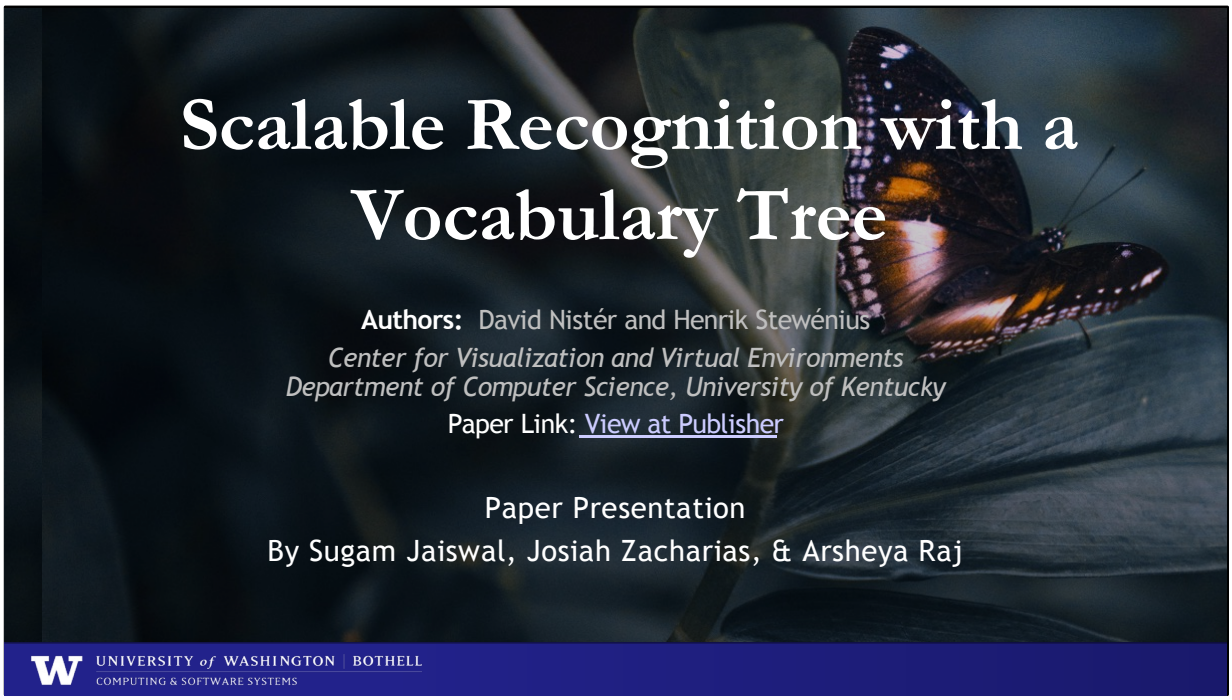


EVERYONE – Arsheya start, 30 sec

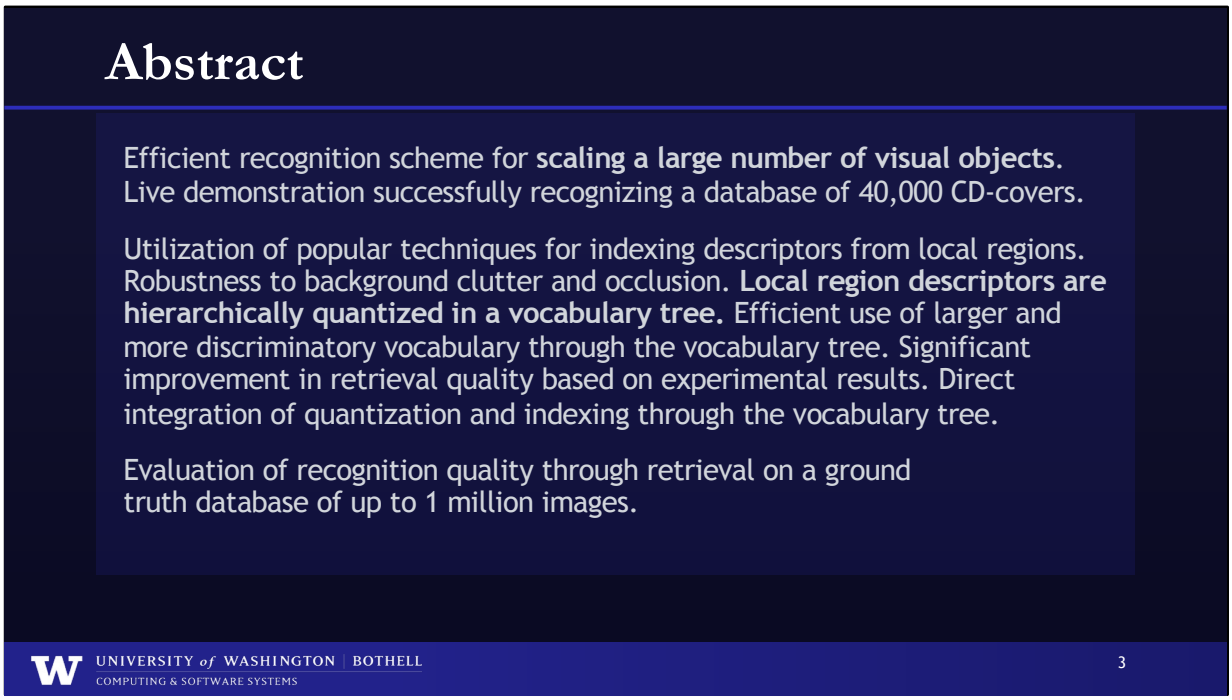
Hi Everyone,
I am Arsheya Raj, AND these are my fellow presenters…[SUGAM AND JOSIAH introduce themselves. Sugam does the worm.]
This paper has been written by David Nister and Henrik Stewenius from the Department of Computer Science,University of Kentucky.

