A Novel Approach to Image Retrieval for Vision-Based Positioning Utilizing Graph Topology

Abdelgwad Elashry [1], Charles Toth [2]

[1] Computer Engineering, Elsewedy University of Technology, Cairo, Egypt. (abdelgwad.elashry@sut.edu.eg)

[2] Dept. of Civil, Environmental and Geodetic Engineering, The Ohio State University, Columbus, OH, USA (elashry.1, toth.2)@osu.edu

KEYWORDS: Vision-based positioning, Image retrieval, Graph topology, The Bag of Visual Words, Fingerprinting.

ABSTRACT:

This research introduces a novel approach to improve vision-based positioning in the absence of GNSS signals. Specifically, we address the challenge posed by obstacles that alter image information or features, making retrieving the query image from the database difficult. While the Bag of Visual Words (BoVW) is a widely used image retrieval technique, it has a limitation in representing each image with a single histogram vector or vocabulary of visual words, i.e., the emergence of obstacles can introduce new features to the query image, resulting in different visual words. Our study overcomes this limitation by clustering the features of each image using the k-means method and generating a graph for each class. Each node or key point in the graph obtains additional information from its direct neighbors using functions employed in graph neural networks, functioning as a feedforward network with constant parameters. This process generates new embedding nodes, and eventually, global pooling is applied to produce one vector for each graph, representing each image with graph vectors based on objects or feature classes. As a result, each image is represented with graph vectors based on objects or feature classes. In the presence of obstacles covering one or more graphs, there is sufficient information from the query image to retrieve the most relevant image from the database. Our approach was applied to indoor positioning applications, with the database collected in Bolz Hall at The Ohio State University. Traditional BoVW techniques struggle to properly retrieve most query images from the database due to obstacles like humans or recently deployed objects that alter image features. In contrast, our approach has shown progress in image retrieval by representing each image with multiple graph vectors, depending on the number of objects in the image. This helps prevent or mitigate changes in image features caused by obstacles covering or adding features to the image, as demonstrated in the results.

1. INTRODUCTION

In recent decades, cameras have achieved widespread acceptance and remarkable advancements across various computer vision and photogrammetry applications, including obstacle avoidance, image recognition, tracking, SLAM, and image-based localization. This is particularly notable in GPS/GNSS-denied environments [1]. Researchers have endeavoured to address this challenge by exploring sensors such as cameras [2], LiDAR [3], radar [4], and radio signals. Among these sensors, the camera stands out as the most extensively utilized due to its compatibility with a wide range of smart devices, robots, cars, etc., and its affordability. The domain of indoor positioning, which has garnered increased attention recently, presents a significant challenge due to weak or absent GNSS signals and the diverse infrastructure within indoor environments. Numerous radio frequency (RF)-based localization approaches have been proposed as solutions for indoor positioning systems (IPS) [5]. These approaches can be categorized into Time of Flight (TOF) [6], encompassing time of arrival and time difference of arrival, as well as angular positioning (AoA, Angle of Arrival). Then using range and/or angle measurements between the access points, the user's position is computed by trilateration and triangulation. Unfortunately, these methods encounter limitations such as non-line-of-sight (NLoS) and signal multipath effects [7]. The fingerprinting technique [7] based on Received Signal Strength (RSS) is a widely adopted approach, where the location is determined by comparing received signal strength data with a predefined calibration point database (radio map). Fingerprinting is advantageous for its independence from time synchronization and lack of requirement for line of sight. However, it necessitates a training phase to create the radio map. The Light Detection and Ranging sensor (LiDAR) [3] is crucial data acquisition technology for topographic and infrastructure mapping, and more recently, in autonomous driving. While LiDAR requires accurate georeferencing based on integrating Inertial Navigation Systems (INS) with GPS/GNSS in outdoor scenarios, its direct applicability to indoor navigation is limited, though with the introduction of efficient LiDAR SLAM methods recently, it is changing. When combined with other sensors, LiDAR can be effective but suffers from size, computational cost, and power consumption compared to cameras. Cameras offer diverse solutions for GNSS-denied environments [2], making them suitable for various devices due to their compact size, lightweight, low power consumption, and ability to provide rich information such as colors and features. The visual fingerprinting technique, especially using the Bag of Visual Words (BoVW) [8], is commonly employed in indoor environments to determine user positions by comparing query images with predefined geotagged database images. However, a limitation of BoVW arises when obstacles in the image cover or alter features, leading to changes in visual words, especially those associated with rare occurrences in the database. This poses a challenge in accurately retrieving the query image from the database. To address this limitation, we propose a new solution that represents the image using a set of graphs. By employing the k-means clustering technique on image features and constructing object graph vectors based on Delaunay triangulation [10], spatial relationships between nodes (key points) are more effectively captured. This results in the extraction of graph information for each object, utilizing knowledge about the direct neighbors of each node to build adjacency matrices representing each graph [16]. Consequently, each image is represented by different vectors based on the objects within, and these vectors are stored in the database. Global pooling is then applied to each graph in the image to obtain a unique vector representing the entire graph. Despite the presence of obstacles that may alter graph vector information by covering or adding features to the original image,

https://doi.org/10.5194/isprs-annals-X-2-2024-49-2024

information from other graph vectors related to the query image can still be leveraged for retrieval based on a similarity approach.

