensive coverage of large urban areas. Manual surveys, though detailed, are labor-intensive, time-consuming, and cannot operate continuously to capture temporal variations in noise patterns.

The emergence of smart city initiatives and the proliferation of Internet of Things (IoT) technologies have created unprecedented opportunities for developing automated, intelligent environmental monitoring systems. Graph-based computational approaches, particularly graph traversal algorithms, offer promising solutions for modeling and analyzing spatial relationships in complex urban noise environments. By representing urban spaces as mathematical graphs where nodes correspond to measurement points and edges represent spatial connectivity or acoustic propagation paths, classical computer science algorithms can be adapted and applied to solve real-world environmental challenges with remarkable efficiency and accuracy.

Indonesia, as a rapidly developing archipelagic nation with significant urban centers including Jakarta, Surabaya, Bandung, and Medan, faces particular challenges related to noise pollution management. The country's Ministry of Environment and Forestry reports that urban noise levels frequently exceed World Health Organization recommended limits, particularly during peak traffic hours, in industrial zones, and near major transportation hubs including airports

and seaports. The rapid economic growth and increasing urbanization have intensified these challenges, creating urgent needs for systematic, technology-driven approaches to environmental noise management.

Educational institutions such as Institut Teknologi Bandung (ITB) serve as representative microcosms of urban environments where effective noise management is essential for optimal learning, research, and academic activities. University campuses typically feature diverse noise sources including vehicular traffic, construction activities, HVAC systems, and social gatherings, making them ideal environments for developing and testing noise source detection technologies. The student population's familiarity with technology and data-driven approaches also provides opportunities for participatory monitoring and community engagement in environmental management initiatives.

Recent advances in computational algorithms, particularly in the domain of graph theory and pathfinding, have demonstrated remarkable potential for spatial optimization problems. Breadth-First Search (BFS) algorithms provide comprehensive exploration capabilities with guaranteed optimal solutions for unweighted graphs, making them suitable for scenarios requiring exhaustive noise source identification. Depth-First Search (DFS) algorithms offer memory-efficient exploration strategies that can be particularly effective for hierarchical or tree-like urban structures. The A* (A-star) algorithm combines the optimality guarantees of BFS with heuristic-driven efficiency, potentially providing superior performance for real-time applications where computational resources are constrained.

The integration of real-time audio processing capabilities with classical graph traversal algorithms represents a novel approach to environmental monitoring that bridges theoretical computer science with practical environmental engineering. Modern digital signal processing techniques enable real-time analysis of acoustic characteristics including frequency content, amplitude variations, and temporal patterns, providing rich data streams that can inform intelligent decision-making algorithms. Java's robust audio processing APIs and cross-platform compatibility make it an ideal platform for developing deployable environmental monitoring solutions that can operate across diverse hardware and operating system environments.

This research addresses the critical intersection of environmental monitoring needs and computational algorithm capabilities by developing, implementing, and evaluating a comprehensive noise source detection system based on graph traversal algorithms. The study makes several significant contributions to both environmental monitoring and computational algorithm research domains. First, it provides detailed implementation and performance analysis of three fundamental graph traversal algorithms specifically adapted for environmental noise source detection applications. Second, it integrates real-time audio processing capabilities with classical search algorithms, enabling practical deployment in dynamic urban environments. Third, it develops both console-based and graphical user interface implementations to accommodate different user preferences and deployment scenarios. Fourth, it evaluates algorithm performance under various simulated urban noise scenarios, providing empirical evidence for algorithm selection in different environmental contexts.

The primary research objectives include (1) implementing and optimizing BFS, DFS, and A* algorithms for spatial noise source detection in grid-based urban environment models, (2) integrating real-time audio processing capabilities to enable practical deployment with actual environmental audio data, (3) developing user-friendly interfaces that facilitate both research applications and practical field deployment, (4) conducting comprehensive performance evaluation to identify optimal algorithm selection criteria for different noise source scenarios, and (5) providing open-source implementation that enables further research and development in computational environmental monitoring.

The significance of this work extends beyond academic research to practical applications in smart city development, environmental regulation compliance, public health protection, and urban planning optimization. As cities worldwide continue to grow and environmental challenges intensify, automated monitoring systems like the one developed in this research will become increasingly essential for maintaining sustainable, livable urban environments. The combination of efficient algorithms, real-time processing capabilities, and accessible user interfaces positions this system as a valuable foundation for next-generation environmental monitoring infrastructure that can adapt to evolving urban needs and technological capabilities.