Abstract 1. Introduction - ALL
2. Approach and its Relation to Previous Work - Sugam
3. Building and Using the Vocabulary Tree - Josiah
4. Definition of Scoring - All
5. Implementation of Scoring - Arsheya
6. Results - ALL
7. Conclusions - ALL References - Read the abstracts of some of the related papers in the references

Contents

SUGAM / EVERYONE – 30 sec

A quick preview of what we'll cover and the main section of the paper. Main topics include their methodology and scoring algorithms.



ALL / ARSHEYA – 45 sec

The paper which we chose is called 'Scalable Recognition with a Vocabulary Tree'.

- The paper is **aimed at solving the problem of scalable object recognition** by proposing a scheme to identify and label the objects in images from a large set of labels.
- **Object recognition is one of the most fundamental problems in the domain of computer vision** and it has its own challenges like complexity and variability in images.
- Additionally, an **extremely large number of objects to be recognized severely degrades performance** of object recognition methods and requires large-scale object recognition techniques.
- The **paper solves this problem by proposing a novel approach for object recognition** to handle many objects accurately**, like a bag-of-words(features).**
- To do so, the **paper presents a hierarchical vocabulary tree constructed by recursively applying k-means clustering on a large set of features** to index and match the local features of an image.
- The paper **also uses an inverted file system that stores the image IDs** and feature weights for each leaf node allowing fast retrieval of relevant images.

Introduction

1. Addresses **large scale object recognition** problem in computer vision.
2. Introduces a solution that **scales logarithmically, not linearly**.
3. Utilizes **a tree-structured visual vocabulary with an inverted file system.**
4. Adapts **TF-IDF** (Term Frequency Inverse Document Frequency) **scheme from text retrieval** for object recognition.
5. Potential applications include image matching and video tracking.

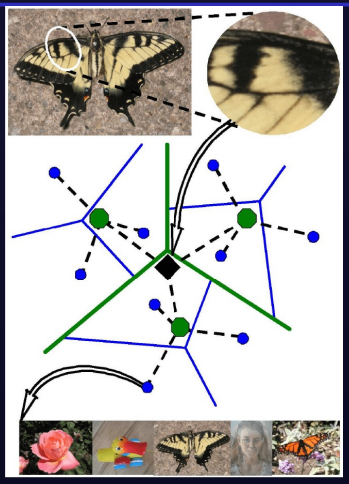


Figure 1: Example of a tree-structured visual vocabulary and an inverted file built from it

ALL / JOSIAH – 1min

1.The paper discusses the issue of large scale object recognition in the domain of computer vision. The scale of this issue has been expanding due to the growing capacity of digital databases and the need to match visual data across an ever-increasing pool of resources.

2.The authors introduce a novel approach which scales logarithmically with the number of objects in the database, making it much more efficient than previous methods that scale linearly.

3.The approach uses an inverted file system with a tree-structured visual vocabulary. The visual vocabulary is created with hierarchical k-means clustering, which is much more effective than traditional flat k-means clustering in dealing with large datasets.

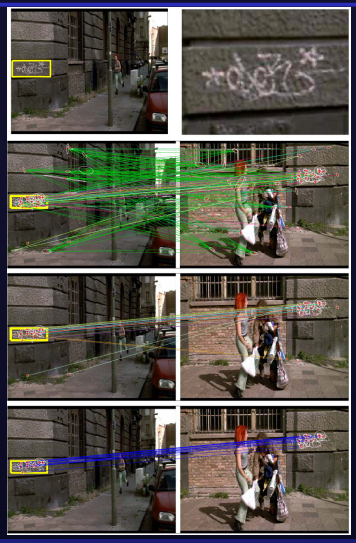
4.The authors explain their use of the TF-IDF (Term Frequency-Inverse Document Frequency) scheme which is a commonly used technique in text retrieval. They show how this can be adapted to the problem of object recognition, which is usually not

straightforward because of the high dimensionality of the descriptor space.