2. THE BAG OF VISUAL WORDS (BOVW)

The Bag of Visual Words (BoVW) [11] is a method employed to identify similar images in a database when provided with a query image. This technique is widely used in computer vision, photogrammetry, and robotics, particularly in tasks involving place or location recognition, where the objective is to find one or more images of the same location. A familiar example is the Google image search, allowing users to drag and drop an image into the search window, yielding similar images in return. The underlying system stores a concise representation of reference images in the database, often in the form of a histogram capturing specific feature occurrences. Subsequently, all comparisons are conducted based on these histograms. BoVW [9] characterizes images by tallying the occurrences of individual visual words in an image. It's important to note that popular feature extractors like SIFT and SURF are utilized for extracting key points and computing their feature descriptors. Figure 1 [11] illustrates different visual words marked with colored shapes, where each shape denotes the occurrence of a specific visual word. While this depiction is a simplified illustration showcasing a few visual words, in reality, each image is distilled to visual words, rendering the actual pixel values inconsequential. Within the bag of words framework, each image transforms into a histogram by merely counting the occurrences of visual words within that image. The x-axis of the histogram represents the visual words, while the y-axis indicates how frequently individual visual words appear in the image. A visual word, in this context, refers to a feature descriptor computed from the mean of several similar descriptors grouped using a clustering algorithm like k-means [12].

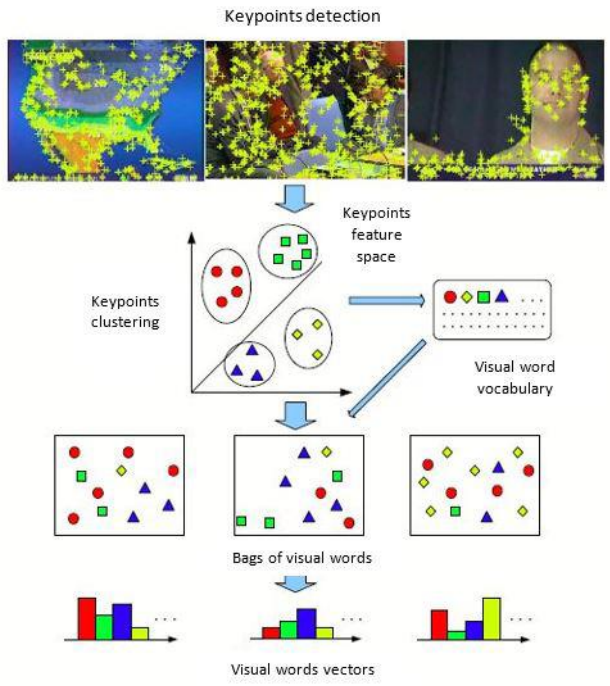


Figure 1: Bag of visual words representation

Each image within the database undergoes transformation into a histogram, where the x-axis is defined by visual words, and the y-axis signifies their frequency of appearance in the image. Consequently, instead of storing M images in the database, M histograms are stored. Certain visual words prove more apt for conducting comparisons, while others may lack expressiveness, such as a word occurring in every image, offering limited support for meaningful comparisons. Therefore, to prioritize important visual words over others based on anticipated information, a weight is computed for each visual word. This weighting process is accomplished using Term-Frequency Inverse-Document Frequency (TF-IDF) [15], as illustrated in Equation 1.

$$t_{id} = \frac{n_{id}}{n_d} log \frac{N}{n_i} \quad (1)$$

t_{id}: Histogram bin of word *i* for image *d*.
n_{id}: Occurrences of word *i* in image *d*.
n_d: Number of word occurrences in image *d*.
n_i: Number of images that contain word *i*.
N: Number of images.

TF-IDF serves to diminish the significance of words appearing in every image, reducing their weight to zero, while assigning higher weights to infrequently occurring words. TF-IDF computations are then applied to all histograms within both the database and the query image. Typically, these histograms are compared using cosine distance, a metric ranging from 0 to 1. Subsequently, the cost matrix derived from these histogram comparisons reflects a distance of 0 when a histogram is compared with itself, whereas all other comparisons yield higher values. Given a user-captured query image, the algorithm retrieves the *N*=20 most similar images from the database, prioritizing those with the smallest cosine distances in the cost matrix.

3. INTRODUCTION TO GRAPHS AND NEW TRENDS

Graph theory, pioneered in the 18th century by the Swiss mathematician Leonhard Euler, provides a mathematical framework for modeling pairwise relationships between objects. It serves as a valuable tool in addressing real-world problems, including applications such as Google Maps for finding the shortest path to home, and across a broad spectrum of disciplines. The fundamental components of any network or graph are its nodes, representing elements, and edges, representing the relationships between nodes. In our work, a key focus after extracting image features is the examination of geometric relations among key points. These key points are treated as nodes, forming graphs where their descriptors serve as the features of these nodes. Geometric graphs, specifically those in the 2D Euclidean space, are employed in this context, aligning with the image domain. Given the similarity in highly overlapping images, the graphs defined by key points should exhibit topological similarity, which we can be exploited. To address this, we adopt a traditional graph representation with functions inspired by graph neural networks, effectively operating as a feedforward neural network. This system aggregates information from the nodes' direct neighbors and updates itself through message-passing layers, including the graph attention network and graph convolution network. In recent years, there has been a substantial surge in interest in applying deep learning to graphs, often referred to as graph neural networks (GNN) or geometric deep learning [13]. This approach has seen significant research in graph representation learning, becoming one of the fastest-growing areas in machine learning. The integration of graph theory with deep learning has gained traction across various fields, including social networks (e.g., Facebook), recommendation systems, medicine (disease classification), pharmacy (learning molecular fingerprints), and intelligent transportation [14]. Despite this growth in GNN, achieving optimal advancements in graph networks proves challenging

with traditional machine learning methods. This challenge arises because the graph structure is generic and cannot be easily represented in a sequence or a grid. Modern deep learning tools like convolution filters and pooling, designed for simple sequences and grids, are less effective when dealing with networks or graphs possessing arbitrary sizes, complex topological structures, and no fixed node ordering or reference point, as illustrated in Figure 2. Graph neural networks emerge as valuable tools in addressing non-Euclidean data structures within machine learning in the graph domain [17]. They overcome the limitations of traditional machine learning in non-Euclidean domains, such as issues with pooling operators and localized convolution filters, as demonstrated in Figure 3. Consequently, graph theory-based deep learning has garnered significant attention in contemporary research.

