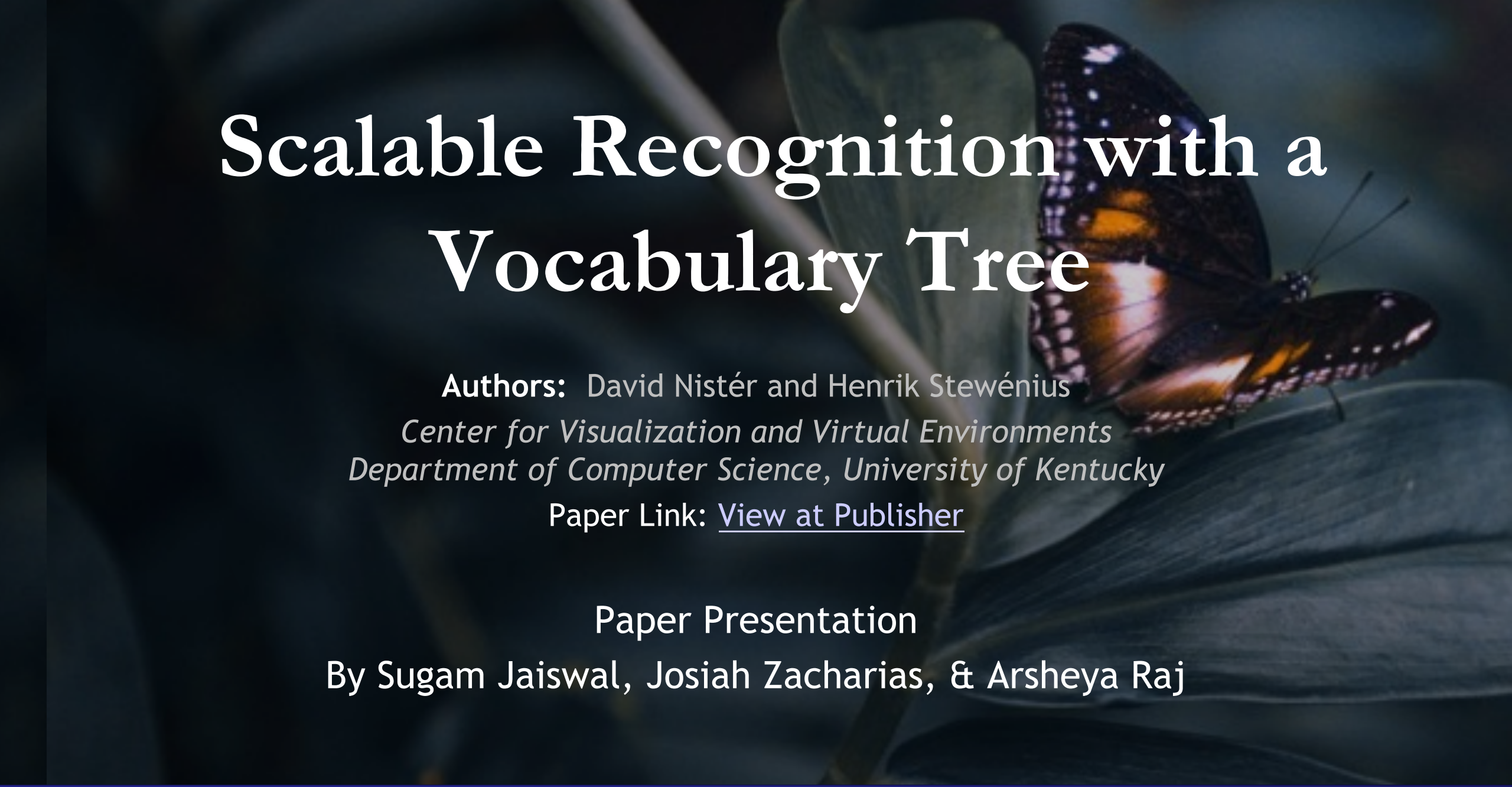


Scalable Recognition with a Vocabulary Tree

Authors: David Nistér and Henrik Stewénius

Center for Visualization and Virtual Environments
Department of Computer Science, University of Kentucky

Paper Link: View at Publisher

Paper Presentation

By Sugam Jaiswal, Josiah Zacharias, & Arsheya Raj

Contents

Abstract

Efficient recognition scheme for **scaling a large number of visual objects**. Live demonstration successfully recognizing a database of 40,000 CD-covers.

Utilization of popular techniques for indexing descriptors from local regions. Robustness to background clutter and occlusion. **Local region descriptors are hierarchically quantized in a vocabulary tree.** Efficient use of larger and more discriminatory vocabulary through the vocabulary tree. Significant improvement in retrieval quality based on experimental results. Direct integration of quantization and indexing through the vocabulary tree.

Evaluation of recognition quality through retrieval on a ground truth database of up to 1 million 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.

