Content-based image and video analysis

Copy/Near Duplicate Detection

20.06.2011
Google Image Search

Content-based image and video retrieval
Google Image Search

Content-based image and video retrieval
What is a copy/near-duplicate?

Near-duplicate image:
An image is called a near-duplicate of a reference image if it is “close”, according to some defined measure, to the reference image.

Another definition:
An image that appears to a human observer to be identical or very similar.

Near duplicate image cases:
• Being perceptually identical
• Being images of the same 3D scene
Copy / Similarity

The Robustness issue:
Two videos which are copies
Source video: *Système deux*. C. Fayard 1975 (c)INA

The Discriminability issue:
Two similar videos which are not copies (different ties)
Applications

- Detection of copyright violations
- Detection of doubles in large image databases
  - Reduction of required disk space
- Grouping images in search results
- Commercial tracking
- Compression of video files
Sample application

Results for the query “flight of a bee” using Google Image Search

Dali

Greater diversity of images on the results first page
- Video Copy Detection: a Comparative Study

- Julien Law-To, Li Chen, Alexis Joly, Ivan Laptev, Olivier Buisson, Valerie Gouet-Brunet, Nozha Boujemaa, Fred Stentiford
Global Descriptors

- **Temporal**

\[ a(t) = \sum_{i=1}^{N} K(i)(I(i, t) - I(i, t - 1))^2 \]

- \( a(t) \): global temporal activity, \( I \): image, \( N \): number of pixels for each image, \( K(i) \): weight function to enhance the importance of the central pixels

- A signature is computed around each maxima of the temporal activity \( a(t) \)

- Spectral analysis by FFT leads to a 16-dimensional vector based on the phase of the activity.
Global Descriptors II

- **Ordinal Measurement**
  - Partition the image into \( N \) blocks
  - Sort the blocks using their average gray level
  - Signature \( S(t) \) uses the rank \( r_i \) of each block \( I \)

\[
S(t) = (r_1, r_2, \ldots, r_N)
\]

- The distance \( D(t) \) is defined for computing the similarity of two videos (a reference \( R \) and a candidate \( C \)) at a time \( t \) where \( T \) is the length of the considered segment.

\[
D(t) = \frac{1}{T} \sum_{i=t-T/2}^{t+T/2} |R(i) - C(i)|
\]
Global Descriptors III

- **Temporal Ordinal Measurement**
  - Rank the regions along the time
  - Each frame is divided into $K$ blocks, $\lambda^k$ is the rank of the region $k$ in a temporal window with the length $M$.
  - The distance $D$ between a query video $V_q$ and a reference video $V_r$ at the time $t$ is:
    \[
    D(V_q, V_r^p) = \frac{1}{K} \sum_{k=1}^{K} d^p(\lambda^k_q, \lambda^k_r)
    \]
    where,
    \[
    d^p(\lambda^k_q, \lambda^k_r) = \frac{1}{C_M} \sum_{i=1}^{M} |\lambda^k_q(i) - \lambda^k_r(p + i - 1)|
    \]
    $p$ is the temporal shift tested and $C_M$ is a normalizing factor. The best temporal shift $p$ is selected.
Local Descriptors: AJ

- Key-frame detection ($X, Y$ image dimensions)

$$a(t) = \sum_{x=1}^{X} \sum_{y=1}^{Y} \frac{|I(x, y, t + 1) - I(x, y, t)|}{X \times Y}$$

- An interest point detection
  - An improved version of the **Harris** interest point detector
- Computation of local differential descriptors around each interest point:

$$f = \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial x \partial y}, \frac{\partial^2 I}{\partial x^2}, \frac{\partial^2 I}{\partial y^2} \right)$$

- Computed at four different spatio-temporal positions distributed around the interest point making in total $5 \times 4 = 20$-dimensional fingerprint $F$
Local Descriptors: ViCopT

- **Harris** interest points are extracted on every frame.
- Signal description similar to the one used in **AJ** is computed.
- The local differential descriptors are extracted at four spatial positions around an interest point, leading to 20-dimensional signatures for each frame.
- The interest points are associated from frame to frame to build trajectories.
- For each trajectory, the signal description is the average of each component of the local description.
- A label is also assigned to local descriptions: Background & Motion
Local Descriptors: STIP