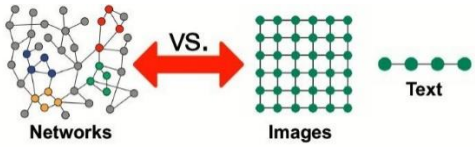


Figure 2: Network structure vs image and text structures

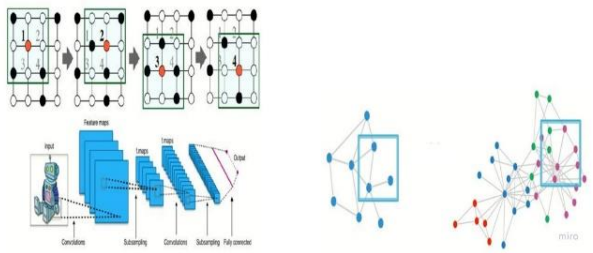


Figure 3: Left figure shows an image in regular grid format, and the right figure shows graphs with arbitrary size and complex topological structure.

Various operations or tasks, as depicted in Figure 4, can be executed on graphs. Firstly, there is the option of node-level predictions or node classification, involving the prediction of attributes for unlabeled nodes and their subsequent classification. In this scenario, the Graph Neural Network (GNN) utilizes information from other nodes in the graph to deduce attributes for these unlabeled nodes. Secondly, another feasible task is link prediction or edge-level prediction, where the aim is to predict connections between two nodes within the graph. Lastly, the entire graph can serve as input for predicting a specific attribute or class label associated with graphs in a dataset. Examples of applications include predicting molecule properties, analyzing social networks, addressing cyber-security challenges, and optimizing GPS/Google Maps for finding the shortest path to a destination. Additionally, graph matching can be applied to support collaborative navigation, particularly in PNT (Positioning, Navigation, and Timing) anomaly detection, as an effective tool for identifying discrepancies in corrupted local positions.

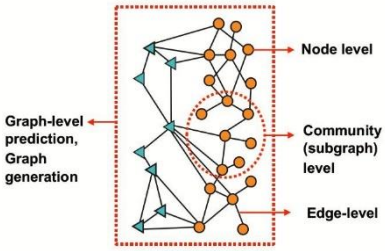


Figure 4: Different GNN task operations

The core concept behind Graph Neural Networks (GNN) is to train neural networks to effectively represent graph data, a process referred to as representation learning. Leveraging all available information about the graph, which includes both node features and connections stored in the adjacency matrix, the GNN generates new representations, also known as embedding nodes, as illustrated in Figure 5. These embedding nodes contain information from other nodes in the graph, and this embedding can be utilized for making predictions. When nodes possess similar features, GNN ensures that their respective embeddings are also similar, consequently leading to comparable node representations. Similarly, for entire graphs, GNN produces similar graph embeddings when dealing with graphs that share common features. The core building blocks of graph neural networks are the message-passing layers, which play a pivotal role in combining node and edge information into the node embedding.

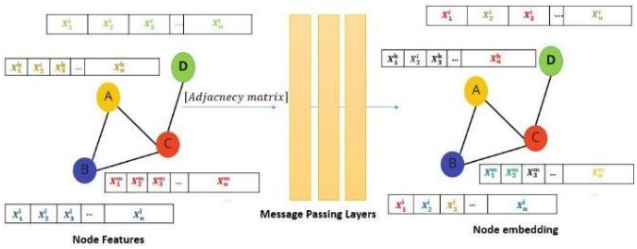


Figure 5: GNN structure

The basic idea of GNNs is to learn the embedding nodes by iteratively combining the node information in a local neighborhood; in other words, the nodes learn something about the direct neighbors, then the neighbors’ neighbors, and so on. The message-passing layers consist of update and aggregation functions as shown in Equation 2:

$$h_u^{(k+1)} = UPDATE^{(k)}\left(h_u^{(k)}, \text{AGGREGATE } E^{(k)}\left(\{h_v^{(k)}, \forall v \in \mathcal{N}(u)\}\right)\right) \quad (2)$$

Where $h_u^{(k)}$ is the current node features, and $h_u^{(k+1)}$is the node embedding, and $\mathcal{N}(u)$ is the direct neighbors. The aggregation function uses the information of the direct neighbors of a node u and aggregates them in a specific way. Then, it updates the current state in step k and combines them with the aggregated neighbor states, as shown in Figure 6; the orange node will get information from its direct neighbors (blue nodes) in the first layer and then update itself; in the second layer, it can get information from the neighbors’ neighbors (green node).

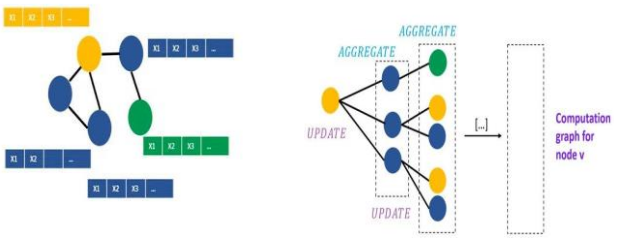


Figure 6: Aggregation and Updating functions.

