Video Google: A Text Retrieval Approach to Object Matching in Videos

Josef Sivic and Andrew Zisserman
Goal

- Google search for videos
- Query is an portion of a frame of a video selected by the user
Google Text Search

- Web pages are parsed into words
- Words are replaced by their root word
- Stop list to filter common words
- Remaining words represent that web page as a vector weighted based on word frequency
Text Retrieval

• Efficient retrieval for with an index
• Text is retrieved by computing its vector of word frequencies, return documents with the closest vectors
• Consider order and location of words
Approach

- Apply text search properties to image search
Video Google: Descriptors

- Compute two types of covariant regions: Shape Adapted and Maximally Stable
- Regions computed in grayscale
Descriptors
Descriptors

• Each elliptical region is then represented by a SIFT descriptor
• Descriptor is averaged over the frames the region exists in
• Reduce noise: filter regions which do not exist in more than 3 frames
• Reject 10% of the regions with the largest diagonal covariance matrix
Build “Visual Words”

• Quantize the descriptors into visual words for text retrieval
• 1000 regions per frame and 128-vector descriptor
• Select 48 scenes containing 10,000 frames
• 200K descriptors
Clustering descriptors

• K-means clustering
• Run several times with random initial assignments
• $D(x_1, x_2) = \sqrt{(x_1 - x_2)^T \sum^{-1}(x_1 - x_2)}$
• MS and SA regions are clustered separately
Indexing using text retrieval methods

• Term frequency - inverse document frequency used for weighting the words of a document

\[ t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i} \]

• Retrieval: documents are ranked by their normalized scalar product between the query vector and all the document vectors
Image Retrieval

- Video google: The visual words of the query are the visual words in the user-specified portion of a frame.
- Search the index with the visual words to find all the frames which contain the same word.
- Rank all the results, return the most relevant results.
Stop List

- Visual words in the top 5% and bottom 10% are stopped
Spatial Consistency

• Google increases the ranking of documents where the query words appear close together in the searched text
• In video: 15 nearest neighbors defines search area
• Regions in this area by the query region vote on each match
• Re-ranked on the number of votes
Evaluation

• Tested on feature length movies with 100K - 150K frames
• Use one frame per second
• Ground truth determined by hand
• Retrieval performance measured by averaged rank of relevant images
Example
Questions?
Scalable Recognition with a Vocabulary Tree

David Nister and Henrik Stewenius
Vocabulary Tree

• Continuation of Video google
• 10,000 visual words in the database
• Offline crawling stage to index video takes 10 seconds per frame
Vocabulary Tree

• Too slow for a large database
• Larger databases result in better retrieval quality
• More words utilizes the power of the index: less database images must be considered
• On the fly insertion of new objects into the database
Training

• Training with hierarchical k-means
• More efficient than k-means
• 35,000 training frames instead of 400 with video google
Feature Extraction

- Maximally Stable regions used only
- Build SIFT descriptor from the region
Building Vocab Tree

• Hierarchical k-means, with k being the number of children nodes
• First run k-means to find k clusters
• Recursively apply to each cluster L times
• Visual words become the nodes
Performance

• Increasing the size of the vocabulary is logarithmic
• $K = 10$, $L = 6$: one million leaf nodes
Retrieval

• Determine the visual words from the query
• Propagate the region descriptor down the tree selecting the closest cluster at each level
Scoring

• Determine the relevance of a query image to a database image based on the similarity of their paths down the tree
• Use TD-IDF to assign weights to the query and database image vector
Scoring

• Use TD-IDF for weights of descriptor vectors
• Normalized relevance score:

\[ s(q, d) = \| \frac{q}{\|q\|} - \frac{d}{\|d\|} \| \]

• \( L_1 \)-normalization is the most effective
Results

• Tested on a ground truth database of 6,376 images
• Groups of four images of the same object
Results
A-1M Leaves L1
N-1M Leaves L2
Q-10K Non-hierarchical L1
R-10K Leaves L1
T-10K Non-hierarchical L2
U-10K Leaves L2
V-10K Non-hierarchical, Non-entropy L2
Results