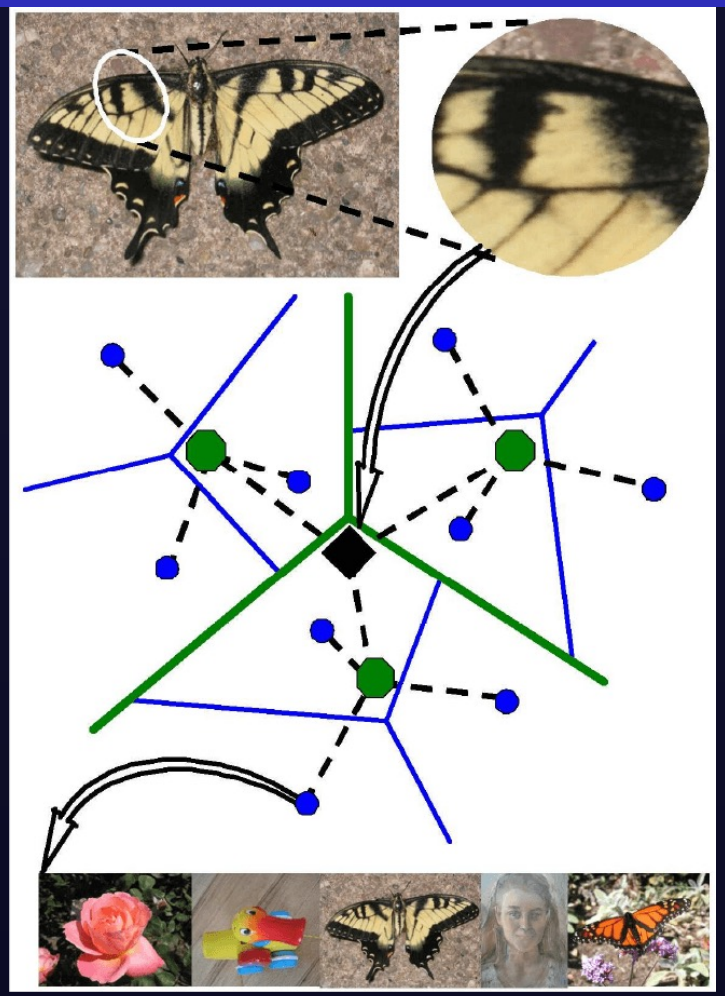
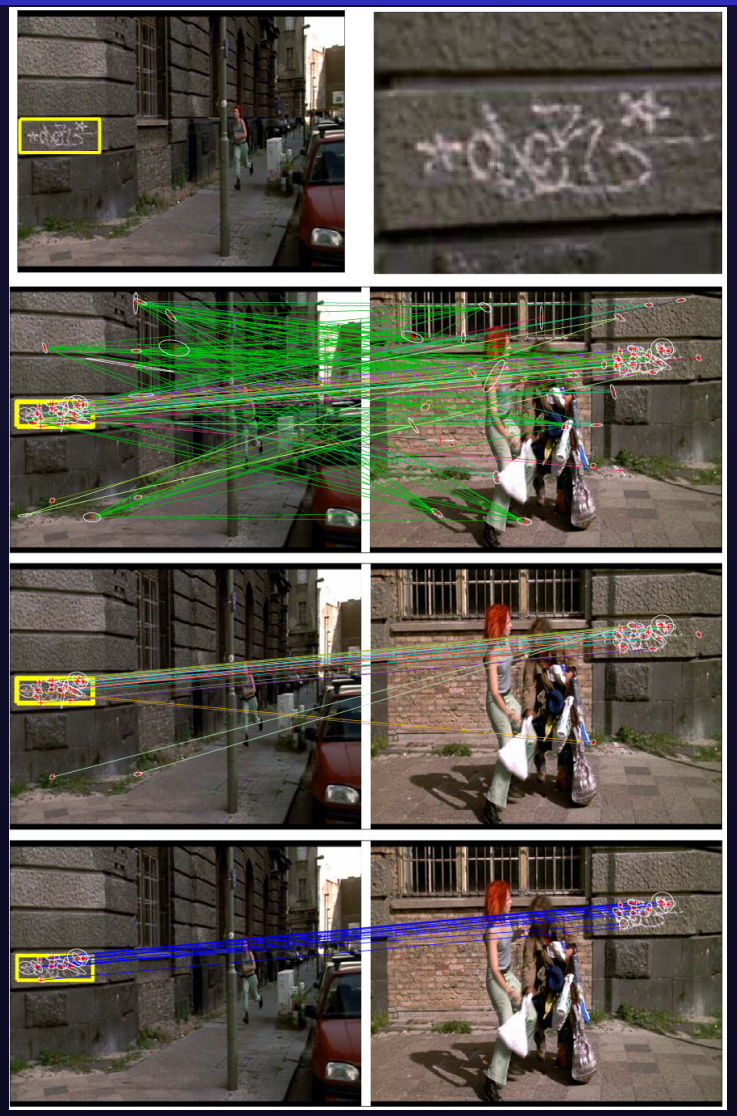