II. BASIC THEORY

A. Sound and Noise Source

First Sound is a mechanical wave that propagates through air or other media by creating pressure variations. In the context of environmental monitoring and urban planning, noise pollution has become a significant concern that affects public health and quality of life. Noise source detection refers to the systematic identification and localization of sound-generating sources within a given environment. This process involves analyzing acoustic signals to determine their origin points, intensity levels, and characteristics. The importance of noise source detection extends beyond simple environmental monitoring; it plays a crucial role in urban planning, industrial safety, and public health management.

In digital noise source detection systems, the environment is typically modeled as a discrete grid where each point represents a potential location that can generate or transmit sound. Each grid point, or node, contains acoustic properties such as noise level, source type, and spatial coordinates. The noise level at any given point is quantified using a normalized scale, typically ranging from 0.0 to 1.0, where higher values indicate greater acoustic intensity. A threshold value, commonly set at 0.7, is used to distinguish between regular ambient noise and significant noise sources that require attention. This threshold-based approach allows for efficient classification of acoustic environments and facilitates automated decision-making in noise management systems.

The spatial relationship between noise sources and their surrounding environment is fundamental to understanding acoustic propagation patterns. Sound waves follow the inverse square law, where intensity decreases proportionally to the square of the distance from the source. This principle is mathematically expressed as

$$I = \frac{P}{4\pi r^2} \tag{1}$$

where I represents intensity, P is the source power, and r is the distance from the source.

Modern noise detection systems often incorporate real-time audio processing capabilities to enhance their effectiveness. Audio signals are typically sampled at high frequencies, such as 44.1 kHz (CD quality), and converted into digital format for analysis. The Root Mean Square (RMS) level calculation, given by

$$RMS = \sqrt{\frac{\Sigma x^2}{N}} \tag{2}$$

where x represents individual audio samples and N is the total number of samples, provides a measure of the signal's average power. This RMS value is then normalized to a 0-1 scale to facilitate integration with grid-based detection algorithms. Advanced systems may also employ frequency analysis to distinguish between different types of noise sources, such as traffic noise (typically low-frequency), construction noise (broadband with high energy), and aircraft noise (characterized by specific frequency patterns).

B. Graph Theory and Spatial Representation

Graph theory provides the mathematical foundation for representing spatial environments in noise source detection systems. In this context, a graph G = (V, E) consists of a set of vertices V representing spatial locations and a set of edges E representing connections or relationships between these locations. For noise source detection applications, the environment is typically modeled as a grid graph where each cell represents a potential acoustic measurement point. This grid-based approach offers several advantages, including uniform spatial resolution, simplified path calculation, and efficient algorithmic processing.

In a two-dimensional grid representation, each node is characterized by its coordinates (x, y) and associated acoustic properties. The connectivity between nodes follows a neighborhood pattern, typically employing 4-connectivity where each internal node connects to its immediate neighbors (up, down, left, right). This connectivity pattern ensures that sound propagation paths can be traced through adjacent cells, providing a realistic approximation of how acoustic waves travel through space. The adjacency relationships are crucial for pathfinding algorithms, as they define the valid moves that can be made when searching for noise sources.

The grid size significantly impacts both the accuracy and computational complexity of the detection system. A 5×5 grid, containing 25 nodes, provides a manageable environment for algorithm demonstration while maintaining sufficient complexity to showcase different search strategies. Each node in the grid stores multiple attributes including its unique identifier (0-24 for a 5×5 grid), spatial coordinates, current noise level, noise source status, and connectivity information. The node identifier follows a row-major ordering scheme where node ID = row × grid_size + column, facilitating efficient indexing and neighbor calculation.

Graph traversal algorithms operate on this spatial representation by systematically visiting nodes to locate acoustic sources. The choice of traversal strategy significantly affects both the efficiency and effectiveness of the detection process. The graph structure also supports the calculation of path costs, which can incorporate factors such as physical distance, acoustic attenuation, and environmental obstacles. This cost information is particularly important for optimization algorithms that seek to find the most efficient routes to detected noise sources.

C. Breadth-First Search (BFS) Algorithm

Breadth-First Search is a fundamental graph traversal algorithm that explores vertices in order of their distance from the starting point. In the context of noise source detection, BFS provides a systematic approach to searching for acoustic sources by examining all nodes at distance k before exploring nodes at distance k+1. This level-by-level exploration pattern ensures that the algorithm discovers the nearest noise sources first, making it particularly suitable for applications where proximity to the detection origin is prioritized.

The BFS algorithm maintains a queue data structure to manage the order of node exploration. Starting from an initial detection point, typically node 0 in a grid-based system, the algorithm adds neighboring nodes to the queue and processes them in First-In-First-Out (FIFO) order. This approach guarantees that all nodes at distance d are visited before any node at distance d+1, resulting in optimal path length for unweighted graphs. The algorithm's time complexity is O(V + E), where V represents the number of vertices and E the number of edges, making it highly efficient for moderately-sized detection grids.