- STIP: Space Time Interest Points
- STIP correspond to points where the image values have significant local variation in both space and time

- STIP points are described by the spatio-temporal third order local jet leading to 34-dimensional vector.

\[ j = (L_x, L_y, L_t, L_{xx}, \ldots, L_{ttt}) \]
Evaluation

- BBC open news archives. 79 videos (~3.1 hours)

- Performance measures: Precision, recall, average precision
Single Transformations

Exact Copy  Contrast75  Contrast125  Crop  Blur

Letter-box  Insert  Zoom 1.2  Zoom 0.8
Outcomes

- For all transformations, **Temporal Ordinal Measure** presents excellent results: all the segments have been found with no false alarm.
- The **Ordinal Measure** presents poor results for zooming, cropping and letter-box transformations.
- Local descriptors based on Harris points of interest are not robust to a decrease of the contrast because the value of the corners can become too low.
Combined Transformations

(a) Source video: yachtclub (c)BBC

(b) Source video: sheepclone (c)BBC.

(c) Source video: manhattanaftermath (c)BBC.

Content-based image and video retrieval
Results

<table>
<thead>
<tr>
<th>Technique</th>
<th>AveP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ViCopT</td>
<td>0.86</td>
</tr>
<tr>
<td>AJ_SpatioTemp</td>
<td>0.79</td>
</tr>
<tr>
<td>AJ_Temp</td>
<td>0.68</td>
</tr>
<tr>
<td>Temp. Ord. Meas.</td>
<td>0.65</td>
</tr>
<tr>
<td>STIP</td>
<td>0.55</td>
</tr>
<tr>
<td>Temporal</td>
<td>0.51</td>
</tr>
<tr>
<td>Ord. Meas.</td>
<td>0.36</td>
</tr>
</tbody>
</table>

Content-based image and video retrieval
- Scalable Near Identical Image and Shot Detection
  O. Chum, J. Philbin, M. Isard, A. Zisserman

- Credit: Michael Weber
Tasks

- **Enumeration of duplicates in a corpus**
  - **Given**
    - Set of images
    - Query image
  - **Wanted**
    - Corpus with all duplicate images

- **Identification of a duplicate image**
  - **Given**
    - Set of images
    - Query image
  - **Wanted**
    - Pruned list from the top matches
Problems

- Partial occlusion
- Digitalization artifacts
- Compression artifacts
- Differing levels of compression
- Mild photometric distortions
- Image blur
## Copy Duplicate Detection Methods

<table>
<thead>
<tr>
<th>Color Histograms &amp; LSH</th>
<th>SIFT Features &amp; Min Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ Color Histograms</td>
<td>▪ SIFT Features</td>
</tr>
<tr>
<td>▪ Locality Sensitive Hashing</td>
<td>▪ Min Hash</td>
</tr>
</tbody>
</table>

### Requirements in large databases
- Small amount of data stored for each image
- Queries must be very fast

- Both presented methods achieve these requirements
Color Histograms & LSH

Color Histograms
- Used for image representation
  - Extremely compressed feature vector
  - Stored information size per image is 384 bytes

Locality Sensitive Hashing
- Used to find near duplicate histograms
  - Efficiently finds all points within a given radius with high probability
- Euclidean distance between feature vectors is used as measure of similarity
Color Histograms I

Opponent color model

- Simple to compute
- 3 channels I, O₁ and O₂

\[
I = \frac{(R + G + B)}{3}
\]

\[
O₁ = \frac{(R + G - 2B)}{4 + 0.5}
\]

\[
O₂ = \frac{(R - 2G + B)}{4 + 0.5}
\]
Color Histograms II

Spatial pyramid scheme

- Jointly encodes global and local information
- 3 levels of data – every level has 128 bytes
- The higher the level, the more local the data
- \( I \) has double the amount of data as \( O_1 \) and \( O_2 \) because more information is contained in the intensity information
- Each bin is represented by a single byte