5. The paper's main focus is on object recognition, but the authors state that their approach could also be used for image matching and video tracking.

Related Work

- **Video Google Paper -** Revolutionizing object and scene retrieval in videos through text retrieval techniques.
- **Visual Words** - Using viewpoint invariant region descriptors, vector quantized into 'visual words'.
- **Efficient Object Location & Retrieval** - Utilization of an inverted file system for storing visual word occurrences in video key frames or shots.
- **Efficacy** - Demonstrated high speed and accuracy in object retrieval and localization on two full-length feature films.



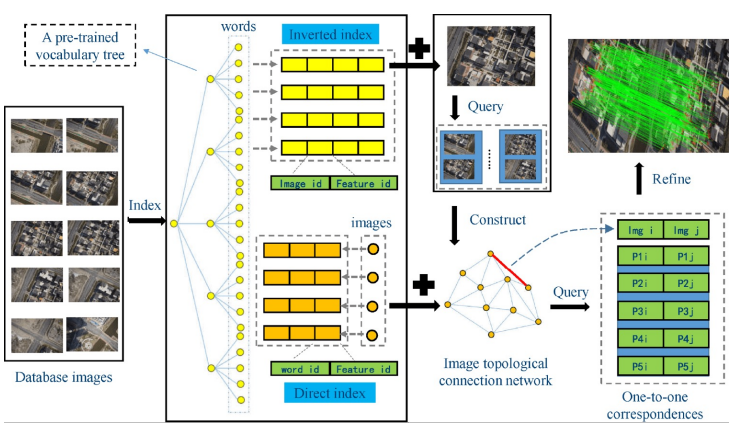
ALL / SUGAM – 1.5m

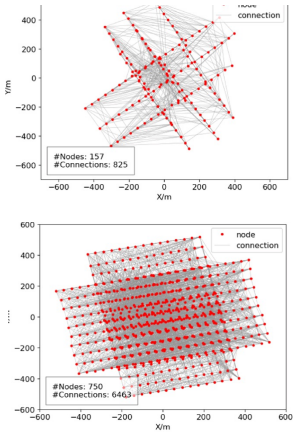
- Differences

Object recognition vs object matching
Vocabulary tree to quantize and index the descriptors vs inverted file to index the features
Images vs videos

•**Video Google Paper** - Published in 2003 by Oxford University, this paper stands as a fundamental pillar in the domain of object and scene retrieval in videos. The approach was a unique blend of text retrieval methods, like those used by Google, applied to object matching in videos.
•**Viewpoint Invariant Region Descriptors** - Central to the paper's method is the representation of objects through a set of viewpoint invariant region descriptors, providing a means of robust object identification regardless of perspective shift.
•**Vector Quantization to Visual Words** - These region descriptors are vector quantized to create 'visual words', converting the visual information into a format akin to text data.
•**Inverted File System** - The paper uses an inverted file system to store the occurrences of each visual word in each key frame or shot of the video, facilitating efficient object location and retrieval.
•**Fast and Efficient Retrieval** - The use of an inverted file system allows for quick and efficient retrieval of all the frames or shots containing a specific object, as outlined by a user query region.
•**Successful Demonstrations** - The paper demonstrated the efficacy of this approach using two full-length feature films, showing high accuracy and speed in retrieving and localizing various objects and scenes.

Related Work





https://www.sciencedirect.com/science/article/abs/pii/S0924271622000727

ALL / JOSIAH – 2.5 minutes

Related paper which helps illustrate the concepts well: “Leveraging vocabulary tree for simultaneous match pair selection and guided feature matching of UAV images” by Wanshou Jiang (https://www.sciencedirect.com/science/article/pii/S0924271622000727)

Overview:

1.Extraction of Local Descriptors and Vocabulary Tree Construction: The algorithm begins by extracting local descriptors from a collection of training images. These descriptors capture crucial visual elements like corners and edges. These are used to represent the content of images. The hierarchical K-means clustering is then employed to construct a vocabulary tree. Here, each node in the tree represents a "visual word", while the centroids at the leaf nodes demonstrate the finest segmentation of the descriptor space.

2.Image Indexing using Pre-trained Vocabulary Tree: Database images are indexed using this pre-trained vocabulary tree. In this process, two types of indices are built - word-image inverted index and image-word direct index. The inverted index maps visual words to the images they appear in, while the direct index maps images to their associated visual words.

3.Similar Image Retrieval and Image Topological Connection Network (TCN): For each query image, similar images are retrieved using the inverted index structure. These results are then used to construct an Image Topological Connection Network. The TCN can be visualized as a graph where each image is a node and connections exist between images with shared visual words.

