Image Retrieval for Image-Based Localization Revisited

Torsten Sattler\textsuperscript{1} Tobias Weyand\textsuperscript{2}
Bastian Leibe\textsuperscript{2} Leif Kobbelt\textsuperscript{1}

\textsuperscript{1}Computer Graphics Group, RWTH Aachen University
\textsuperscript{2}Computer Vision Group, RWTH Aachen University
Determine **position & orientation** of query image
Image-Based Localization

Determine **position & orientation** of query image
Image-Based Localization

Determine position & orientation of query image
Image-Based Localization

Determine **position & orientation** of query image

2D-to-3D correspondences
Image-Based Localization

• Structure-from-Motion point cloud
  • associate image descriptors with 3D points
  → descriptor matching problem
Image-Based Localization

- Structure-from-Motion point cloud
- associate image descriptors with 3D points
  ➔ descriptor matching problem

<table>
<thead>
<tr>
<th></th>
<th>Scalability</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image retrieval</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>
Image-Based Localization

- Structure-from-Motion point cloud
- associate image descriptors with 3D points
  ➡ descriptor matching problem

<table>
<thead>
<tr>
<th></th>
<th>Scalability</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image retrieval</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Direct matching</td>
<td>×</td>
<td>✓</td>
</tr>
</tbody>
</table>
Image-Based Localization

• Structure-from-Motion point cloud
• associate image descriptors with 3D points

.descriptor matching problem

<table>
<thead>
<tr>
<th></th>
<th>Scalability</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image retrieval</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Direct matching</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>
Overview

- Image Retrieval & Direct Matching
- Image Retrieval Revisited
- Efficient Correspondence Selection
Irschara, Zach, Frahm, Bischof. *From Structure-from-Motion Point Clouds to Fast Location Recognition*. CVPR’09
Inverted file entries correspond to 3D points
Inverted file entries correspond to 3D points

Irschara, Zach, Frahm, Bischof. *From Structure-from-Motion Point Clouds to Fast Location Recognition.* CVPR’09
Irschara, Zach, Frahm, Bischof. *From Structure-from-Motion Point Clouds to Fast Location Recognition*. CVPR’09

Inverted file entries correspond to 3D points
Irschara, Zach, Frahm, Bischof. *From Structure-from-Motion Point Clouds to Fast Location Recognition*. CVPR'09

Inverted file entries correspond to 3D points
Irschara, Zach, Frahm, Bischof. *From Structure-from-Motion Point Clouds to Fast Location Recognition*. CVPR’09

**Diagram:**
- **Query image** (Q) is compared to **visual words**.
- **Inverted file** entries correspond to 3D points in the **database images** (A, B, C, ...).

**Text:**
Inverted file entries correspond to 3D points.
Inverted file entries correspond to 3D points
Inverted file entries correspond to 3D points
Irschara, Zach, Frahm, Bischof. *From Structure-from-Motion Point Clouds to Fast Location Recognition*. CVPR’09

Inverted file entries correspond to 3D points
Inverted file entries correspond to 3D points

Irschara, Zach, Frahm, Bischof. *From Structure-from-Motion Point Clouds to Fast Location Recognition*. CVPR’09
Irschara, Zach, Frahm, Bischof. *From Structure-from-Motion Point Clouds to Fast Location Recognition*. CVPR’09

Inverted file entries correspond to 3D points
Query image

Visual words

Inverted file

Scoring

Ranked images

Top-k

Feature matching

Pose estimation: RANSAC + n-point-pose

3D point cloud

Inverted file entries correspond to 3D points

Irschara, Zach, Frahm, Bischof. *From Structure-from-Motion Point Clouds to Fast Location Recognition*. CVPR’09
Irschara, Zach, Frahm, Bischof. *From Structure-from-Motion Point Clouds to Fast Location Recognition*. CVPR’09

**Inverted file entries correspond to 3D points**

**Pose Estimation:** RANSAC + n-point-pose
Image Retrieval for Localization

Irschara, Zach, Frahm, Bischof. *From Structure-from-Motion Point Clouds to Fast Location Recognition*. CVPR’09

Inverted file entries correspond to 3D points
Choose pose with most inliers as final pose
Direct Matching

Sattler, Leibe, Kobbelt. *Fast Image-Based Localization using Direct 2D-to-3D Matching*. ICCV’11

100k visual words

3D point cloud

assign descriptors of points to words *(offline)*
Sattler, Leibe, Kobbelt. *Fast Image-Based Localization using Direct 2D-to-3D Matching*. ICCV’11