<table>
<thead>
<tr>
<th>Level</th>
<th>( I )</th>
<th>( O_1 )</th>
<th>( O_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>64</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Locality Sensitive Hashing

- Hashes similar histograms into same bins
- Several hashing functions are used because of boundary effects

Index Building

- Builds a family of hashing functions
- Hashes every image vector from the database into the hash tables of all hash functions

Query Processing

- Hashes the query vector into every hash table
- Points in the same bins of the hash tables are the near duplicate images
Color Histograms & LSH Summary

- Creates a p-dimensional vector $v$ from the color histogram

- Each hashing function from locality sensitive hashing generates an integer hash value from $v$

- All images in the database have to be hashed this way into the hash tables

- For queries only the hash values have to be calculated and searched in the hash tables
SIFT Features & Min Hash

Image description
- SIFT features are used for image representation
  - Robust and insensitive to small local distortions
- Visual words are used because Min Hash was developed for text near duplicate detection

Min Hash
- Used to find images whose similarity is above a given threshold for a given query
- Used in this method because the feature vectors do not have to be as similar as in LSH
- Finds similar images in constant time
SIFT Features

- **Scale Invariant Feature Transform (SIFT)**
  - Transformation of image data into scale-invariant coordinates relative to local features
  - Insensitive to small local geometric and photometric image distortions

(a) Original image. (b) Image with SIFT features
Image Description

- Representation of feature regions by SIFT descriptors
- A visual vocabulary $V$ - a set of visual words is constructed quantizing the SIFT features with K-means
- Each K-means cluster center is a visual word
- Visual words are used to compare images
- Each image is represented as a set $A_i$ of words $A_i \subset V$
- Distance between two images: $sim (A_1, A_2) = \frac{|A_1 \cap A_2|}{|A_1 \cup A_2|}$
Min Hash

- Developed for text near duplicate detection
- Works with visual words
- Algorithm
  - Create random permutations $\pi$
  - For each document $A_i$ a min hash $\min \pi (A_i)$ is recorded
- Estimation of $\text{sim}(A_1, A_2)$:
  - $\text{sim}(A_1, A_2) = \ell/N$
  - $N$ is the number of independent permutations $\pi_j$
  - $\ell$ is the number of how many times $\min \pi_j (A_1) = \min \pi_j (A_2)$
Min Hash Example

Vocabulary $V = \{A, B, C, D, E, F\}$

Three sets $\{A, B, C\}$, $\{B, C, D\}$ and $\{A, E, F\}$

Estimated similarities

- $\{A, B, C\}$ with $\{B, C, D\}$: $3/4$
- $\{A, B, C\}$ with $\{A, E, F\}$: $1/4$
- $\{A, E, F\}$ with $\{B, C, D\}$: $0/4$
SIFT Features & Min Hash summary

- Each feature region is represented by a SIFT descriptor
- K-means quantizes the SIFT descriptors of features to sets of visual words
- Min hash efficiently finds near duplicate images of a query image in the data set
- Similarity is computed using a set overlap measure
Experiment I – data set

TRECVID 2006 database

- 165 hours (17.8M frames, 127 GB) MPEG-1 news footage from different TV stations from all over the world
- For image detection taken 146,588 frames
- Resolution is 352x240 pixels
- Data is not labeled
- Frames contain
  - Compression artifacts
  - Jitter
  - Noise
Experiment I

- True similarity set
  - Images that match the similarity definition:
    - Images whose histograms are within a given distance to the reference histogram
    - Set of images whose similarity is above a threshold

- Raw approximate similarity set
  - Images found by
    - Using the reference histogram for LSH query
    - Having at least one matching tuple of visual words
    - Includes many false positives

- Verified approximate similarity set
  - Filtered version of the raw approximate similarity set
Experiment I - results

- Used parameters for verification
  - Histogram method
    - Distance < 200
  - Min hash method
    - Similarity > 35%

- Random pairs
  - In 99.9% of cases
    - Histogram method
      - Distance > 500
    - Min hash method
      - Similarity < 5%
Experiment II

- Several selected sets of 30-40 near duplicate images from the TRECVID 2006 database

- Results
  - No false positives found (manual verification)
  - No exact knowledge about false negatives, but
  - Each method has a small number of false negatives compared with the other
Experiment II - results