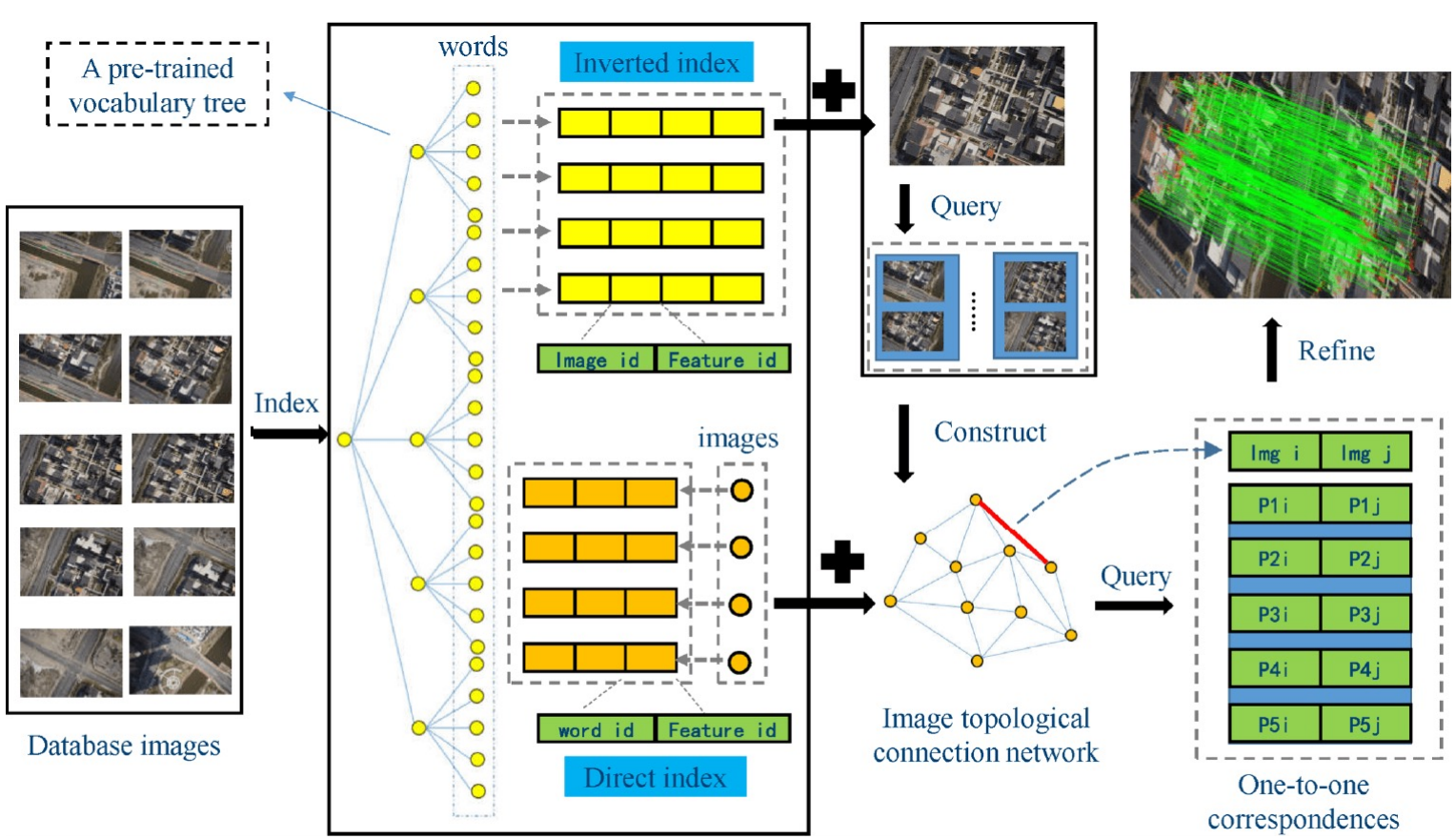
Figure 1: Example of a tree-structured visual vocabulary and an inverted file built from it