In preceding research, various researchers have devised different methods for aggregation and updating functions within message layers. In our study, we employed the feedforward neural network concept, utilizing the idea of having these functions without learning parameters to obtain node embeddings. The message-passing layer constitutes a crucial component of the Graph Neural Network (GNN) layer for model construction. Numerous researchers have explored different methodologies for message-passing layers, varying in their approaches to

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-2024-49-2024 | © Author(s) 2024. CC BY 4.0 License.

neighborhood aggregation functions and update functions, which aggregate neighbors' information through methods such as averaging or maximizing. Some studies have incorporated neural networks and recurrent neural networks for acquiring neighbors' information and updating nodes [17]. The fundamental approach is illustrated in Equation 3, which involves averaging neighbor messages, updating the node itself, and subsequently applying the neural network.

$$\begin{gathered} \mathrm{h}_v^0 = X_v \\ \mathrm{h}_v^{(l+1)} = \sigma\left(\mathrm{W}_l \sum_{u \in \mathrm{N}(v)} \frac{\mathrm{h}_u^{(l)}}{|\mathrm{N}(v)|} + \mathrm{B}_l \mathrm{h}_v^{(l)}\right), \forall l \in \{0, \ldots, L-1\} \quad (3) \\ \mathrm{Z}_v = h_v^L \end{gathered}$$

where $\mathrm{h}_v^0 = X_v$ is the initial 0_{th} layer represented by node features, h_v^L is the node embedding at L layers of neighborhood aggregation, L is the total number of layers, and σ is a nonlinear function, such as RELU. W_l and B_l are learnable parameters. $\sum_{u \in \mathrm{N}(v)} \frac{\mathrm{h}_u^{(l)}}{|\mathrm{N}(v)|}$ is the average of the neighbor’s previous layer embedding. Z_v is the final node embedding which can be fed into any loss function and then running stochastic gradient descent (SGD) to train the weight parameters. Figure 7 shows how one node can learn from its neighbors based on aggregation and update functions. In the first GNN layer (message passing layer), the node gets information from the direct neighbors, and in the second GNN layer, it learns from the indirect neighbors.

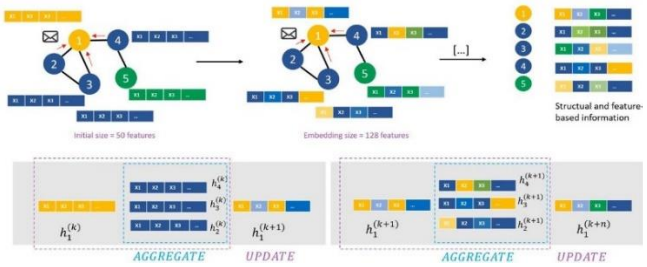


Figure 7: Aggregation and updating processing.

4. PROPOSED ALGORITHMS

The specific research objective is to address the issue of obstacles present in images, which can alter visual words and diminish the effectiveness of Bag of Visual Words (BoVW) methods in the search process. To tackle this problem and enhance applicability to image recognition applications for querying databases, an algorithm inspired by graph topology, akin to Graph Neural Networks (GNN), is proposed. Numerous applications, including recognition, vision-based positioning, visual SLAM, and other vision-related tasks, rely on querying images matched against databases to determine the user's location. The proposed model encodes both the database and query images to generate image vectors. Figure 8 illustrates the workflow for obtaining the encoded image fingerprint, starting with the application of the k-means algorithm to the image's key points to exclude those distant from each centroid. Subsequently, a Delaunay triangulation is applied to the clustered key points, providing information about the key points (nodes), such as their neighbors and edges. This information is then used to construct an adjacency matrix for creating image graphs. Each image's graphs serve as input to the aggregation and updating functions, employed in graph convolutional networks with consistent parameters. This process generates a new embedding feature by sharing information between nodes' direct local neighbors through the initial message-passing layer, iteratively learning from neighbors of neighbors. Two message-passing layers are utilized to generate distinct node embedding features. Finally, a global graph pooling algorithm is applied to each graph, yielding a single vector to represent each graph within the image. Figure 9 shows the process of representing the image by a set of graphs, describing the geometric relationships of features which should be similar in both highly overlapping images and graph topologies because of the similarity in the 2D keypoint locations.

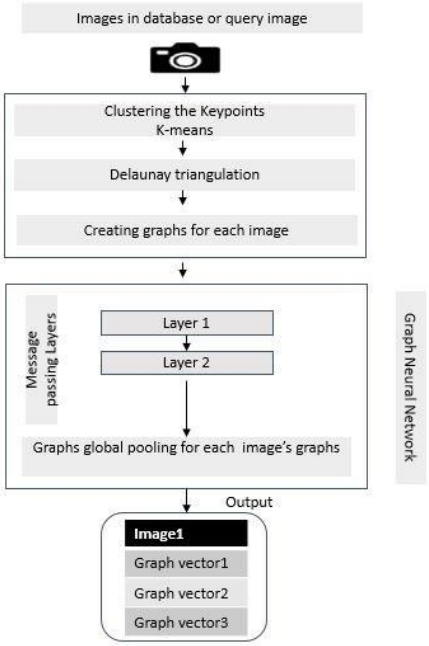
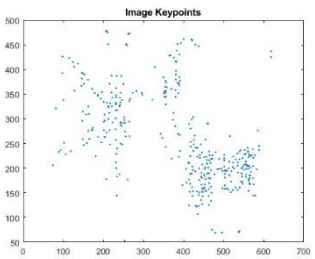
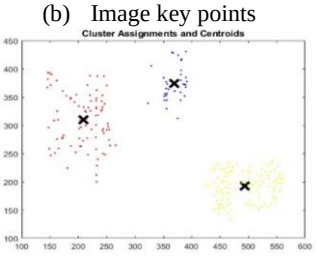


Figure 8: Workflow for getting encoding image fingerprint.