Detected by both / Color Histograms & LSH only / SIFT Features & Min Hash only
Conclusions

Color Histograms & LSH
- Sensitive to occlusion
- Fairly insensitive to
  - Compression / digitalization artifacts
  - Noise
  - Image blur

SIFT Features & Min Hash
- Tolerates occlusion that preserve a sufficiently high percentage of visual words
- Sensitive to all deformations, increasing or decreasing the number of features like noise, image blur and strong artifacts
TRECVID 2008 Content-based Copy Detection (CBCD)

- A pilot task with synthetic queries
- Audio handled in a separate condition
- Task has both a detection and localization component
- Detection measure based on error rates
- Weighted trade-off of false alarms and misses
CBCD Task Overview

- **Goal:**
  - Build a benchmark collection for video copy detection methods

- **Task:**
  - Given a set of reference video collection and a set of 2000 queries,
  - determine for each query if it contains a copy of video from the reference collection
  - and if so, from where in the reference collection the copy comes

- Three main task types were derived: Video-only, audio-only, video + audio
Datasets and Queries

- **Dataset:**
  - Reference video collection: TV2007 and TV2008 sound & vision data (~200 hr)
  - Non-reference video collection: TV2007 BBC rushes data

- **Query types:**
  - Type 1: composed of a reference video only. (1/3)
  - Type 2: composed of a reference video embedded in a non-reference video. (1/3)
  - Type 3: composed of a non-reference video only. (1/3)

- **Number of queries:**
  - 201 total original queries were created
  - 67 queries for each type
Datasets and Queries

- After creating the queries, each was transformed:
  - 10 video transformations
  - 7 audio transformations
- Yielding:
  - $10 \times 201 = 2010$ video queries
  - $7 \times 201 = 1407$ audio queries
  - $10 \times 7 \times 201 = 14070$ audio+video queries
Video Transformations

- Cam Cording (T1)
- Picture in picture (T2)
- Insertions of pattern (T3)
- Strong re-encoding (T4)
- Change of gamma (T5)
- Decrease in quality (T6, T7) – by introducing a combination of *Blur, Gamma, Frame dropping, Contrast, Compression, Ratio, White noise*
  - For T6, 3 transformations are randomly selected and combined
  - For T7, 5 transformations are randomly selected and combined

Content-based image and video retrieval
Video Transformations

- Post production (T8, T9) – by introducing a combination of Crop, Shift, Contrast, Text insertion, Vertical mirroring, Insertion of pattern, Picture in picture,
  - For T8, 3 transformations are randomly selected and combined
  - For T9, 5 transformations are randomly selected and combined
- Combination of 5 randomly selected transformations chosen from T1-9 (T10)
Video Transformations Examples

Picture in Picture  Blur  Insertion of pattern  Strong re-encoding
Noise  Contrast  Change in gamma  Mirroring
Ratio  Crop  Shift  Text insertion

Content-based image and video retrieval
Sample Query Clips

Content-based image and video retrieval
Evaluation

- 22 participant teams
- 55 submitted runs (48 runs for video-only, 1 run for audio-only and 6 runs for mixed).

Criteria:
- How many queries they find the reference data or correctly tell there is none to find
- When a copy is detected, how accurately the run locates the reference data in the test data
- How much time is required for query processing
Measures

- Minimal Normalized Detection Cost Rate (NDCR): A cost-weighted combination of the probability of missing a true copy and the false alarm rate.

\[
DCR = C_{Miss} \times P_{Miss} \times R_{target} + C_{FA} \times R_{FA}
\]

\[
NDCR = \frac{DCR}{C_{Miss} \times R_{target}}
\]

- Copy rate \((R_{target})\): 0.5/hr, Cost of a miss \((C_{Miss})\): 10, Cost of a false alarm \((C_{FA})\): 1
- \(P_{Miss}\): Probability of a miss, \(R_{FA}\): False alarm rate

- Copy location accuracy: mean F1 (harmonic mean of Precision (P) and Recall (R): \(F1 = \frac{2 \times P \times R}{P + R}\))
- Copy detection time: mean processing time (s)
Top 10 Min. NDCR Performance