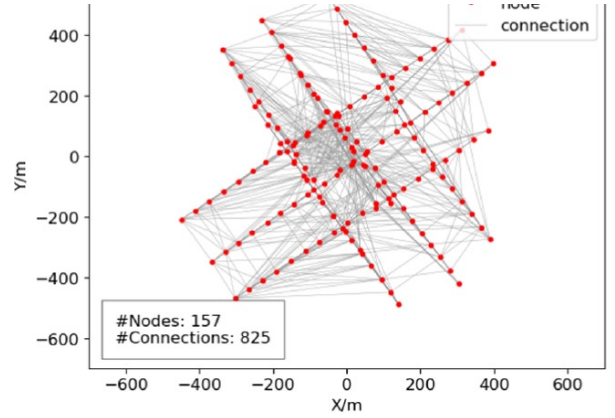
Related Work

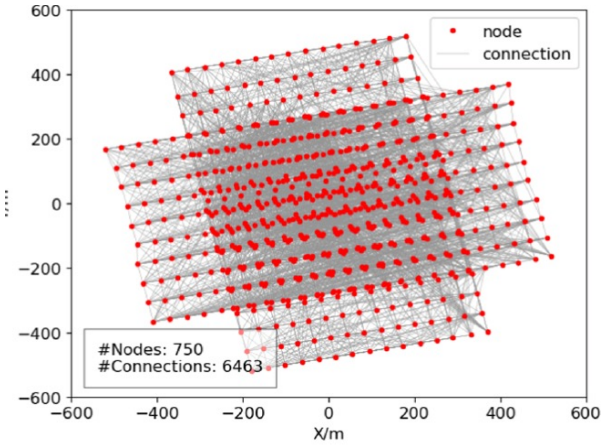
- **Video Google Paper -** Revolutionizing object and scene retrieval in videos through text retrieval techniques.
- **Visual Words -** Using viewpoint invariant region descriptors, vector quantized into 'visual words'.
- **Efficient Object Location & Retrieval -** Utilization of an inverted file system 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.