(a) Original image

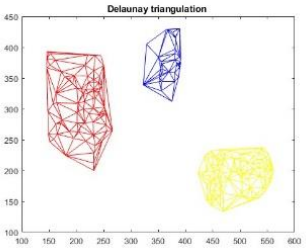


(b) Image key points

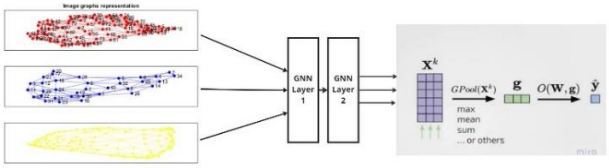


(c) K-means clustering algorithm.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-2024-49-2024 | © Author(s) 2024. CC BY 4.0 License.



(d) Delaunay triangulation



(e) Image graph representation and global pooling.

Figure 9: The process of representing the image by a set of graphs depending on the features’ geometric relationship.

The proposed matching algorithm relies on k-dimensional trees, utilizing cosine similarity approaches to calculate the distance or similarity between two fingerprints. In our study, the feature size is 64 as we employ SURF descriptors as a vector representation for classification through the k-dimensional trees algorithm. In the context of Bag of Visual Words (BoVW), the cosine similarity approach is adopted, given the large magnitude of vectors, and the visual word vocabulary size of 20,000. For vision-based indoor positioning, our proposed algorithm follows a workflow akin to the visual fingerprinting (VF) algorithm, representing the image with a visual word vocabulary in BoVW. In instances where an obstacle obscures image features, BoVW may struggle to find a match for the query image. The algorithm we propose addresses this limitation by representing the image with multiple graph embedding vectors. Even when an obstacle covers a portion of the image, we may retain information about the query image from other vectors, facilitating similarity algorithms in identifying the closest image from the alternate graph vectors. The workflow of the proposed algorithm is illustrated in Figure 10.

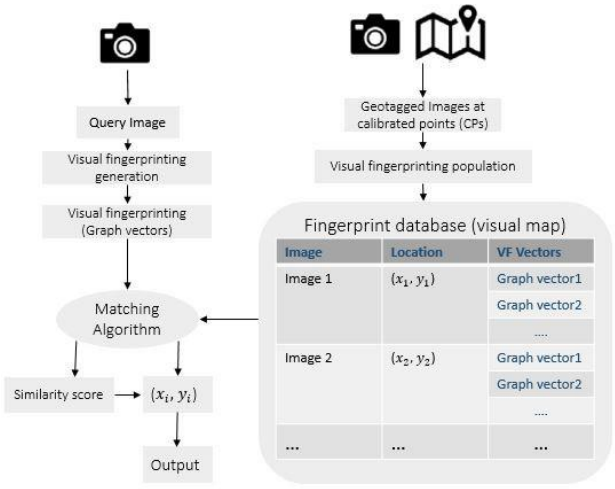


Figure 10: Workflow of the proposed algorithm.

5. EXPERIMENTAL RESULTS

The experiments utilized three distinct datasets gathered within Bolz Hall at the Ohio State University. Data collection occurred on the 2nd, 3rd, and 4th floors employing the LooMo robot from the SPIN Lab. The robot was equipped with a GoPro HERO5 camera mounted on its top, capturing training datasets from the center of the corridor, as depicted in Figure 11. To generalize the problem across the entire building, all datasets from each floor were combined. This approach ensures that the robot can recognize its position in any location of interest within the real environment.

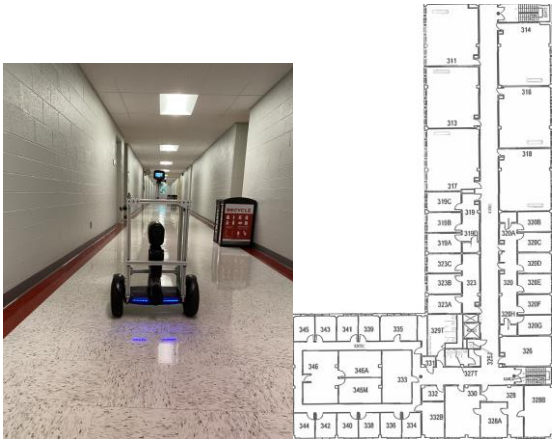


Figure 11: LooMo robot with the GoPro HERO5 camera.

While these datasets were acquired in the same building, they are different in terms of the number of objects on each floor that impacts the number of image features. For example, the first dataset on the 2nd floor has less features because it has fewer objects in each image, as shown in Figures 12 and 13.

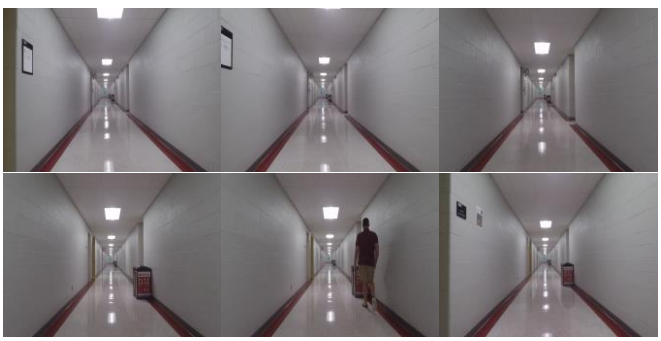


Figure 12: Image samples from the 2nd floor.

In contrast to the 2nd floor, the dataset of the 3rd has more objects, as shown in Figure 13.

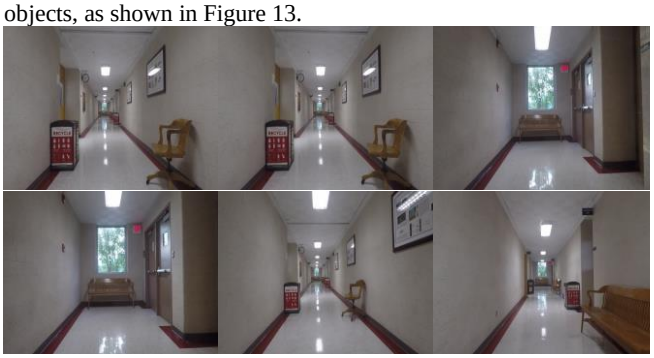


Figure 13: Image samples from the 3rd floor

The 4th floor shows a more typical pattern with many different objects, as shown in Figure 14.