**Direct Matching**

query image

100k visual words

assign descriptors of points to words (**offline**)
Direct Matching

Sattler, Leibe, Kobbelt. *Fast Image-Based Localization using Direct 2D-to-3D Matching*. ICCV’11

query image

100k visual words

3D point cloud

assign descriptors of points to words (offline)
Direct Matching

Sattler, Leibe, Kobbelt. *Fast Image-Based Localization using Direct 2D-to-3D Matching*. ICCV’11

query image

100k visual words

find nearest neighbors

assign descriptors of points to words *(offline)*

3D point cloud

Q

f

q

p

d_{f,q}

d_{f,p}

f

Image Retrieval for Image-Based Localization Revisited
Torsten Sattler
Direct Matching

Sattler, Leibe, Kobbelt. *Fast Image-Based Localization using Direct 2D-to-3D Matching*. ICCV’11

Image Retrieval for Image-Based Localization Revisited
Torsten Sattler

Establish match \( f \leftrightarrow p \) if

\[
\frac{d_{f,p}}{d_{f,q}} < 0.6
\]
Sattler, Leibe, Kobbelt. *Fast Image-Based Localization using Direct 2D-to-3D Matching*. ICCV’11

**Direct Matching**

- Query image
- 100k visual words
- 3D point cloud
- Assign descriptors of points to words (*offline*)
- Establish match if \( \frac{d_{f,p}}{d_{f,q}} < 0.6 \)
- Pose Estimation: RANSAC + n-point-pose
## The Performance Gap

### Scalability

<table>
<thead>
<tr>
<th></th>
<th>Inverted file entry size</th>
<th>Run time cost / entry</th>
<th>Registration Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image retrieval</td>
<td>image id (4 bytes)</td>
<td>vote for image</td>
<td>6-18% less images</td>
</tr>
<tr>
<td>Direct matching</td>
<td>SIFT descriptor (128 bytes)</td>
<td>descriptor distance computation</td>
<td>state-of-the-art</td>
</tr>
</tbody>
</table>

- **Image retrieval**: The image id (4 bytes) for each entry results in a lower run time cost per entry compared to direct matching. It achieves 6-18% less images in registration performance.
- **Direct matching**: Using a SIFT descriptor (128 bytes) for each entry increases the run time cost per entry but maintains state-of-the-art registration performance.
### The Performance Gap

<table>
<thead>
<tr>
<th></th>
<th>Scalability</th>
<th>Registration Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inverted file entry size</td>
<td>Run time cost / entry</td>
</tr>
<tr>
<td><strong>Image retrieval</strong></td>
<td>image id (4 bytes)</td>
<td>vote for image</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6-18% less images</td>
</tr>
<tr>
<td><strong>Direct matching</strong></td>
<td>SIFT descriptor (128 bytes)</td>
<td>descriptor distance computation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>state-of-the-art</td>
</tr>
</tbody>
</table>
## The Performance Gap

<table>
<thead>
<tr>
<th>Scalability</th>
<th>Registration Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inverted file entry size</td>
<td>Run time cost / entry</td>
</tr>
<tr>
<td>Image retrieval</td>
<td>image id (4 bytes)</td>
</tr>
<tr>
<td>Direct matching</td>
<td>SIFT descriptor (128 bytes)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Performance Gap

<table>
<thead>
<tr>
<th>Scalability</th>
<th>Registration Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image retrieval</strong></td>
<td></td>
</tr>
<tr>
<td>Inverted file entry size</td>
<td>vote for image</td>
</tr>
<tr>
<td>image id (4 bytes)</td>
<td></td>
</tr>
<tr>
<td>Run time cost / entry</td>
<td></td>
</tr>
<tr>
<td><strong>Direct matching</strong></td>
<td></td>
</tr>
<tr>
<td>SIFT descriptor (128 bytes)</td>
<td>descriptor distance computation</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Performance gap caused by **failure to rank any relevant image high enough**
query image

visual word $w$
query image

visual word $w$
Image Retrieval Revisited

query image

visual word $w$

correct votes:
descriptors from corresponding 3D point
Image Retrieval Revisited

query image

visual word $w$

incorrect votes:
descriptors from other 3D points

correct votes:
descriptors from corresponding 3D point

8
Image Retrieval for Image-Based Localization Revisited

- Query image
- Visual word $w$
- Incorrect votes: descriptors from other 3D points
- Correct votes: descriptors from corresponding 3D point