Related Work

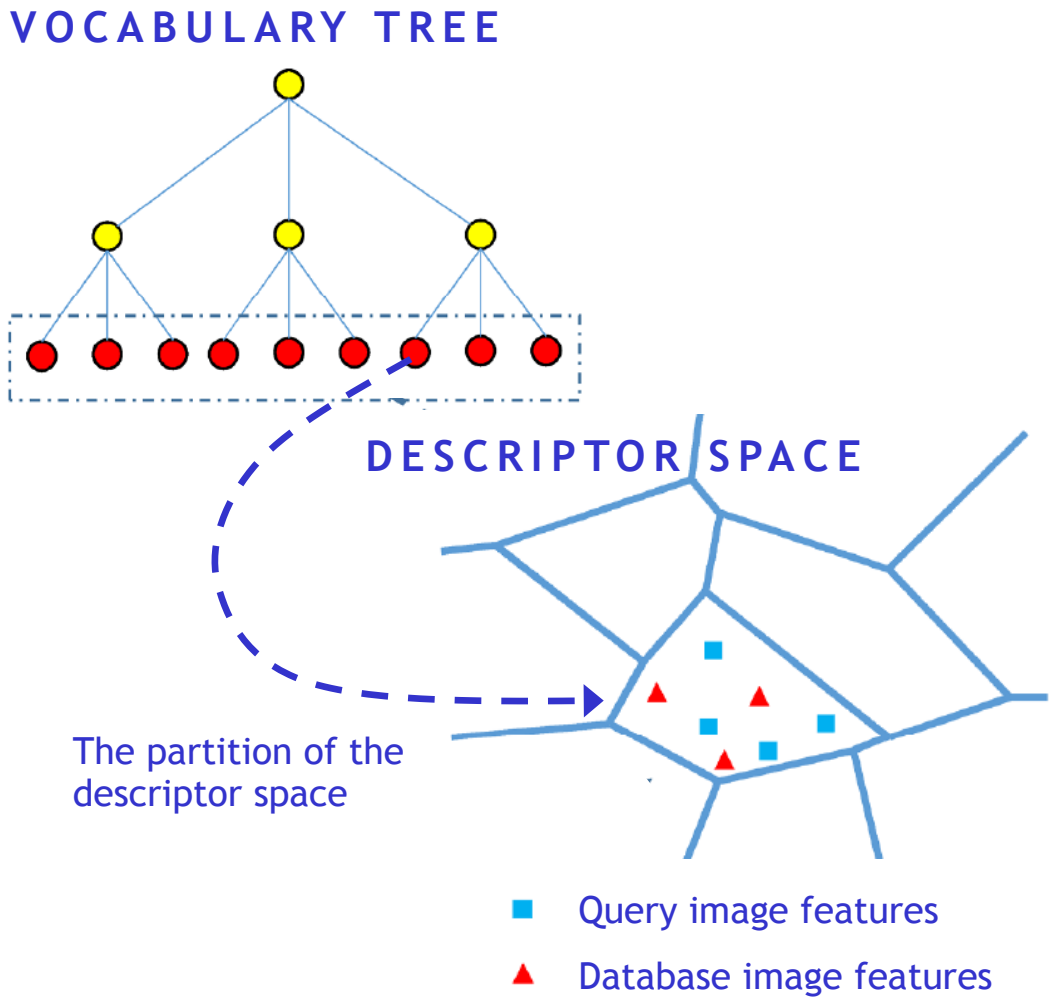


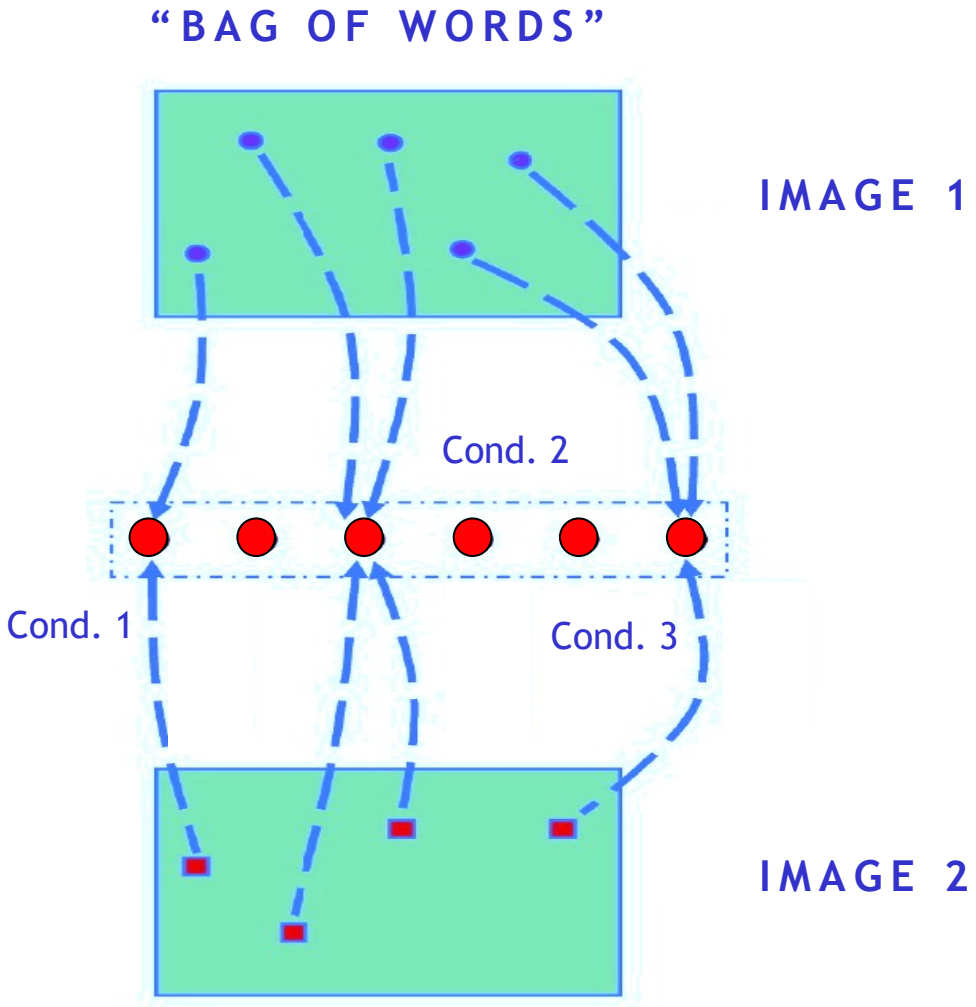




https://www.sciencedirect.com/science/article/abs/pii/S0924271622000727

Related Work

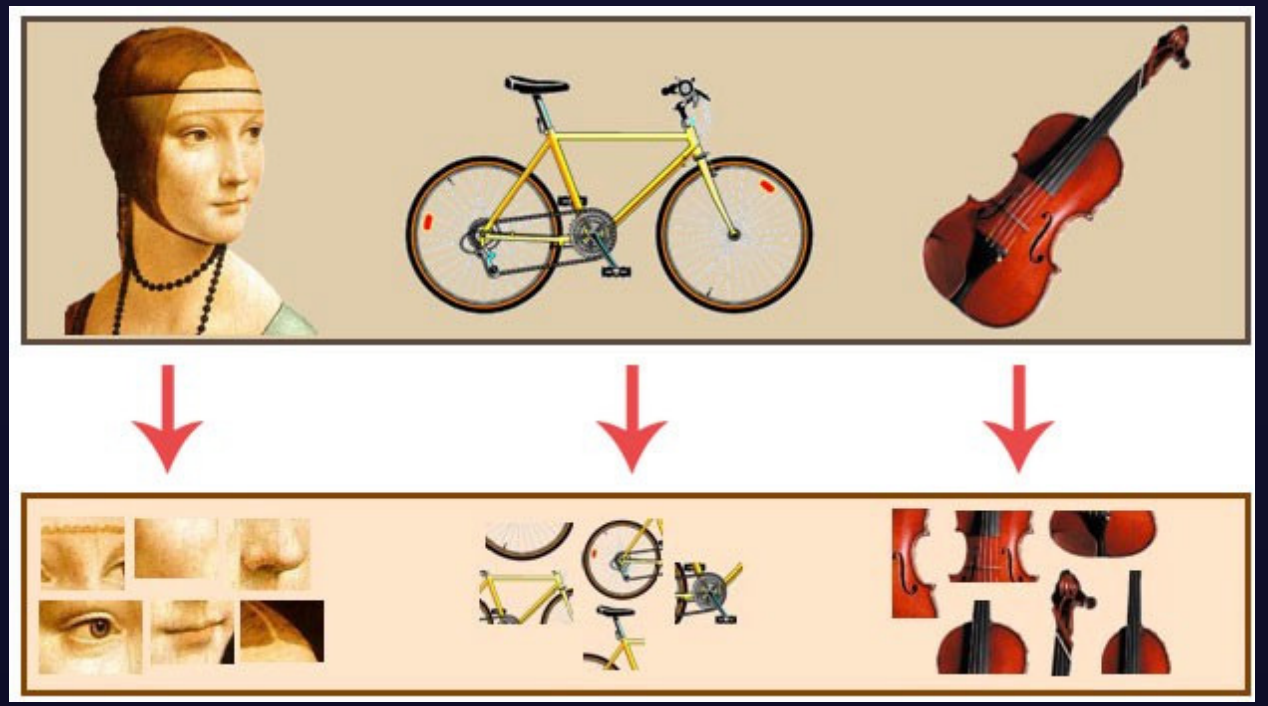




https://www.sciencedirect.com/science/article/abs/pii/S0924271622000727

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.

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

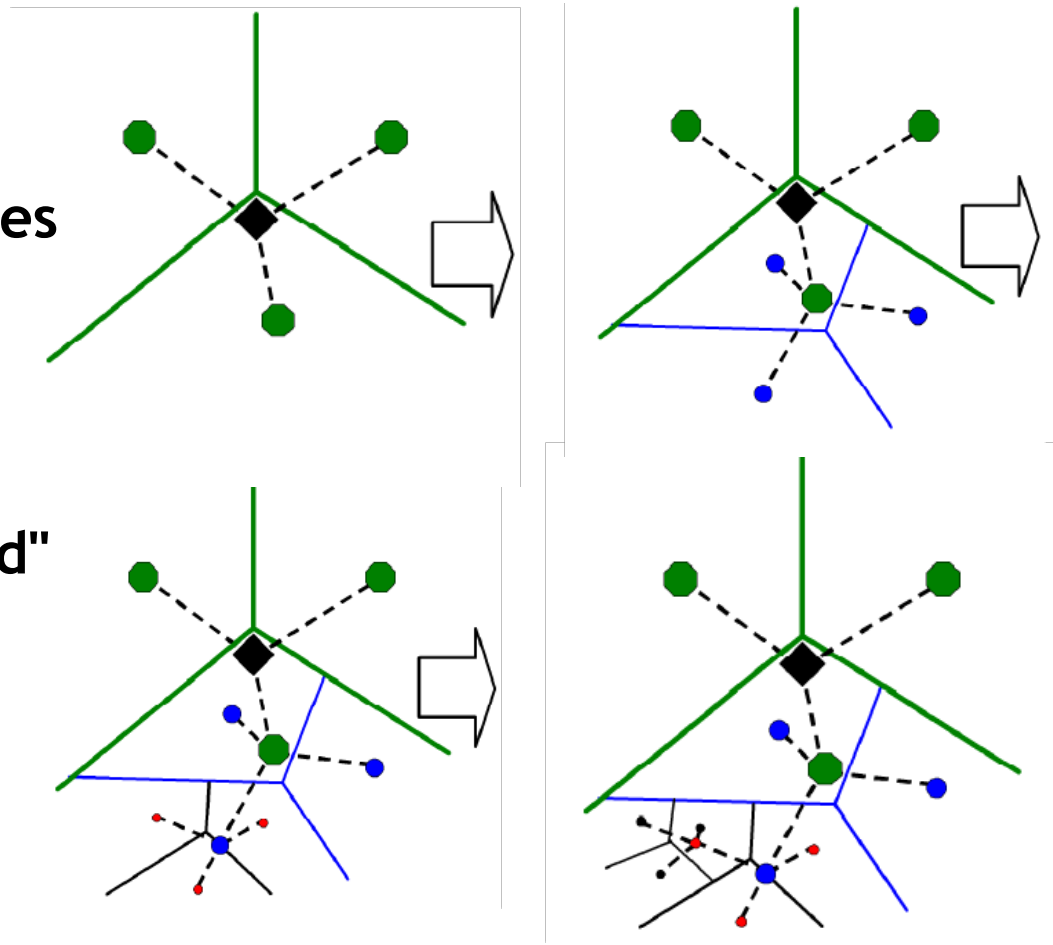


Figure 2: Illustration of the Vocabulary Tree with depiction of the branching factor (k), tree depth (L), and the voting process

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

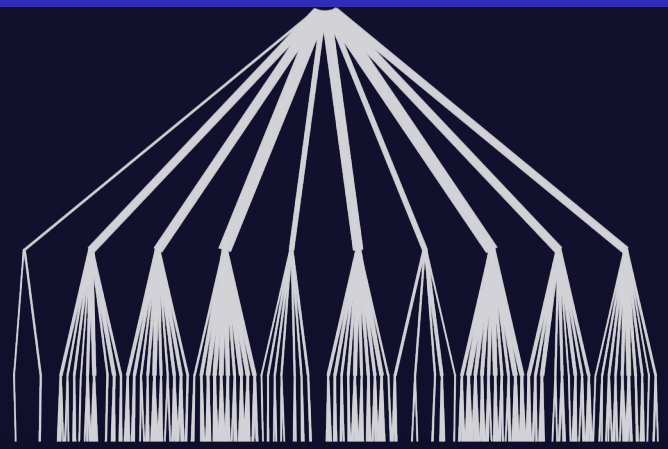


Figure 3: Illustration of an inverted file / tree with 3 levels: branch factor 10 on image with 400 features

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

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.