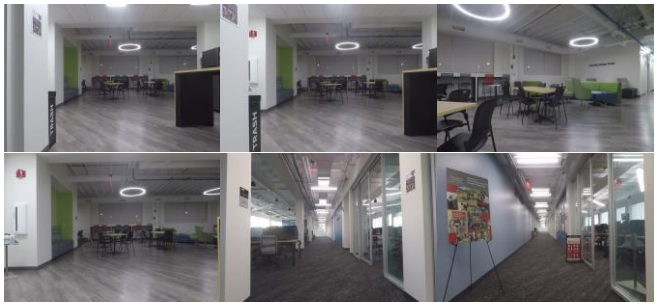


Figure 14: Image samples from the 4th floor.

The process of visual fingerprinting is divided into two primary phases, illustrated in Figure 15. Initially, during the training phase, images are captured at predetermined calibration points and processed through our algorithm to produce image graph vectors, subsequently populating the database. In the test phase, a mobile user captures an image at an unknown position, and its graph vectors are calculated. These vectors are then compared to the image graph vector entries in the database using the previously described similarity approach. Ultimately, the user's position can be determined by associating it with the closest pre-defined location of an image in the database.

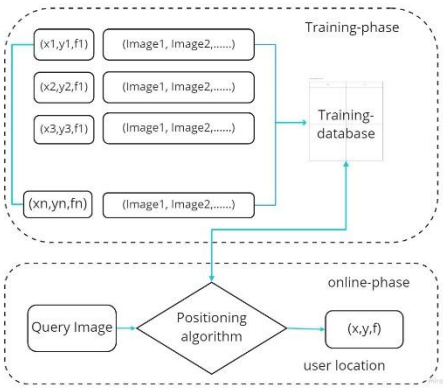


Figure 15: Workflow of the fingerprinting approach.

The corridor area in the building is (38m x 3m) and images were recorded continuously. For the visual fingerprinting, the corridors were divided into equal spaces, and the images were collected in both directions at accurately surveyed locations which were distributed evenly on each floor. The total number of database images that were collected in the experiment is 1,220 for the whole building.

Figure 16: Corridor layout, cells, and calibration points

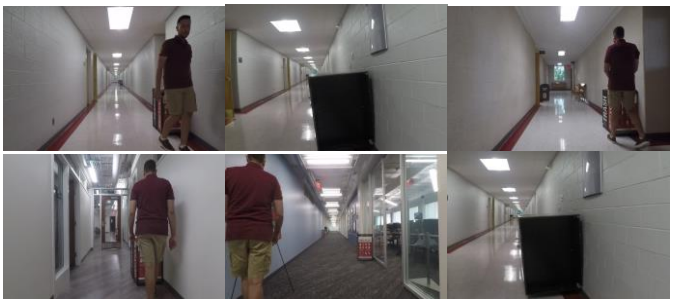


Figure 17: Samples of test images with obstacles.

In addition, test images were collected at different locations to add some obstacles to the images and thus to present the real world as shown in Figures 17 and 18, illustrating the effect of the obstacle features on the original image.

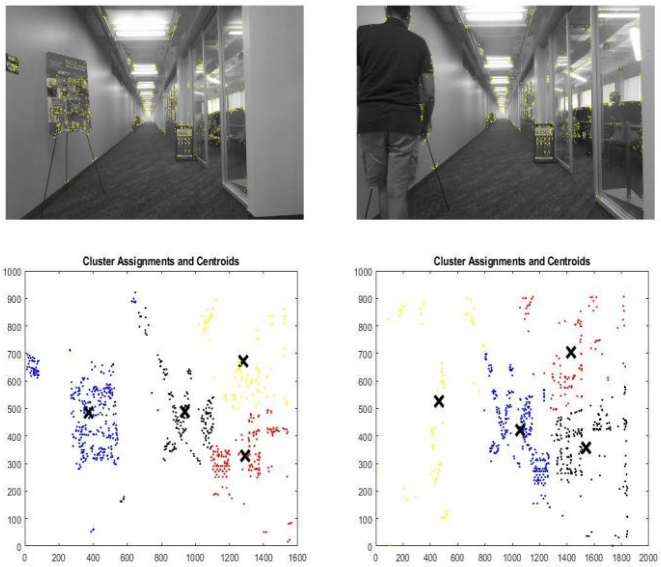


Figure 18: The effect of obstacle to the features of the original image

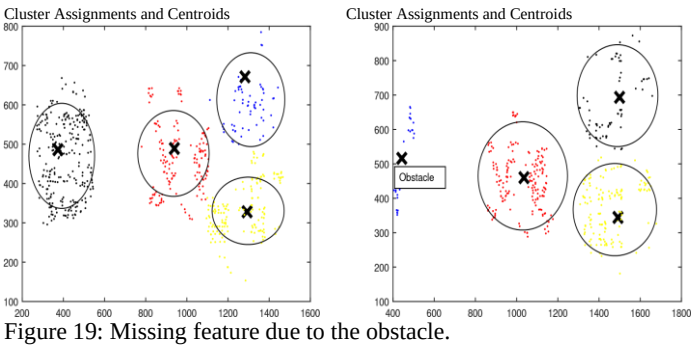


Figure 19: Missing feature due to the obstacle.