4.Match Pairing and Refinement: For each match pair, initial matches are first acquired using the acceleration of the direct index. These are then refined through a blend of local and global geometric constraints. This involves using both direct and indirect matches to find geometrically consistent correspondences, leading to a 'one-to-one correspondence' table.

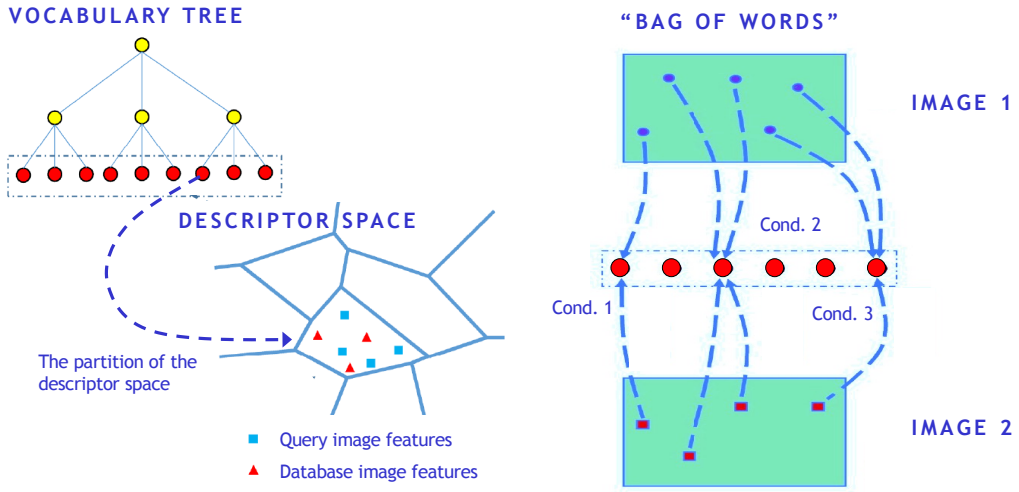
5.Creating a compact representation: Each image is then represented as a histogram of visual words. This allows for a compact and distinctive image representation, enabling efficient comparison of images using metrics like Euclidean or cosine distance.

6.Adaptive Vocabulary Tree-Based Retrieval Technique: An interesting highlight of this work is the use of an adaptive vocabulary tree-based retrieval technique to construct the TCN. This method adjusts the tree structure according to the query image, effectively enhancing retrieval performance.

Refer to Fig. 1 for the workflow of the proposed algorithm and Fig. 12 for illustrations of the Topological Connection Network created using match pairs retrieved by the adaptive vocabulary tree-based retrieval technique. Red circles and gray lines in these

figures denote image positions and match pairs, respectively.

Related Work



https://www.sciencedirect.com/science/article/abs/pii/S0924271622000727

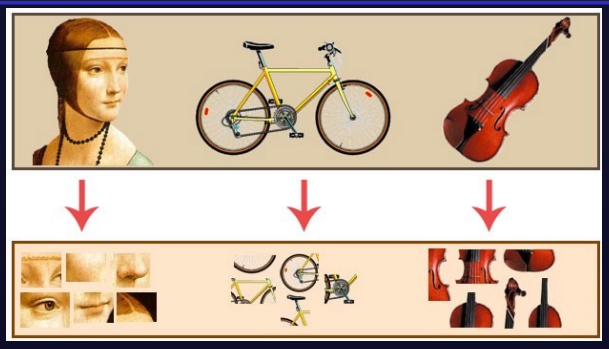
ALL / SUGAM – 2 words

K-means clustering is an unsupervised machine learning algorithm that groups similar data points together based on their features. The algorithm partitions the dataset into K clusters, each with a centroid (center point) that represents the mean of the cluster members. The objective is to minimize the distance between each data point and its respective centroid.

The main idea behind this approach is to represent images by local descriptors (e.g., SIFT) and then cluster these descriptors into a hierarchical structure known as a vocabulary tree.
Explain branching factor and level for heirarchial k-means clustering ->Right image: Example of node formation using bag of visual

Approach – Feature Extraction

1. Use MSERs to **detect stable regions** in images.
2. Make **regions invariant to rotation and scale** by warping into circular patches.
3. Determine **canonical directions based on histogram** of image gradients.
4. **Extract SIFT descriptors** relative to canonical directions.



5. Quantize SIFT descriptors with vocabulary tree.
6. Use vocabulary tree and hierarchical scoring to retrieve images.

SUGAM First, ARSHEYA second – 2.5 min

- Detect Maximally Stable External Regions(MSERs), which are regions of pixels that have stable boundaries under different transformations.
- Warp each MSER region into a circular patch to make it invariant to rotation and scale.
- Find canonical directions for each patch based on the histogram of image gradients, which are changes in pixel intensity.