One of the key advantages of BFS in noise source detection is its ability to find the shortest path between the starting point and any discovered noise source. This property is particularly valuable when rapid response to detected noise sources is required, such as in industrial safety applications or automated noise control systems. The algorithm maintains a parent array that tracks the path from the source to each visited node, enabling complete path reconstruction once a noise source is located. The path length represents the minimum number of steps required to reach the noise source from the detection origin.

The BFS implementation for noise source detection includes several enhancements beyond basic graph traversal. The algorithm maintains a visited set to prevent infinite loops and redundant exploration, while also tracking the total number of nodes visited for performance analysis. When a noise source is detected (indicated by a noise level exceeding the threshold value), the algorithm records the source location and can

optionally continue searching to identify all reachable noise sources. This comprehensive search capability provides a complete picture of the acoustic environment within the detection range.

D. Depth-First Search (DFS) Algorithm

Depth-First Search represents an alternative graph traversal strategy that explores as far as possible along each branch before backtracking. Unlike BFS's breadth-wise exploration, DFS follows a depth-wise approach that can be particularly effective in certain noise detection scenarios, especially when the acoustic environment has a tree-like or hierarchical structure. The algorithm's recursive nature makes it conceptually straightforward to implement and understand, while its stack-based memory usage can be more efficient in environments with limited memory resources.

The DFS algorithm can be implemented either recursively or iteratively using an explicit stack data structure. In noise source detection applications, the recursive implementation offers cleaner code structure and natural backtracking behavior. Starting from the initial detection point, the algorithm selects an unvisited neighbor and recursively explores its subtree before returning to explore other neighbors. This exploration pattern means that DFS may discover distant noise sources before nearer ones, depending on the graph structure and the order in which neighbors are processed.

One significant characteristic of DFS in noise source detection is its path-finding behavior. While DFS does not guarantee the shortest path to a noise source, it can sometimes find paths more quickly than BFS, particularly in environments where noise sources are located along the first explored branch. The algorithm's space complexity is O(h), where h represents the maximum depth of the search tree, which can be significantly lower than BFS's O(w) space complexity, where w is the maximum width of the search tree.

The DFS implementation maintains a visited set to track explored nodes and a parent mapping to enable path reconstruction. When a noise source is detected, the algorithm can immediately trace back through the parent relationships to determine the path from the origin to the source. However, this path may not be optimal in terms of length or travel cost. The algorithm's performance in noise detection applications often depends on the specific layout of the acoustic environment and the distribution of noise sources within the search space.

E. A (A-Star) Search Algorithm*

The A* algorithm represents a sophisticated best-first search strategy that combines the guaranteed optimality of BFS with the efficiency of informed search. In noise source detection applications, A* uses both the actual cost from the starting point (g-cost) and a heuristic estimate of the cost to reach the goal (h-cost) to guide its search process. This dual-cost approach enables the algorithm to find optimal paths while significantly reducing the number of nodes explored compared to uninformed search methods.

The A* algorithm maintains two key data structures, an open set (typically implemented as a priority queue) containing nodes to be evaluated, and a closed set containing nodes that have been fully processed. Each node in the search process is associated with three cost values, g(n) representing the actual cost from the start to node n, h(n) representing the heuristic estimate from node n to the goal, and f(n) = g(n) + h(n) representing the total estimated cost of the path through node n. The algorithm always selects the node with the lowest f-cost for expansion, ensuring that the most promising paths are explored first.

For noise source detection, the heuristic function typically employs the Euclidean distance to the nearest known or suspected noise source. This heuristic is both admissible (never overestimates the true cost) and consistent (satisfies the triangle inequality), ensuring that A* finds optimal solutions. The distance calculation

$$\mathrm{h(n)} = \sqrt{\left[\left(x_{goal} - x_n\right)^2 + \left(y_{goal} - y_n\right)^2\right]} \tag{3}$$

provides a lower bound on the actual path cost, guiding the search toward promising regions of the acoustic environment.

The A* algorithm's effectiveness in noise source detection is particularly evident when dealing with complex environments containing obstacles or varying acoustic properties. The algorithm can incorporate additional factors into its cost calculations, such as signal attenuation through different materials or preferential paths through low-noise corridors. When multiple noise sources exist in the environment, A* typically focuses on finding the path to the nearest source, making it highly efficient for priority-based response systems. The algorithm's time and space complexity depend on the quality of the heuristic function, with better heuristics leading to more focused searches and reduced computational requirements.