To construct the BoVW database, a total of 494,473 features are extracted from 1,220 training images. Subsequently, 80 percent of the strongest features from each category are retained. Utilizing k-means clustering, we create visual vocabulary words by determining the mean of each group from the entire database. Each image is then represented based on the length of the vocabulary words, depending on the words present in the image. As a result of the k-means output, each image is represented by a length of 20,000 visual words. As illustrated in Figure 19, numerous features are eliminated from the original image features, and obstacle features are introduced. Despite this, our solution generally retains sufficient information from the image

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-2024-49-2024 | © Author(s) 2024. CC BY 4.0 License.

for a successful match. The query image can be retrieved from the database based on strong vectors with high similarity scores. In contrast, as demonstrated in Figure 20, the Bag of Visual Words fails to retrieve the nearest image to the query image in most of the test samples due to alterations in image features.

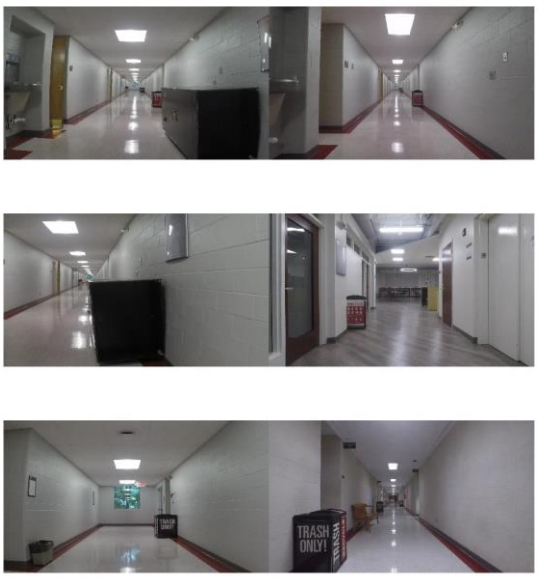
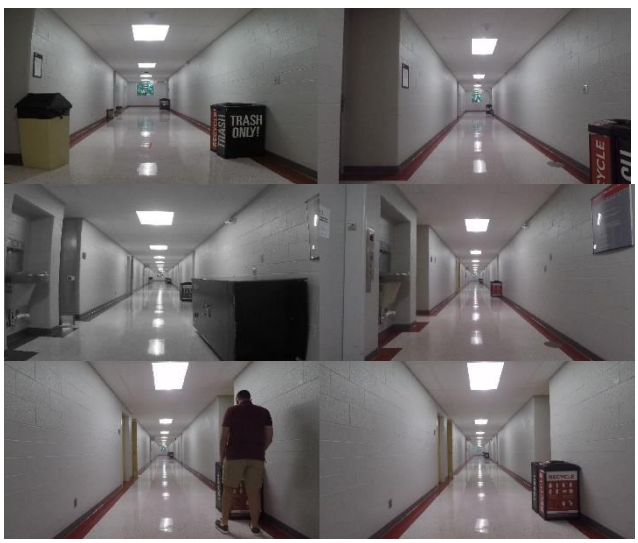


Figure 20: BoVW performance illustration: query (left) and retrieved (right) images.

When assessing image retrieval performance, our methodology was implemented on identical datasets and image samples. Illustrated in Figure 21, our approach consistently achieves success in identifying the closest image to the query images across all instances. The k-d tree algorithm is employed for obtaining results from the nearest neighbor search, providing optimal candidate solutions. Following the determination of the nearest image to the query image, the user's position can be allocated to the pre-defined position associated with the closest image stored in the database.



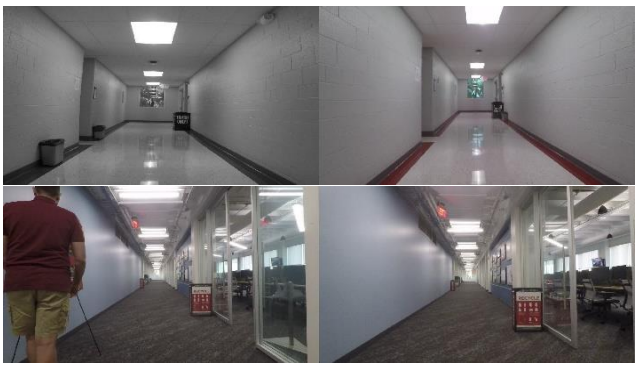


Figure 21: Performance of our approach: query (left) and retrieved (right) images.

To compare findings using Visual Fingerprinting based on BoVW and our proposed algorithm, we employed the cosine similarity approach, represented on a scale from 0 to 1, as depicted in Figure 22. Given the lengthy visual word representation in BoVW, consisting of 20,000 bins, the similarity results for the query image are notably low. In contrast, our proposed algorithm utilizes a visual vector with a length of 64, resulting in a higher cosine similarity compared to BoVW. Figure 23 illustrates that our proposed algorithm successfully identifies the nearest image to the query image, whereas BoVW fails in this regard. Furthermore, Figure 22 displays the cosine distance between the query image and the 20 best candidates for both approaches.

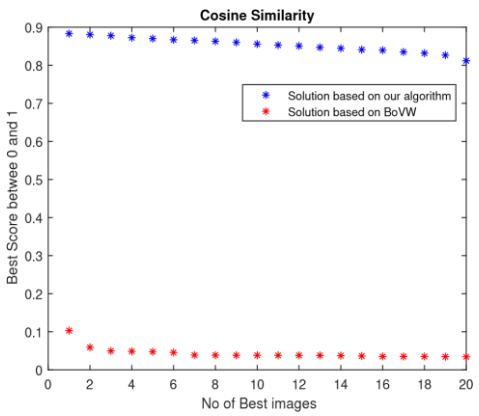


Figure 22: Best similarity score of BoVW and our proposed algorithm candidates.