Arsheya

- Extract SIFT descriptors for each patch relative to the canonical directions. SIFT descriptors are vectors that describe the local appearance of the patch.
- Quantize the SIFT descriptors with a vocabulary tree, which is a hierarchical structure that clusters similar descriptors into nodes.
- Score and retrieve images from a database using the vocabulary tree and a hierarchical scoring scheme, which compares the frequency of nodes between query and database images

Approach – Building Tree

1. Building vocabulary tree **starts with a training set of images**
2. **Hierarchical k-means clustering organizes visual words** into a tree structure
3. Branching factor and tree depth determine total number of leaf nodes
4. **Each leaf node represents a "visual word"** - a cluster of similar SIFT descriptors
5. **Image descriptors vote for most similar leaf node** when query image is processed
6. Votes form a histogram, or a **"bag of words" representation** of an image

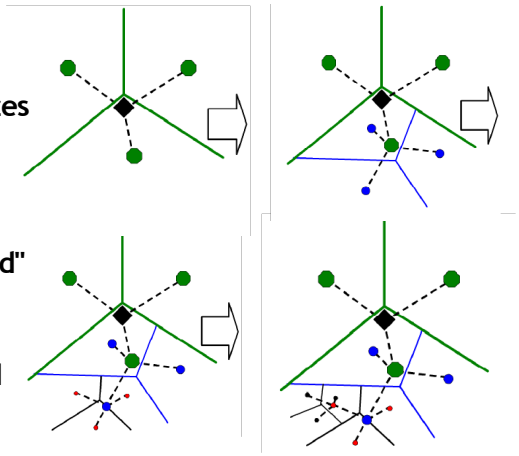


Figure 2: Illustration of the Vocabulary Tree with depiction of the branching factor (k), tree depth (L), and the voting process

JOSIAH – 2 min

1. The authors discuss the building of the vocabulary tree, which begins with a training set of images. SIFT descriptors are extracted from these images to serve as "visual words".
2. Hierarchical k-means clustering is used to organize these visual words into a tree structure, with each node representing a cluster of descriptors and the root node representing the entire training set.
3. The authors use a branching factor (k) and tree depth (L) to determine the total number of leaf nodes in the tree (visual words in the vocabulary).
4. Each leaf node in the vocabulary tree corresponds to a "visual word" and represents a cluster of similar SIFT descriptors. The leaf nodes are where the actual image descriptors from the training set are stored.
5. To use the vocabulary tree for image retrieval or object recognition, SIFT descriptors are extracted from a query image and then propagated down the tree. Each descriptor "votes" for the most similar leaf node (or visual word) it encounters.
6. Votes are accumulated in an inverted file to form a histogram that represents the frequency of occurrence of each visual word in the image. The paper refers to this

histogram as the "bag of words" representation of an image.

Approach – Image Indexing

1. **Image Indexing:** Map visual words to images where they occur by creating an "inverted file"
2. Descriptors from each image **vote for the most similar leaf node** in the Vocabulary Tree
3. **Votes are stored in the inverted file**, associating visual words with images
4. Resulting **histogram (or bag of words) is used for image retrieval** or recognition
5. **Logarithmic indexing significantly reduces the size of descriptor vectors** without compromising performance

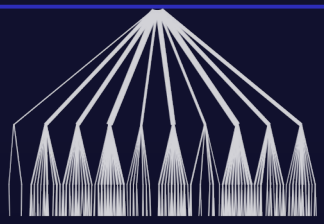


Figure 3: Illustration of an inverted file / tree with 3 levels: branch factor 10 on image with 400 features

$$\sum_{i=1}^{L} k^i = \frac{k^{L+1} - k}{k - 1} \approx k^L$$

Total Descriptor Vectors in Tree: Computational cost in the hierarchical approach is logarithmic in the number of leaf nodes. For D-dimensional descriptors represented, as *char* the size of the tree is approximately Dk^L bytes. With our current implementation, a tree with D = 128, L = 6 and k = 10, resulting in 1M leaf nodes, uses 143MB of memory.

JOSIAH / SUGAM? 2 - 2.5

1. Image Indexing involves mapping visual words (leaf nodes of the Vocabulary Tree) to the images where they occur. This is accomplished by creating an "inverted file".
2. Inverted files are data structures used in information retrieval systems to store a mapping from content, such as words or numbers, to their locations in a database file, or a document.
3. When an image is processed, its descriptors are propagated down the tree. Each descriptor "votes" for the most similar leaf node (or visual word) it encounters.

Sugam finishes

1. These votes are stored in the inverted file, allowing each visual word to be associated with the images in which it occurs.
2. The result is a histogram, or a "bag of words" representation of an image, which is used for image retrieval or object recognition.
3. Logarithmic indexing is also discussed in the paper, where the authors argue that the size of the descriptor vectors used in image recognition systems can be significantly reduced without compromising performance.