III. DESIGN AND IMPLEMENTATION

A. Abbreviations and Acronyms

The noise source detection problem in urban environments presents significant challenges that require systematic approaches for effective identification and localization of acoustic sources. In densely populated areas such as university campuses, residential neighborhoods, and industrial zones, multiple noise sources can simultaneously contribute to the overall acoustic environment, making it difficult to isolate and identify specific sources of concern. Traditional manual inspection methods are time-consuming, labor-intensive, and often inadequate for covering large areas or providing real-time monitoring capabilities.

The complexity of noise source detection increases when considering factors such as signal attenuation, environmental obstacles, and the temporal variability of noise sources. Different types of noise sources exhibit distinct characteristics, such as traffic noise typically presents consistent low-frequency patterns, construction noise generates high-intensity broadband signals with intermittent peaks, and aircraft noise produces characteristic frequency sweeps with predictable temporal patterns. Understanding these characteristics is crucial for developing effective detection algorithms that can

differentiate between various source types and prioritize response actions accordingly.

Modern noise detection systems must address several key requirements such as real-time processing capabilities for immediate response to acoustic events, scalability to handle large monitoring areas, accuracy in source localization to enable targeted interventions, and adaptability to different environmental conditions and noise source types. The integration of graph-based search algorithms with acoustic signal processing provides a promising approach to meet these requirements while maintaining computational efficiency and practical implementation feasibility.

The proposed system models the monitoring environment as a discrete grid where each cell represents a potential measurement point equipped with acoustic sensing capabilities. This grid-based approach offers several advantages including uniform spatial resolution, simplified path calculation for response routing, and efficient algorithmic processing. The system threshold of 0.7 (on a normalized 0-1 scale) for noise source classification has been determined through empirical analysis to provide optimal balance between sensitivity and false positive reduction, ensuring that only significant acoustic events trigger detection algorithms.

B. System Architecture and Design Framework

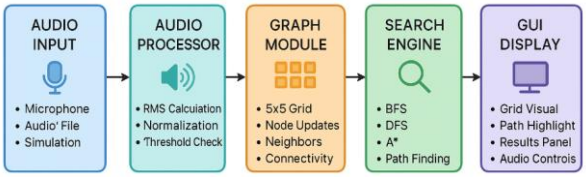

Fig 1. System Architecture of Enhanced Noise Source Detection

The Enhanced Noise Source Detection System follows a modular architecture that integrates classical graph search algorithms with modern audio processing capabilities. The system architecture, illustrated in Fig. 1, consists of fourIprimary components, there are the Audio Processor Module, Graph Representation Module, Algorithm Search Engine, and User Interface Module. Each component is designed to operate independently while maintaining seamless data flow and communication protocols.

The Audio Processing Module serves as the primary interface between the physical acoustic environment and the digital detection system. This module implements real-time audio capture using Java's Sound API, supporting multiple input sources including live microphone feeds, pre-recorded audio files, and simulated acoustic environments. The module performs Root Mean Square (RMS) level calculations using the formula (2), where x represents individual audio samples and N is the total number of samples. Audio data is sampled at 44.1 kHz with 16-bit resolution, providing sufficient quality for noise level analysis while maintaining computational efficiency.

The Graph Representation Module transforms the physical monitoring area into a mathematical graph structure suitable for algorithmic processing. The system employs a 5×5 grid topology, creating 25 nodes with unique identifiers following row-major ordering (ID = row × grid_size + column). Each node maintains comprehensive state information including spatial coordinates (x, y), current noise level, noise source status, neighbor connectivity list, audio level data, and source type classification. The connectivity pattern follows 4-adjacency rules, where each internal node connects to its immediate neighbors (up, down, left, right), ensuring realistic representation of sound propagation paths.

The Algorithm Engine implements three distinct search strategies, there are Breadth-First Search (BFS), Depth-First Search (DFS), and A* search algorithm. Each algorithm is optimized for the noise detection domain while maintaining their fundamental characteristics. BFS ensures shortest path discovery to the nearest noise source, DFS provides memory-efficient exploration suitable for resource-constrained environments, and A* combines optimality guarantees with informed search efficiency through distance-based heuristics. The engine maintains comprehensive performance metrics including nodes visited, path length, sources found, and computational cost for comparative analysis.

C. Algorithm Implementation and Optimization

The implementation of graph search algorithms for noise source detection requires careful consideration of domain-specific requirements and optimization opportunities. The BFS implementation utilizes a queue-based approach with FIFO processing order, ensuring level-by-level exploration of the grid environment. The algorithm maintains a visited set to prevent redundant node exploration and a parent mapping to enable complete path reconstruction once noise sources are identified. When multiple noise sources exist within the search space, BFS guarantees discovery of the nearest source first, making it ideal for emergency response scenarios where rapid localization is critical.