T1: Cam Cording  
T3: Insertion of patterns  
T4: Re-encoding  
T5: Change of gamma  
T6, T7: Decrease in quality  
T8, T9: Post Production  
T10: Random combination of 5 transformations

Content-based image and video retrieval
Top 10 F1 Performance

T1: Cam Cording  T3: Insertion of patterns  T5: Change of gamma  T8, T9: Post Production
T2: Pict. In Pict.  T4: Re-encoding  T6, T7: Decrease in quality  T10: Random combination of 5 transformations
### Top 10 sites per transformation (Min. NDCR)

<table>
<thead>
<tr>
<th>Site Name</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
<th>T6</th>
<th>T7</th>
<th>T8</th>
<th>T9</th>
<th>T10</th>
</tr>
</thead>
<tbody>
<tr>
<td>INRIA-LEAR.v.Strict</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>INRIA-LEAR.v.Soft</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>INRIA-IMEDIA.v.fusion</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>INRIA-LEAR.v.KeysAdv</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>-</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>INRIA-IMEDIA.v.joly</td>
<td>9</td>
<td>-</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>5</td>
</tr>
<tr>
<td>OrangeLabsICM.v.Run2</td>
<td>10</td>
<td>6</td>
<td>6</td>
<td>-</td>
<td>-</td>
<td>9</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>OrangeLabsICM.v.Run1</td>
<td>-</td>
<td>5</td>
<td>5</td>
<td>-</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>OrangeLabsICM.v.Run3</td>
<td>-</td>
<td>3</td>
<td>7</td>
<td>-</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>MCG-ICT-CAS.v.ICTCVDREL</td>
<td>-</td>
<td>9</td>
<td>-</td>
<td>9</td>
<td>-</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>-</td>
<td>10</td>
</tr>
<tr>
<td>INRIA-IMEDIA.v.ViCopT</td>
<td>6</td>
<td>-</td>
<td>9</td>
<td>-</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>8</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>MCG-ICT-CAS.v.ICTCVDALL</td>
<td>-</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>CRIMontreal.v.Run1</td>
<td>5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>8</td>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>thu-intel.v.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>5</td>
<td>8</td>
<td>9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>thu-intel.v.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>6</td>
<td>6</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BeijingUPT.v.run1</td>
<td>-</td>
<td>-</td>
<td>10</td>
<td>8</td>
<td>-</td>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CRIMontreal.v.Run2</td>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>9</td>
<td>-</td>
<td>9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CRIMontreal.v.Run2Faster</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ColumbiaU.v.baseLocal</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>9</td>
<td>-</td>
</tr>
<tr>
<td>thu-intel.v.1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MCG-ICT-CAS.v.ICTCVDTOA</td>
<td>-</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Content-based image and video retrieval
INRIA-LEAR’s Video Copy Detection System

Matthijs Douze, Adrien Gaidon, Herve Jegou, Marcin Marszalek, Cordelia Schmid
Large scale object/scene recognition

- Each image described by approximately 2000 descriptors
  - $2 \times 10^9$ descriptors to index!

- Database representation in RAM:
  - Raw size of descriptors: 1 TB, search+memory intractable
State-of-the-art: Bag-of-words [Sivic & Zisserman’03]

“visual words”:
- 1 “word” (index) per local descriptor
- only images ids in inverted file
=> 8 GB fits!
Two Main Contributions

- Hamming Embedding
- Weak Geometry Consistency
Overview of INRIA-LEAR Copyright Detection System

Content-based image and video retrieval
Frame Extraction

- **Uniform subsampling:**
  - A fixed number of frames per time unit is extracted (2.5 frames per second). Used in STRICT and SOFT runs.