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

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).

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

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

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

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

- **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).

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

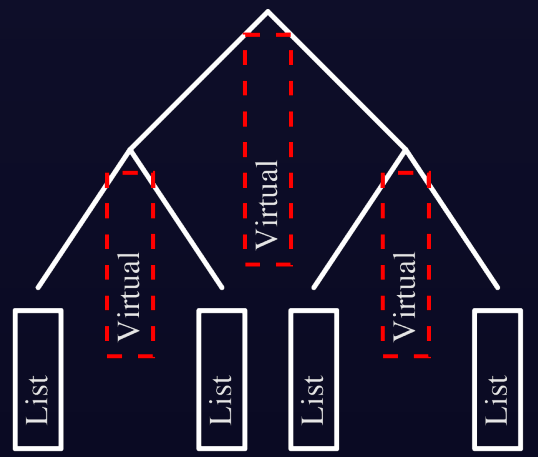


Figure 4: The database structure shown with two levels and a branch factor of two. The leaf nodes have explicit inverted files, and the inner nodes have virtual inverted files that are computed as the concatenation of the inverted files of the leaf nodes.

Results - Dataset

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

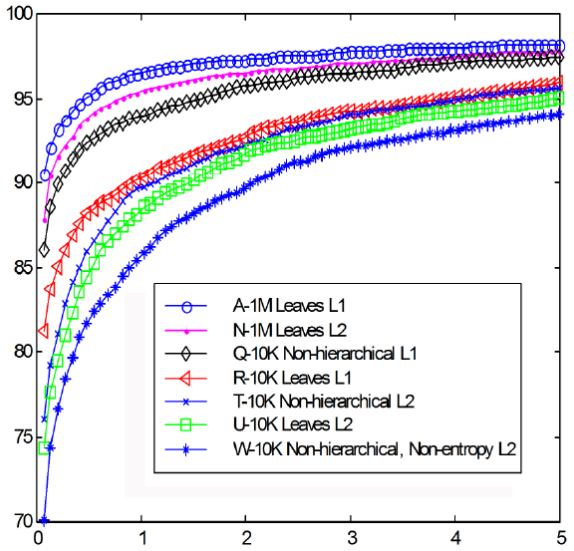


Figure 6: Depicts top query results accuracy for a 1400 image database.

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	l10	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

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.

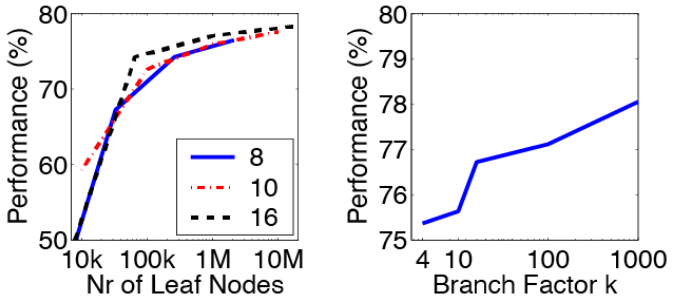


Figure 7: Testing tree shapes on 6376 images. Left: Performance increases with leaf nodes. Right: Moderate boost with branch factor k.

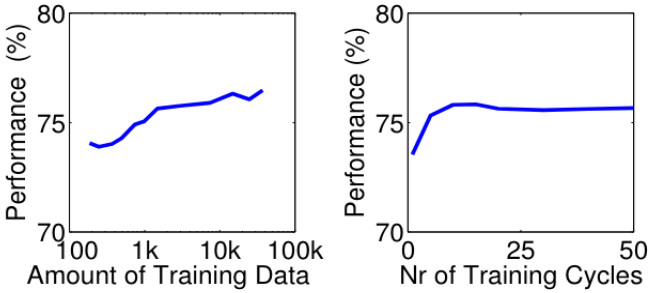


Figure 9: Effects of unsupervised vocabulary tree training. Left: Performance versus training data volume (20 training cycles). Right: Performance versus training cycles (7K frames). Training was separate from the database, tests ran on a 6 x 10 vocabulary tree on the 6376 image set.