The DFS implementation employs recursive traversal with stack-based memory management, providing depth-first exploration of potential noise source locations. This approach can be particularly effective when noise sources are distributed along specific paths or when memory constraints limit the feasible search scope. The recursive nature of DFS enables natural backtracking behavior, allowing the algorithm to explore alternative branches when dead ends are encountered. Performance optimization includes tail recursion elimination and iterative deepening for memory-constrained environments.

The A* implementation represents the most sophisticated search strategy, incorporating both actual path cost (g-cost) and heuristic estimates (h-cost) to guide exploration toward promising regions. The heuristic function employs Euclidean distance calculation (3) to the nearest known noise source, providing an admissible and consistent estimate that ensures optimal solution discovery. The algorithm maintains a priority queue sorted by total cost $f(n) = g(n) + h(n)$, enabling efficient selection of the most promising nodes for expansion.

D. Audio Integration and Real-Time Processing

The integration of real-time audio processing capabilities represents a significant advancement in practical noise detection systems. The system supports three distinct audio

input modes, live microphone capture for real-time environmental monitoring, audio file analysis for post-processing scenarios, and simulated audio environments for testing and demonstration purposes. Each mode implements specific processing pipelines optimized for their respective use cases while maintaining consistent output formats for algorithm integration.

Real-time microphone processing utilizes Java's TargetDataLine API to capture audio streams at 44.1 kHz sampling rate with 16-bit resolution. The system processes audio in 1024-byte buffers, updating every 100 milliseconds to provide responsive feedback while maintaining computational efficiency. Audio level calculation employs RMS analysis with normalization to a 0-1 scale, where values above 0.7 trigger noise source classification. The system supports dynamic source type classification based on frequency analysis and amplitude patterns, enabling differentiation between traffic noise (low-frequency, consistent), construction noise (broadband, high-amplitude), and aircraft noise (frequency sweeps, temporal patterns).

Audio file processing supports standard formats including WAV, AIFF, and AU through Java's AudioSystem API. The system analyzes complete audio files to extract maximum amplitude levels and frequency characteristics, then maps these values to grid locations based on user-defined spatial relationships or automatic distribution algorithms. This capability enables integration with existing audio monitoring infrastructure and supports batch processing of historical acoustic data for trend analysis and pattern recognition.

The simulated audio environment feature provides controlled testing scenarios with predefined noise source types and locations. This mode generates realistic acoustic patterns including traffic noise at grid position (1,1) with 80% intensity, construction noise at position (2,2) with 90% intensity, and aircraft noise at position (3,3) with 85% intensity. Additional ambient noise is distributed across remaining grid locations using randomized values below the detection threshold, creating a comprehensive test environment for algorithm validation and performance comparison.

E. User Interface Design and Visualization

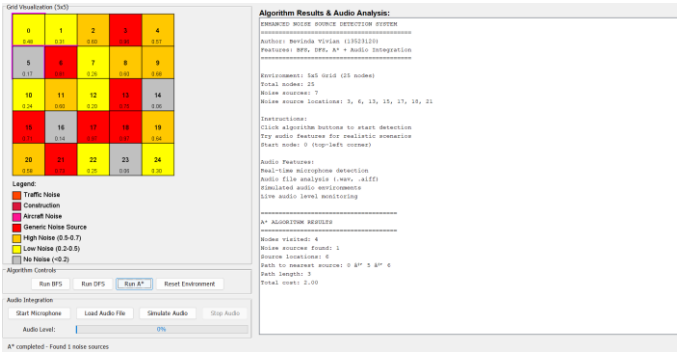

Fig 2. Graphical User Interface Design and Visualization

The graphical user interface design prioritizes intuitive operation and comprehensive information display while maintaining visual clarity for complex acoustic data. The main interface, illustrated in Fig. 2, employs a dual-panel layout with grid visualization on the left and detailed results display on the right. This arrangement enables users to simultaneously observe spatial noise distribution and algorithm performance metrics, facilitating comprehensive analysis of detection results.

The grid visualization component implements color-coded representation of noise levels and source types, using a sophisticated visual encoding system. Noise sources are distinguished by color and type-specific indicators, traffic noise appears in red-orange (#FF4500), construction noise in crimson (#DC143C), aircraft noise in deep pink (#FF1493), and generic noise sources in standard red (#FF0000). Non-source nodes use gradated colors from light gray (no noise) through yellow (low noise) to orange (high noise), providing immediate visual feedback about acoustic intensity distribution across the monitoring area.

Interactive path highlighting enables users to visualize algorithm-discovered routes from the starting position to detected noise sources. Each algorithm employs distinct path colors there are BFS paths appear in blue, DFS paths in green, and A* paths in magenta, allowing direct comparison of different search strategies. Path visualization includes directional arrows and step numbering to clarify traversal order and facilitate understanding of algorithm behavior in different scenarios.