For the comparison between our approach and the BoVW, we utilized 20 query images, with sample results depicted in Figures 20 and 21. The visual words in BoVW have a length of 20,000 bins, resulting in a very low similarity score for the query image, approximately around 0.1, rendering it highly sensitive to any changes in the visual words, as evidenced in Figure 22. In contrast, our proposed algorithm utilizes a visual vector with a size of 64, leading to a generally higher cosine similarity compared to BoVW. Figure 22 illustrates the cosine distance between the query image and the 20 best candidates for both approaches.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-2-2024-49-2024 | © Author(s) 2024. CC BY 4.0 License.

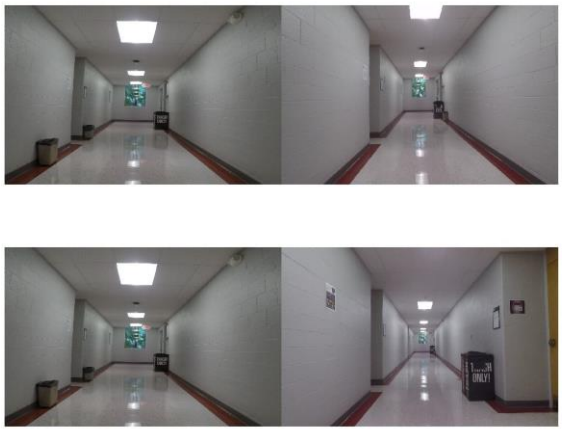


Figure 23: A comparison: proposed method (top row) and BoVW (bottom row).

6. CONCLUSION

This study aims to contribute to vision-based positioning through an image retrieval approach by presenting a solution to address the impact of obstacles in images, which can alter the visual words of the image and potentially lead to the failure of retrieving the correct image(s) from the database. The effectiveness of Bag of Visual Words (BoVW) techniques in image tracking or retrieval applications relies on the stability of the query image features. However, when obstacles are present, these features may undergo changes, resulting in alterations to the image's visual word and, consequently, hindering the retrieval process. In contrast, our proposed algorithm adopts a clustering approach for image features based on objects. This design minimizes the impact of obstacle features on the overall visual word. Even when certain parts of the image are obscured, causing changes in one or more vectors, the remaining vectors remain generally unaffected. By generating a unique vector for each graph, we apply aggregation and updating functions to construct a new node embedding for each node. A global pooling process is then employed to obtain a global vector for each graph. This approach yields distinctive vectors for both the dataset and the query image, thereby enhancing the likelihood of accurate image retrieval. The results demonstrate the efficacy of our solution in overcoming this issue compared to the BoVW technique based on a moderate size image datasets from an office environment. While the BoVW technique achieved a 30% success rate, our method achieved an 80% success rate based on testing with 20 images. Obviously, more testing with larger diverse datasets is required to better assess the performance of the proposed method.

7. REFERENCES

[1] Shady Abd El-Kader Zahran. Enhanced uav navigation under challenging conditions 2019.

[2] MM Mostafa, AM Moussa, Naser El-Sheimy, and Abu B Sesay. Optical flow-based approach for vision aided inertial navigation using regression trees. In Proceedings of the 2017 International Technical Meeting of The Institute of Navigation, pages 856–865, 2017.

[3] Eric Schnipke, Steve Reidling, Jesse Meiring, William Jeffers, Mehdi Hashemi, Ruoyu Tan, Alireza Nemati, and Manish Kumar. Autonomous navigation of uav through gps-denied indoor environment with obstacles. In AIAA Infotech@ Aerospace.

[4] Joel Barnes, Chris Rizos, Mustafa Kanli, David Small, Gavin Voigt, Nunzio Gambale, Jimmy Lamance, Terry Nunan, and Chris Reid. Indoor industrial machine guidance using locata: A pilot study at bluescope steel. In Proceedings of the 60th Annual Meeting of The Institute of Navigation (2004).

[5] Teemu Roos, Petri Myllym¨aki, Henry Tirri, Pauli Misikangas, and Juha Siev¨anen. A probabilistic approach to wlan user location estimation. International Journal of Wireless Information Networks, 9(3):155–164, 2002.

[6] Abd Elgwad M El Ashry, Ezz Eldin Farouk, and Bassem I Sheta. A comparison study of indoor localization methods using available wi-fi signals. In The International Conference on Electrical Engineering, volume 11, pages 1–17. Military Technical College, 2018.

[7] Yuan Yang. Indoor Positioning System for Smart Devices. PhD thesis, The Ohio State University, 2021.

[8] Wisam A Qader, Musa M Ameen, and Bilal I Ahmed. An overview of bag of words; importance, implementation, applications, and challenges. In 2019 International Engineering Conference (IEC), pages 200–204. IEEE, 2019.

[9] Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sa˘so D˘zeroski. Improving bag-of-visual-words image retrieval with predictive clustering trees. Information Sciences, 329:851–865, 2016.

[10] Ehsan Shojaedini, Mahshid Majd, and Reza Safabakhsh. Novel adaptive genetic algorithm sample consensus. Applied Soft Computing, 77:635–642, 2019.

[11] Jun Yang, Yu-Gang Jiang, Alexander G Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In Proceedings of the international workshop on Workshop on multimedia information retrieval, pages 197–206, 2007.

[12] Sergei Vassilvitskii and David Arthur. k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pages 1027–1035, 2006.

[13] Lingfei Wu, Peng Cui, Jian Pei, Liang Zhao, and Le Song. Graph neural networks. In Graph Neural Networks: Foundations, Frontiers, and Applications, pages 27–37. Springer, 2022.

[14] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. arXiv preprint arXiv:1709.05584, 2017.

[15] Dawood Al Chanti and Alice Caplier. Improving bag-of-visual-words towards effective facial expressive image classification. arXiv preprint arXiv:1810.00360, 2018.

[16] William Thomas Tutte and William Thomas Tutte. Graph theory, volume 21. Cambridge university press, 2001.

[17] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. AI Open, 1:57–81, 2020.