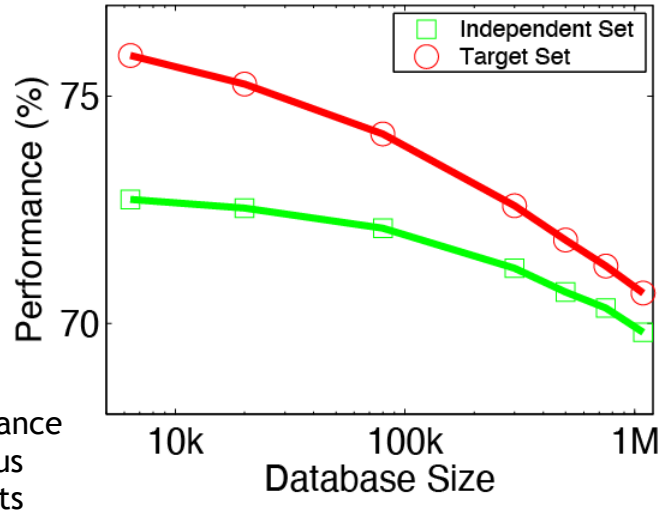


Figure 8: Performance assessed for database sizes up to 1 million images. Vocabulary tree entropy weighting defined two ways, with significant results from video-independent definition. Comparisons also made using ground truth target image subset.

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:

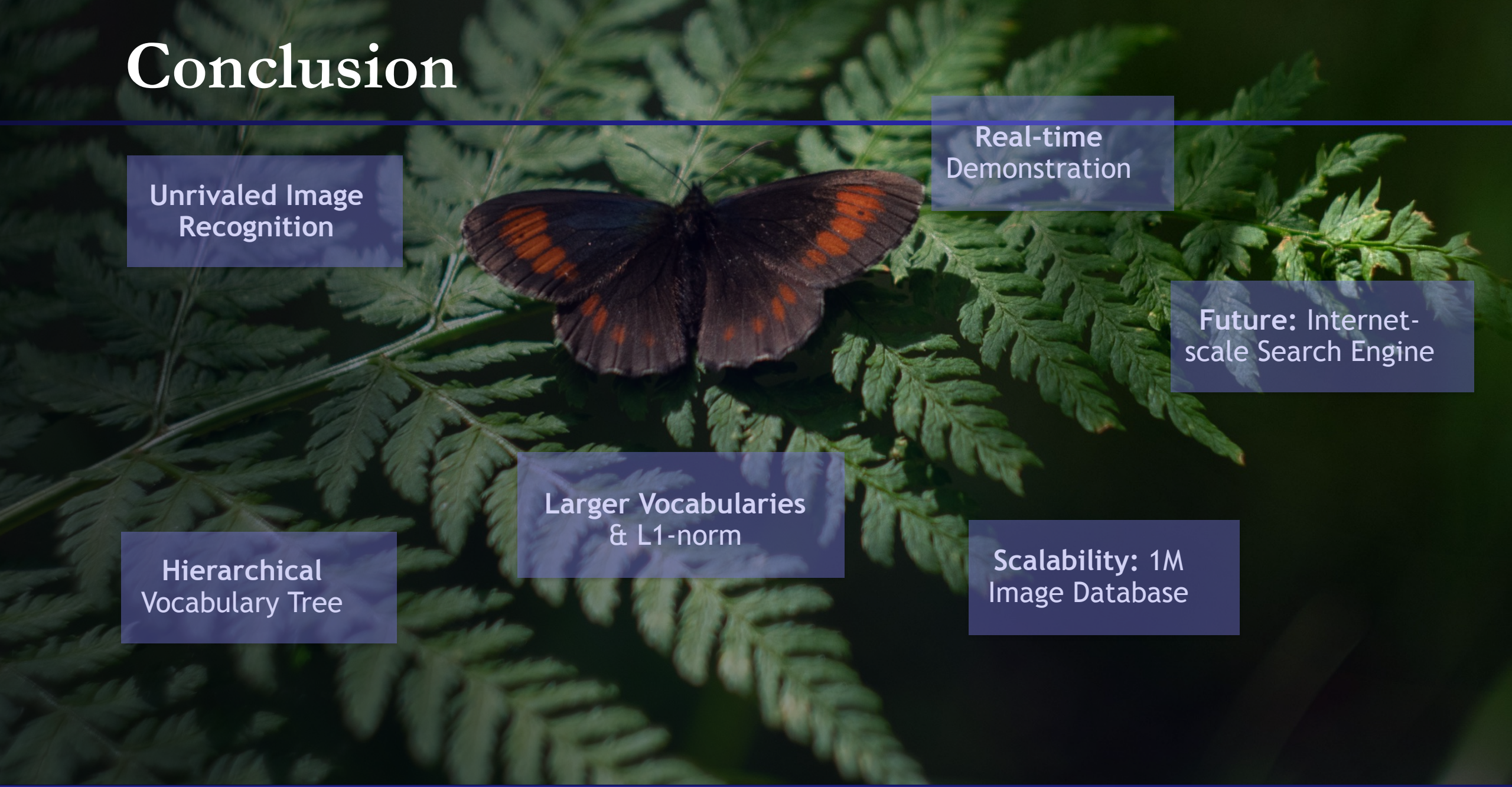
Top: Less reliable face search, performed on smaller 300K frame database. Both use images separate from the movies.

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

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.

Conclusion



Unrivaled Image Recognition

Real-time Demonstration

Future: Internet-scale Search Engine

Larger Vocabularies & L1-norm

Hierarchical Vocabulary Tree

Scalability: 1M Image Database

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].