Scoring – Definition

Query vs database image:

- Relevance **based on similarity of descriptor paths** in tree
- **Assign weights to nodes** based on entropy for query and db vectors

$$q_i = n_i w_i \quad (1)$$

$$d_i = m_i w_i \quad (2)$$

- **Calculated using normalized difference** between query and db vectors

$$s(q, d) = \left|\left| \frac{q}{||q||} - \frac{d}{||d||} \right|\right| \quad (3)$$

- **L1-norm > L2-norm:** achieves better results
- Entropy weighting improves retrieval performance **(TF-IDF scheme)**

$$w_i = \ln \frac{N}{N_i} \quad (4)$$

- Consideration of **frequency of occurrence and dependencies** within the path
  - Handling weights for different levels of the vocabulary tree

SUGAM START, / ARSHEYA – 3

- AFTER CLUSTERS ARE DONE

- Hierarchical scoring reducing the risk of vocabulary size overdoing. Increase in retrieval performance with the number of leaf nodes
- Usage of stop lists to set weights to zero for frequent/infrequent symbols. Blocking longer lists in inverted files for efficiency and improved performance. Limited improvement in retrieval quality with stop lists
- Importance of a large vocabulary size and avoiding overly strong weights for inner nodes
- Trade-off between distinctiveness and repeatability in quantization cells and vocabulary tree depth

Scoring – Implementation

- **Inverted Files:** Efficient scoring with large databases. They store image IDs and term frequency for each node.
- **Forward Files:** Complement inverted files, representing leaf nodes explicitly.
- **Normalized Difference:** Calculated in Lp norm using equation (5).

$$\| q - d \|_p^p = \sum_i |qi - di|^p \quad (5)$$

$$= \sum_{i|di=0} |qi|^p + \sum_{i|qi=0} |di|^p + \sum_{i|qi\neq 0,di\neq 0} |qi - di|^p$$

$$= \| q \|_p^p + \| d \|_p^p + \sum_{i|qi\neq 0,di\neq 0} (|qi - di|^p - |q_i|^p - |d_i|^p)$$

$$= 2 + \sum_{i|qi\neq 0,di\neq 0} (|qi - di|^p - |q_i|^p - |di|^p)$$

- **Inverted Files:** Traverse and accumulate database entries for non-zero query dimensions.
- **Query Implementation:** Populate and sort a query tree.
- **L2-Norm:** Simplified equation (5) to equation (6).

$$\| q - d \|_2^2 = 2 - 2 \sum_{i|qi\neq 0,di\neq 0} q_i d_i \quad (6)$$

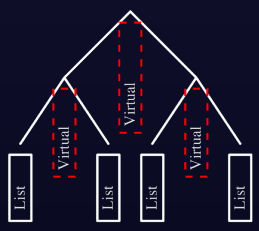


Figure 4: The database structure shown with two levels and a branch factor of two. The leaf nodes have explicit inverted files, and the inner nodes have virtual inverted files that are computed as the concatenation of the inverted files of the leaf nodes.

ARSHEYA - 3

- Usage of inverted files for efficient scoring with large databases. Inverted files store image id-numbers and term frequency for each node
- Forward files as a complement to identify visual words in an image. Explicit representation of leaf nodes, while inverted files of inner nodes are concatenation of leaf node files
- Computation of normalized difference in Lp norm using equation (5)
- Usage of inverted files for non-zero query dimensions to traverse and accumulate database entries
- Implementation of query by populating and sorting a query tree
- Simplification of equation (5) for L2-norm in equation (6)
- Partitioning of scalar product for other norms

- Length of inverted file stored in each node, indicating document frequency and entropy determination
- Blocking scoring for inverted files above a certain length
- Pre-computation or representative database for fixed and known node entropies.

- Pre-computation and normalization of database and query vectors to unit magnitude
- Usage of virtual inverted files for fragmentation of database vector dimensions
- Composition of d_i by remembering the last touched node and accumulated d_i for each database image
- Usage of accumulated d_i in equation (5) for scoring.

Results - Dataset

1. Three datasets used:
   - 40,000 images (University of Kentucky)
   - 10,000 images (Corel)
   - 4.1 million images (Internet)

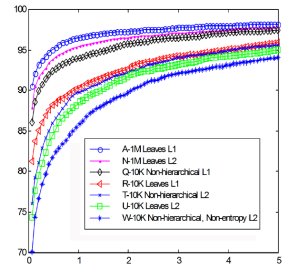


Figure 6: Depicts top query results accuracy for a 1400 image database.
Key insights:
- Larger vocabularies and L1-norm enhance retrieval
- Entropy weighting is crucial for smaller vocabularies
- Method A outperforms Video Google paper's approach.