Selective Voting
Correspondence Voting

Idea: Find corresponding 3D point

query image

visual word \( w \)

image database

...
Correspondence Voting

Idea: Find corresponding 3D point
Correspondence Voting

Idea: Find corresponding 3D point

find 2 nearest neighbors

visual word $w$

$d_1$, $d_2$

query image

image database

...
Correspondence Voting

Idea: Find corresponding 3D point

vote only if \( \frac{d_1}{d_2} < 0.6 \)

find 2 nearest neighbors
Correspondence Voting

Idea: Find corresponding 3D point

Vote only if $\frac{d_1}{d_2} < 0.6$

Find 2 nearest neighbors

$\star$ indicates a match.
Experimental Evaluation

Aachen

Vienna

dataset available at
http://www.graphics.rwth-aachen.de/localization

dataset kindly provided by
A. Irschara [Irschara, CVPR’09]
used in [Irschara, CVPR’09], [Li, ECCV’10], [Sattler, ICCV’11]

<table>
<thead>
<tr>
<th>Dataset</th>
<th># 3D points</th>
<th># db images</th>
<th># query images</th>
<th>mean # features per query</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aachen</td>
<td>1.54M</td>
<td>3047</td>
<td>369</td>
<td>9707.29</td>
</tr>
<tr>
<td>Vienna</td>
<td>1.12M</td>
<td>1324</td>
<td>266</td>
<td>8648.66</td>
</tr>
</tbody>
</table>
Registration Performance

registered@k – Aachen

registered@k – Vienna

Direct matching
[Sattler, ICCV'11]
100k words

tf*idf Weighting
[Sivic, ICCV'03]
100k words 1M words

image retrieval-based
Registration Performance

registered@k – Aachen

registered@k – Vienna

Direct matching
[Sattler, ICCV’11]
100k words

tf*idf Weighting
[Sivic, ICCV’03]
100k words
1M words

Probabilistic Scoring
[Irschara, CVPR’09]
100k words
1M words

image retrieval-based
Registration Performance

registered@k – Aachen

registered@k – Vienna

Direct matching
[Sattler, ICCV’11]
100k words

$\text{tf} \cdot \text{idf}$ Weighting
[Sivic, ICCV’03]
100k words 1M words

Probabilistic Scoring
[Irschara, CVPR’09]
100k words 1M words

Correspondence Voting
100k words

Image Retrieval for Image-Based Localization Revisited
Torsten Sattler
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>Scalability</th>
<th>Registration Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inverted file entry size</td>
<td>Run time cost / entry</td>
</tr>
<tr>
<td>Image retrieval</td>
<td>image id (4 bytes)</td>
<td>vote for image</td>
</tr>
<tr>
<td>Direct matching</td>
<td>SIFT descriptor (128 bytes)</td>
<td>descriptor distance</td>
</tr>
<tr>
<td>Voting</td>
<td>SIFT descriptor (128 bytes)</td>
<td>descriptor distance</td>
</tr>
<tr>
<td></td>
<td>Scalability</td>
<td>Registration Performance</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td>Inverted file entry size</td>
<td>Run time cost / entry</td>
</tr>
<tr>
<td>Image retrieval</td>
<td>image id (4 bytes)</td>
<td>vote for image</td>
</tr>
<tr>
<td>Direct matching</td>
<td>SIFT descriptor (128 bytes)</td>
<td>descriptor distance computation</td>
</tr>
<tr>
<td>Correspondence Voting</td>
<td>SIFT descriptor (128 bytes)</td>
<td>descriptor distance computation</td>
</tr>
</tbody>
</table>
Hamming Voting

Jégou, Douze, Schmid. **Hamming Embedding** and Weak Geometric consistency for large-scale image search. ECCV’08

- Random projection: $\mathbb{R}^{128} \rightarrow \mathbb{R}^d$

- Thresholding per visual word: $\mathbb{R}^d \rightarrow \{0, 1\}^d$

\[ \begin{align*}
\text{visual word } w
\end{align*} \]
Hamming Voting

Jégou, Douze, Schmid. **Hamming Embedding** and Weak Geometric consistency for large-scale image search. ECCV’08

- Random projection: $\mathbb{R}^{128} \rightarrow \mathbb{R}^d$
- Thresholding per visual word: $\mathbb{R}^d \rightarrow \{0, 1\}^d$

![Diagram]

visual word $w$
Hamming Voting

Jégou, Douze, Schmid. **Hamming Embedding** and Weak Geometric consistency for large-scale image search. ECCV’08

- Random projection: \( \mathbb{R}^{128} \rightarrow \mathbb{R}^{d} \)