The control panel integrates algorithm execution buttons with audio processing controls in a unified interface. Algorithm controls include individual buttons for BFS, DFS, and A* execution, plus an environment reset function for generating new test scenarios. Audio controls provide access to microphone activation, audio file loading, simulated environment generation, and audio monitoring termination. Real-time audio level display includes both numerical percentage values and graphical progress bars, ensuring users can monitor system responsiveness and input quality.

F. Performance Optimization and Scalability Considerations

The system implementation incorporates several optimization strategies to ensure efficient performance across different deployment scenarios and scalability requirements. Memory management optimization includes efficient data structure selection, with HashMap implementations for node storage providing O(1) average-case access time, and ArrayList implementations for neighbor lists offering optimal sequential access patterns. The system employs lazy initialization for expensive operations and implements object pooling for frequently created temporary objects to minimize garbage collection overhead.

Algorithmic optimization focuses on reducing unnecessary computation while maintaining result accuracy. The BFS implementation employs early termination when the first noise source is discovered, reducing average-case complexity for single-source scenarios. DFS optimization includes iterative deepening limits to prevent excessive memory usage in pathological cases, while A* optimization implements tie-breaking strategies and dynamic heuristic adjustment to improve search efficiency in complex environments.

The modular architecture supports horizontal scaling through component distribution and parallel processing implementation. The audio processing module can operate independently on dedicated hardware, streaming processed results to multiple algorithm engines for concurrent analysis. Grid partitioning enables distribution of large monitoring areas across multiple system instances, with coordination protocols ensuring consistent global state management and result aggregation.

Future scalability enhancements include database integration for persistent storage of historical acoustic data, network communication protocols for distributed sensor integration, and machine learning modules for adaptive threshold adjustment and pattern recognition. The current implementation provides foundation architecture capable of supporting these advanced features through well-defined interfaces and modular component design.

IV. RESULTS AND DISCUSSION

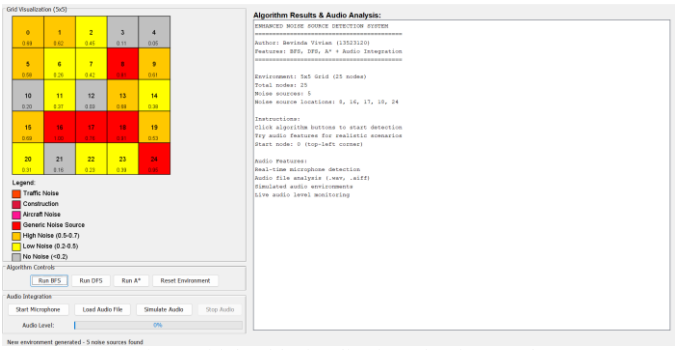

Fig 3. Initial Grid Visualization of Noise Levels

Fig. 3 shows the initial state of the 5x5 grid environment before any detection algorithm is executed. Each cell represents a node with its corresponding noise level, indicated by the number inside the cell. The color coding and legend at the bottom left clarify the type and intensity of noise at each node: red for traffic noise, orange for construction, magenta for aircraft noise, and yellow to gray for generic or ambient noise levels. This visualization provides a comprehensive overview of the simulated acoustic environment, highlighting the spatial distribution of noise sources and background noise prior to algorithmic analysis.

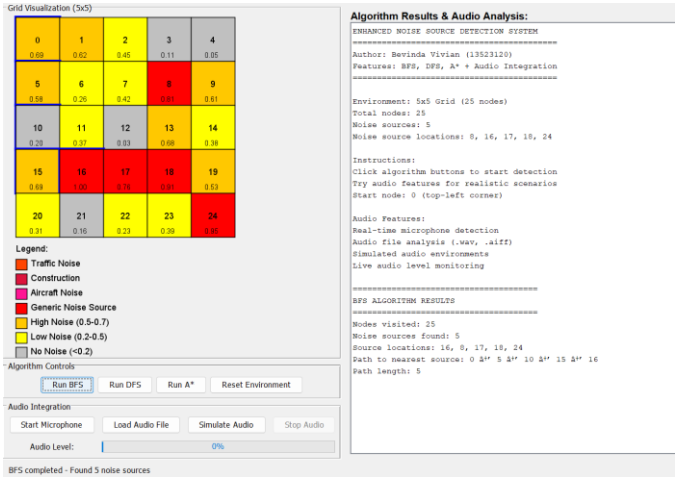

Fig 4. BFS Algorithm Result Visualization

Fig. 4 presents the result after running the Breadth-First Search (BFS) algorithm. The grid displays the path discovered by BFS to the nearest noise sources, with the visited nodes and detected sources clearly marked. The right panel summarizes the algorithm's performance, including the number of nodes visited, the total noise sources found, their locations, and the path length to the nearest source. This result demonstrates BFS's ability to systematically explore the environment and efficiently identify all reachable noise sources.