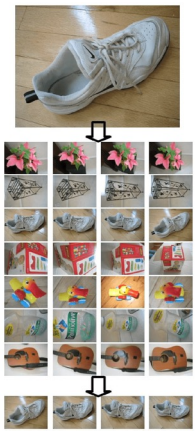


Figure 5: Illustration of the Vocabulary Tree with depiction of the branching factor (k), tree depth (L), and the voting process

Me	En	No	S%	Voc-Tree	Le	Eb	Perf
A	**y/y**	**L1**	**0**	**6x10=1M**	**1**	**ir**	**90.6**
B	y/y	L1	0	6x10=1M	1	vr	90.6
C	y/y	L1	0	6x10=1M	2	ir	90.4
D	n/y	L1	0	6x10=1M	2	ir	90.4
E	y/n	L1	0	6x10=1M	2	ir	90.4
F	n/n	L1	0	6x10=1M	2	ir	90.4
G	n/n	L1	0	6x10=1M	1	ir	90.2
H	y/y	L1	m2	6x10=1M	1	ir	90.0
I	y/y	L1	0	6x10=1M	3	ir	89.9
J	y/y	L1	0	6x10=1M	4	ir	89.9
K	y/y	L1	0	6x10=1M	2	vr	89.8
L	y/y	L1	0	6x10=1M	2	ip	89.0
M	y/y	L1	m5	6x10=1M	1	ir	89.1
N	**y/y**	**L2**	**0**	**6x10=1M**	**1**	**ir**	**87.9**
O	y/y	L2	0	6x10=1M	2	ir	86.6
P	y/y	L1	110	6x10=1M	2	ir	86.5
Q	**y/y**	**L1**	**0**	**1x10K=10K**	**1**	**-**	**86.0**
R	**y/y**	**L1**	**0**	**4x10=10K**	**2**	**ir**	**81.3**
S	y/y	L1	0	4x10=10K	1	ir	80.9
T	**y/y**	**L2**	**0**	**1x10K=10K**	**1**	**-**	**76.0**
U	**y/y**	**L2**	**0**	**4x10=10K**	**1**	**ir**	**74.4**
V	y/y	L2	0	4x10=10K	2	ir	72.5
W	**n/n**	**L2**	**0**	**1x10K=10K**	**1**	**-**	**70.1**

Table 1: Illustration of the Vocabulary Tree with depiction of the branching factor (k), tree depth (L), and the voting process

JOSIAH / ALL – 1.5

1.Datasets: The study utilized diverse databases to evaluate the vocabulary tree approach:

1. University of Kentucky database: 40,000 images, primarily of recognizable objects.
2. Corel database: 10,000 images, covering a range of scenes and objects.
3. Internet database: 4.1 million images (seemingly at random?), demonstrating scalability of the approach.

2.Figure 5: This figure assesses retrieval performance using a large ground truth database (6376 images), with each image group containing four variations of the same object. The goal is for the three variations of an object to top the query result when one is used as the query image. The figure allows comparison against non-hierarchical schemes using a 1400 image subset.

3.Figure 6 - Query Results Accuracy: Illustrates the accuracy of top query results for a 1400 image database, emphasizing the importance of ranking correct images at the top for scalable retrieval. Findings include:

1. Larger vocabularies improve retrieval performance.
2. L1-norm outperforms L2-norm in retrieval.

3. Entropy weighting is crucial for smaller vocabularies.
4. Method A (proposed approach) outperforms the Video Google paper's approach.

1.Table 1: Shows the percentage of queries resulting in perfect retrieval for different scoring methods and settings. It provides insights into the relationship between settings and retrieval performance, particularly in relation to Figure 6.

Results - Performance

1. Retrieval speed increases linearly with logarithm of database size.
2. Recognition performance also increases with database size.
3. Achieved 79% recall at 95% precision on University of Kentucky database.

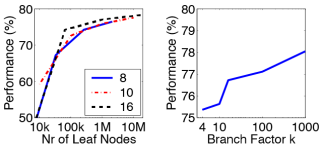


Figure 7: Testing tree shapes on 6376 images. Left: Performance increases with leaf nodes. Right: Moderate boost with branch factor k.

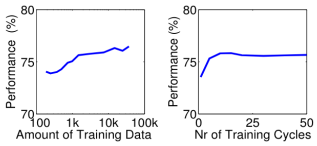


Figure 9: Effects of unsupervised vocabulary tree training. Left: Performance versus training data volume (20 training cycles). Right: Performance versus training cycles (7K frames). Training was separate from the database, tests ran on a 6 x 10 vocabulary tree on the 6376 image set.

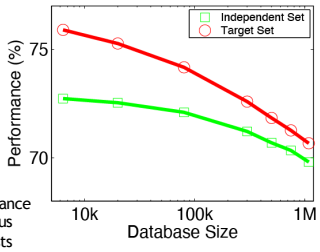


Figure 8: Performance assessed for database sizes up to 1 million images. Vocabulary tree entropy weighting defined two ways, with significant results from video-independent definition. Comparisons also made using ground truth target image subset.