- **Stable keyframes:**
  - Only a few representative keyframes per shot is extracted (1 frame every 6 seconds on average).
  - In the preliminary experiments, it is observed that the stable keyframe selection caused an insufficient number of frame matches, therefore for the KEYSADVES run, an asymmetric sampling strategy is used:
    - Stable keyframes are extracted on the dataset side.
    - The query frames are extracted using uniform subsampling.
Feature Extraction

Keyframe

Set of descriptors

Compute local descriptors on regions of interest

② Hessian-Affine detector

③ SIFT descriptor

Set of descriptors

Quantization

\[ x \rightarrow q(x) \]

Hamming Embedding

\[ x \rightarrow b(x) \]

Interest region: quantized scales and angles

Compact descriptors: Bag-of-features-like representation

Content-based image and video retrieval
First Issue with Bag-of-Features Representation

- The intrinsic matching scheme performed by BOF is weak
  - for a “small” visual dictionary: too many false matches
  - for a “large” visual dictionary: many true matches are missed

- No good trade-off between “small” and “large”!
  - either the cells are too big
  - or these cells can’t absorb the descriptor noise
20K visual word: false matches
200K visual word: good matches missed
Hamming Embedding

Representation of a descriptor $x$

- Vector-quantized to $q(x)$ as in standard BOF
- Short binary vector $b(x)$ for an additional localization in the cell

Two descriptors $x$ and $y$ match iff

$$q(x) = q(y)$$
$$h(b(x); b(y)) < t$$

where $h(a,b)$ is the Hamming distance, $t$ is a threshold
Hamming Embedding

- **Off-line** (given a quantizer)
  - draw an orthogonal projection matrix $P$ of size $d_b \times d$
  - this defines $d_b$ random projection directions
  - for each cell and projection direction, compute the median value for a learning set

- **On-line**: compute the binary signature $b(x)$ of a given descriptor
  - project $x$ onto the projection directions as $z(x) = (z_1, \ldots, z_{db})$
  - $b_i(x) = 1$ if $z_i(x)$ is above the learned median value, otherwise 0
Hamming Embedding: Example
Compared with 20K dictionary: false matches
Hamming Embedding: Example
Compared with 200K visual word: good matches missed
Second Issue with Bag-of-Features Representation

- Re-ranking based on full geometric verification
  - works very well
  - but performed on a short-list only (typically, 100 images)
  - for very large datasets, the number of distracting images is so high that relevant images are not even short-listed!
Weak Geometry Consistency

- Weak geometric information used for **all** images (not only the short-list)
- Each invariant interest region detection has a scale and rotation angle associated, here characteristic scale and dominant gradient orientation

Each matching pair results in a scale and angle difference

For the global image scale and rotation changes are roughly consistent
WGC: Orientation consistency

Max = rotation angle between images
WGC: Scale consistency
Weak Geometry Consistency

- Integrate the geometric verification into the BOF representation
- Only matches that do agree with the main difference of orientation and scale will be taken into account in the final score
- Re-ranking using full geometric transformation still adds information in a final stage
Experimental results

- Evaluation for the INRIA holidays dataset, 1491 images
  - 500 query images + 991 annotated true positives
  - Most images are holiday photos of friends and family
- 1 million distractor images from Flickr
- Dataset size 1,001,491 images
- Vocabulary construction on a different Flickr set
- Almost real-time search speed,
- Evaluation metric: mean average precision (in [0,1], bigger = better)
  - Average over precision/recall curve
Holiday dataset – example queries
Dataset: Venice Channel
Dataset: San Marco square

Query

Base 1

Base 2

Base 3

Base 4

Base 5

Base 6

Base 7

Base 8

Base 9
Example distractors - Flickr
Comparison with state-of-the-art

- Evaluation on the holidays dataset, 500 query images, 1 million distracter images
- Metric: mean average precision

![Graph showing comparison with state-of-the-art]

<table>
<thead>
<tr>
<th>Average query time (4 CPU cores)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute descriptors</td>
</tr>
<tr>
<td>Quantization</td>
</tr>
<tr>
<td>Search – baseline</td>
</tr>
<tr>
<td>Search – WGC</td>
</tr>
<tr>
<td>Search – HE</td>
</tr>
<tr>
<td>Search – HE+WGC</td>
</tr>
</tbody>
</table>
Results – Venice Channel
Results – San Marco

Query

Base 01

Base 02

Base 03

Base 06

Flickr

Flickr
References

- D. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 60(2):91-110, 2004

- TRECVID. http://trecvid.nist.gov