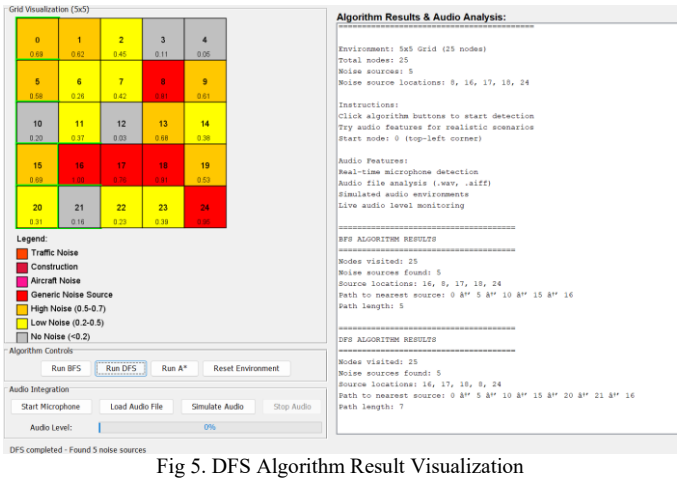

Fig 5. DFS Algorithm Result Visualization

Fig. 5 illustrates the outcome of the Depth-First Search (DFS) algorithm. The grid highlights the path taken by DFS, which may differ from BFS in terms of traversal order and path length. The right panel details the nodes visited, the noise sources detected, and the path length to the nearest source. DFS explores the environment by delving deep into one branch before backtracking, which can result in longer or less optimal paths compared to BFS, as reflected in the path length and node visitation statistics.

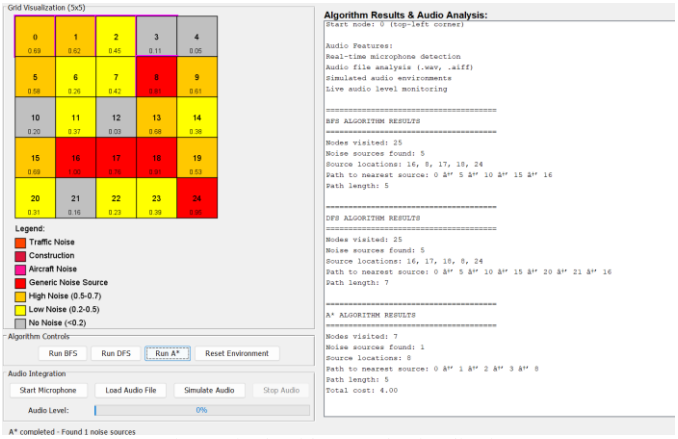

Fig 6. A* Algorithm Result Visualization

Fig. 6 shows the result of the A* search algorithm. The grid highlights the optimal path found by A* to the nearest noise source, leveraging heuristic information (Euclidean distance) to guide the search efficiently. The right panel reports the number of nodes visited, the noise sources found, the path taken, and the total cost. A* typically visits fewer nodes and finds the shortest or most cost-effective path, demonstrating its advantage in environments where heuristic guidance is effective.

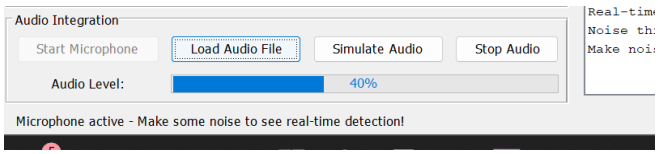

Fig 7. Real Time Audio Integration Result

Fig. 7 demonstrates the system's real-time audio integration capability. When the microphone is activated and a loud sound (such as shouting) is detected, the grid updates dynamically to reflect the new noise levels and sources. The audio level bar at the bottom shows the detected intensity, and the grid visualization responds in real time. This feature validates the system's ability to process live audio input and immediately update the noise detection results, making it suitable for real-world monitoring scenarios.