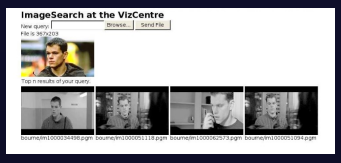
SUGAM / ALL –1.5 mins

1. The paper presents results on three different data sets: a proprietary 40,000-image set from the University of Kentucky, the Corel set of 10,000 images, and a 4.1-million-image set downloaded from the Internet.
2. The authors find that retrieval speed increases linearly with the logarithm of the database size, validating the scalability of their approach.
3. Quantitatively, they demonstrate that recognition performance increases steadily with the database size.
4. On the proprietary University of Kentucky database, they were able to retrieve with 79% recall at 95% precision.

Results – Comparison with Others

1. Significantly better recognition rates than previously published results.
2. Outperforms other scalable recognition systems in retrieval speed and recognition accuracy.
3. Despite database size increase from 10,000 to 4.1 million, recall rate dropped only from 85% to 79%.

Figure 10: CD-cover recognition in action: robust real-time retrieval from 40k images, despite occlusion and changes in view. Successful recognition triggers music.



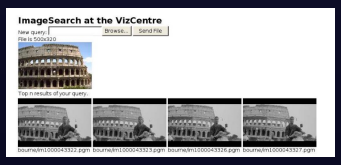


Figure 11:

Top: Less reliable face search, performed on smaller 300K frame database. Both use images separate from the movies.

Bottom: Search in one million images, retrieves rigid objects like CDs, buildings effectively.

ARSHEYA / ALL - 2mins

1. They compare their method with the best reported results on the same University of Kentucky database.
2. The authors find that they were able to achieve significantly better recognition rates than previously published results using the same dataset.
3. Additionally, their results on the Corel and internet databases indicate that their method outperforms other scalable recognition systems in terms of both retrieval speed and recognition accuracy.
4. Despite the increase in database size from 10,000 to 4.1 million images, the recall rate only dropped from 85% to 79% which showcases the robustness of the proposed method.

Discussion

1. **Implications:** The vocabulary tree is a scalable and efficient method for image retrieval, essential for large databases.
2. **Applications:** Potential usage extends to a range of image and video retrieval tasks, including object and scene recognition in multimedia databases and real-world scenarios.
3. **Limitations:** While the method has shown promise, limitations include challenges in recognizing non-rigid, region-poor objects, or faces.
4. **Project Relevance:** The vocabulary tree's scalability and efficiency align with our project's requirements for handling large image databases.

EVERYONE – 1 – 5

1.Implications: The vocabulary tree method can efficiently manage large image databases, addressing scalability issues in image retrieval.
2.Applications: The method's utility extends beyond academic experiments, potentially being applied in object/scene recognition in multimedia databases, surveillance systems, or internet search engines.
3.Limitations: The method's performance may be less reliable when dealing with non-rigid, region-poor objects or faces, warranting further research in these areas.
4.Project Relevance: Given our project's need to handle large image databases efficiently, the vocabulary tree method offers a promising approach that aligns with our objectives.

Sugam start, Josiah ends

1. **Unrivaled Image Recognition:** The approach surpasses the state-of-the-art, delivering powerful recognition capabilities with an advanced indexing scheme.
2. **Hierarchical Vocabulary Tree:** Used for efficient quantization of image keypoints, setting the foundation for effective image retrieval.
3. **Larger Vocabularies & L1-norm:** Enhanced retrieval performance by utilizing larger vocabulary size and adopting L1-norm for defining image similarity.

Josiah starts

1. **Real-time Demonstration:** Proven functionality showcased via a real-time demonstration with 50K CD cover images.
2. **Scalability: 1M Image Database:** The approach demonstrates scalability through successful execution of second timing queries on a 1M image database.
3. **Future: Internet-scale Search Engine:** Presents the potential for development of a content-based image and video search engine on an internet-scale.

References & Appendix

Nister, D., & Stewenius, H. (2006). *Scalable Recognition with a Vocabulary Tree*. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) (Vol. 2, pp. 2161-2168). [IEEE].

Sivic, J., & Zisserman, A. (2003). *Video Google: A text retrieval approach to object matching in videos*. In Proceedings of the Ninth IEEE International Conference on Computer Vision, 1470-1477. https://www.robots.ox.ac.uk/~vgg/publications/2003/Sivic03/sivic03.pdf

Jiang, S.; Jiang, W.; Guo, B. *Leveraging vocabulary tree for simultaneous match pair selection and guided feature matching of UAV images*. ISPRS J. Photogramm. Remote Sens. 2022, 187, 273–293. [ScienceDirect].

ALL -