• Tested on a database of 1 million images of CD covers
• Sub-second retrieval times for a database of a million images
• Performance increases with the number of leaf nodes
<table>
<thead>
<tr>
<th>Me</th>
<th>En</th>
<th>No</th>
<th>S%</th>
<th>Voc-Tree</th>
<th>Le</th>
<th>Eb</th>
<th>Perf</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>y/y</td>
<td>L1</td>
<td>0</td>
<td>6x10=1M</td>
<td>1</td>
<td>ir</td>
<td>90.6</td>
</tr>
<tr>
<td>B</td>
<td>y/y</td>
<td>L1</td>
<td>0</td>
<td>6x10=1M</td>
<td>1</td>
<td>vr</td>
<td>90.6</td>
</tr>
<tr>
<td>C</td>
<td>y/y</td>
<td>L1</td>
<td>0</td>
<td>6x10=1M</td>
<td>2</td>
<td>ir</td>
<td>90.4</td>
</tr>
<tr>
<td>D</td>
<td>n/y</td>
<td>L1</td>
<td>0</td>
<td>6x10=1M</td>
<td>2</td>
<td>ir</td>
<td>90.4</td>
</tr>
<tr>
<td>E</td>
<td>y/n</td>
<td>L1</td>
<td>0</td>
<td>6x10=1M</td>
<td>2</td>
<td>ir</td>
<td>90.4</td>
</tr>
<tr>
<td>F</td>
<td>n/n</td>
<td>L1</td>
<td>0</td>
<td>6x10=1M</td>
<td>2</td>
<td>ir</td>
<td>90.4</td>
</tr>
<tr>
<td>G</td>
<td>n/n</td>
<td>L1</td>
<td>0</td>
<td>6x10=1M</td>
<td>1</td>
<td>ir</td>
<td>90.2</td>
</tr>
<tr>
<td>H</td>
<td>y/y</td>
<td>L1</td>
<td>m2</td>
<td>6x10=1M</td>
<td>1</td>
<td>ir</td>
<td>90.0</td>
</tr>
<tr>
<td>I</td>
<td>y/y</td>
<td>L1</td>
<td>0</td>
<td>6x10=1M</td>
<td>3</td>
<td>ir</td>
<td>89.9</td>
</tr>
<tr>
<td>J</td>
<td>y/y</td>
<td>L1</td>
<td>0</td>
<td>6x10=1M</td>
<td>4</td>
<td>ir</td>
<td>89.9</td>
</tr>
<tr>
<td>K</td>
<td>y/y</td>
<td>L1</td>
<td>0</td>
<td>6x10=1M</td>
<td>2</td>
<td>vr</td>
<td>89.8</td>
</tr>
<tr>
<td>L</td>
<td>y/y</td>
<td>L1</td>
<td>0</td>
<td>6x10=1M</td>
<td>2</td>
<td>ip</td>
<td>89.0</td>
</tr>
<tr>
<td>M</td>
<td>y/y</td>
<td>L1</td>
<td>m5</td>
<td>6x10=1M</td>
<td>1</td>
<td>ir</td>
<td>89.1</td>
</tr>
<tr>
<td>N</td>
<td>y/y</td>
<td>L2</td>
<td>0</td>
<td>6x10=1M</td>
<td>1</td>
<td>ir</td>
<td>87.9</td>
</tr>
<tr>
<td>O</td>
<td>y/y</td>
<td>L2</td>
<td>0</td>
<td>6x10=1M</td>
<td>2</td>
<td>ir</td>
<td>86.6</td>
</tr>
<tr>
<td>P</td>
<td>y/y</td>
<td>L1</td>
<td>110</td>
<td>6x10=1M</td>
<td>2</td>
<td>ir</td>
<td>86.5</td>
</tr>
<tr>
<td>Q</td>
<td>y/y</td>
<td>L1</td>
<td>0</td>
<td>1x10K=10K</td>
<td>1</td>
<td>-</td>
<td>86.0</td>
</tr>
<tr>
<td>R</td>
<td>y/y</td>
<td>L1</td>
<td>0</td>
<td>4x10=10K</td>
<td>2</td>
<td>-</td>
<td>81.3</td>
</tr>
<tr>
<td>S</td>
<td>y/y</td>
<td>L1</td>
<td>0</td>
<td>4x10=10K</td>
<td>1</td>
<td>ir</td>
<td>80.9</td>
</tr>
<tr>
<td>T</td>
<td>y/y</td>
<td>L2</td>
<td>0</td>
<td>1x10K=10K</td>
<td>1</td>
<td>-</td>
<td>76.0</td>
</tr>
<tr>
<td>U</td>
<td>y/y</td>
<td>L2</td>
<td>0</td>
<td>4x10=10K</td>
<td>1</td>
<td>ir</td>
<td>74.4</td>
</tr>
<tr>
<td>V</td>
<td>y/y</td>
<td>L2</td>
<td>0</td>
<td>4x10=10K</td>
<td>2</td>
<td>ir</td>
<td>72.5</td>
</tr>
<tr>
<td>W</td>
<td>n/n</td>
<td>L2</td>
<td>0</td>
<td>1x10K=10K</td>
<td>1</td>
<td>-</td>
<td>70.1</td>
</tr>
</tbody>
</table>
Questions?