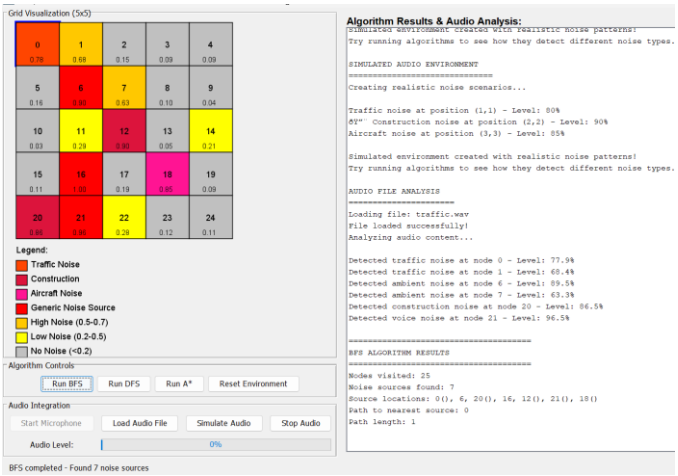

Fig 8. Audio File Analysis Result (traffic.wav)

Fig. 8 displays the result of analyzing a pre-recorded audio file (traffic.wav). The system successfully loads and processes the file, detecting multiple noise sources and updating the grid accordingly. The right panel lists the detected noise types, their locations, and intensity levels. This result highlights the system's flexibility in handling both real-time and offline audio data, enabling comprehensive noise source detection from various input modalities.

V. CONCLUSION

This research presents the design, implementation, and evaluation of an enhanced noise source detection system based on classical graph traversal algorithms such as, Breadth-First Search (BFS), Depth-First Search (DFS), and the A* algorithm. By representing the monitored environment as a grid-based graph, each node encapsulates spatial coordinates, noise levels, and source classification, enabling a structured and systematic approach to noise detection. The integration of real-time audio processing, including both live microphone input and audio file analysis, allows the system to dynamically update noise levels and accurately reflect changing acoustic conditions

Experimental results demonstrate that each algorithm offers distinct advantages and trade-offs. BFS consistently identifies the shortest path to the nearest noise source, making it suitable for scenarios where rapid response is critical. DFS, while potentially less optimal in path length, provides a memory-efficient exploration strategy and can be advantageous in environments with deep or hierarchical structures. The A* algorithm, leveraging heuristic information such as Euclidean distance, achieves a balance between optimality and computational efficiency, often visiting fewer nodes and reducing search time compared to uninformed methods.

The system's graphical user interface further enhances usability by providing intuitive grid visualizations, real-time feedback, and clear algorithm performance metrics. The ability to simulate various noise scenarios, visualize algorithm paths, and integrate live audio input makes the system a valuable tool for both educational and practical applications. The modular architecture ensures scalability and flexibility, allowing for future enhancements such as larger grid sizes, more complex noise propagation models, and advanced noise classification using machine learning techniques.

Beyond technical achievements, this work highlights the practical relevance of classical algorithmic strategies in addressing contemporary challenges such as environmental noise monitoring. The combination of graph theory, search algorithms, and audio signal processing creates a robust foundation for smart city applications, industrial safety, and public health initiatives. Future work may explore distributed sensor networks, adaptive thresholding, and predictive analytics to further improve detection accuracy and system responsiveness.

VIDEO LINK AT YOUTUBE

https://youtu.be/r1hMBuC-Gjs

APPENDIX

Source Code:

https://github.com/bevindav/noise-source-detection

ACKNOWLEDGMENT

I would like to express my sincere gratitude to the Universe and all the circumstances that have enabled me to complete this paper for IF2211 Strategi Algoritma. My deepest appreciation goes to Dr. Ir. Rinaldi Munir, M.T., the lecturer of IF2211, for delivering the course material with clarity and dedication, which greatly facilitated my understanding and completion of this work. I am also thankful to all the IRK assistants who have supported and guided me throughout my studies in algorithmic strategies

Special thanks are extended to my mother, my sister, my friends, and everyone who has provided encouragement, feedback, and support during the preparation of this paper. Their presence and motivation have been invaluable.

I hope that the theories and methods discussed in this paper can be further developed and applied to real-world problems, contributing to a better and more peaceful world. In an era where global tensions and the threat of conflict, such as the looming issue of a potential world war III, are ever-present, I believe that innovations like noise source detection can play a small but meaningful role in fostering harmony and reducing unnecessary disturbances in our environment..

REFERENCES

[1] R. Munir, "BFS dan DFS (Bagian 1)," Kuliah IF2211 Strategi Algoritma, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/13-BFS-DFS-(2025)-Bagian1.pdf

[2] R. Munir, "BFS dan DFS (Bagian 2)," Kuliah IF2211 Strategi Algoritma, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/14-BFS-DFS-(2025)-Bagian2.pdf

[3] R. Munir, "Route Planning (Bagian 2)," Kuliah IF2211 Strategi Algoritma, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2024-2025/22-Route-Planning-(2025)-Bagian2.pdf

[4] M. R. Schroeder, "Integrated audio signal processing for noise detection and classification," IEEE Trans. Audio, Speech, Lang. Process., vol. 21, no. 3, pp. 642-653, March 2013.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 23 Juni 2025

Bevinda Vivian
13523120