- Thresholding per visual word: \( \mathbb{R}^{d} \rightarrow \{0, 1\}^{d} \)
Hamming Voting

registered@10 – Aachen

registered@10 – Vienna

Correspondence
Voting
100k words

8 bits
100k 1M

Hamming Voting
Hamming Voting

registered@10 – Aachen

registered@10 – Vienna

% registered images

Hamming Distance Threshold

Correspondence Voting
100k words

8 bits
100k 1M

16 bits
100k 1M

Hamming Voting

Image Retrieval for Image-Based Localization Revisited
Torsten Sattler
Hamming Voting

registered@10 – Aachen

registered@10 – Vienna

Hamming Distance Threshold

Correspondence Voting
100k words

8 bits
100k 1M 100k 1M

16 bits
100k 1M 100k 1M

32 bits
100k 1M 100k 1M

Hamming Voting

Image Retrieval for Image-Based Localization Revisited
Torsten Sattler
Hamming Voting

registered@10 – Aachen

registered@10 – Vienna

Correspondence Voting
100k words

<table>
<thead>
<tr>
<th></th>
<th>8 bits</th>
<th>16 bits</th>
<th>32 bits</th>
<th>64 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>100k words</td>
<td>100k</td>
<td>1M</td>
<td>100k</td>
<td>1M</td>
</tr>
</tbody>
</table>

Hamming Voting
### Comparison

<table>
<thead>
<tr>
<th></th>
<th>Scalability</th>
<th>Registration Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inverted file entry size</td>
<td>Run time cost / entry</td>
</tr>
<tr>
<td>Image retrieval</td>
<td>image id (4 bytes)</td>
<td>vote for image</td>
</tr>
<tr>
<td>Direct matching</td>
<td>SIFT descriptor (128 bytes)</td>
<td>descriptor distance computation</td>
</tr>
<tr>
<td>Correspondence Voting</td>
<td>SIFT descriptor (128 bytes)</td>
<td>descriptor distance computation</td>
</tr>
<tr>
<td>Hamming Voting (64 bits)</td>
<td>binary descriptor (8 bytes)</td>
<td>Hamming distance computation (10^6 computations ≈ 2ms) + vote</td>
</tr>
</tbody>
</table>
## Comparison

<table>
<thead>
<tr>
<th><strong>Scalability</strong></th>
<th><strong>Registration Performance</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inverted file entry size</td>
</tr>
<tr>
<td>Image retrieval</td>
<td>image id (4 bytes)</td>
</tr>
<tr>
<td>Direct matching</td>
<td>SIFT descriptor (128 bytes)</td>
</tr>
<tr>
<td>Correspondence Voting</td>
<td>SIFT descriptor (128 bytes)</td>
</tr>
<tr>
<td>Hamming Voting (64 bits)</td>
<td>binary descriptor (8 bytes)</td>
</tr>
</tbody>
</table>

Additional cost for Hamming Voting: \(+ \sim 23\text{ms per query image}\) (projection, thresholding)
Correspondence Selection

- Run time cost: Voting + **Regular SIFT matching**
  - Build kd-tree for query features
  - Match database features against kd-tree
  - Introduces additional computations

Pose Estimation: RANSAC + n-point-pose
Correspondence Selection

- Run time cost: Voting + **Regular SIFT matching**
  - Build kd-tree for query features
  - Match database features against kd-tree
  - Introduces additional computations

Pose Estimation: RANSAC + n-point-pose
Correspondence Selection

• Idea: Re-use matches from voting stage
• Problem: Not enough correspondences
Correspondence Selection

- Idea: Re-use matches from voting stage
- Problem: Not enough correspondences

Quantized Matching: Restrict search to visual word [Sattler, ICCV’11]
Correspondence Selection

- Idea: Re-use matches from voting stage
- Problem: Not enough correspondences

Quantized Matching: Restrict search to visual word [Sattler, ICCV’11]
Correspondence Selection

- Idea: Re-use matches from voting stage
- Problem: Not enough correspondences

Quantized Matching: Restrict search to visual word [Sattler, ICCV’11]

Coarser vocabulary from hierarchical clustering, no additional assignment costs
Quantized Matching: Restrict nearest neighbor search to same visual word

<table>
<thead>
<tr>
<th>Matching Method</th>
<th>Voc. Size</th>
<th># Images Registered</th>
<th>Correspondence Search [ms]</th>
<th>RANSAC ok [ms]</th>
<th>err [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular SIFT</td>
<td>-</td>
<td>320 (87%)</td>
<td>300.3</td>
<td>0.9</td>
<td>0.0</td>
</tr>
<tr>
<td>Quantized SIFT</td>
<td>100</td>
<td>319 (86%)</td>
<td>14.5</td>
<td>3.1</td>
<td>155.3</td>
</tr>
<tr>
<td>Quantized Hamming (64-bit)</td>
<td>100</td>
<td>307 (83%)</td>
<td>3.6</td>
<td>141.6</td>
<td>2825.0</td>
</tr>
</tbody>
</table>

median timings per query image - database image pair
**Quantized Matching**: Restrict nearest neighbor search to same visual word

<table>
<thead>
<tr>
<th>Matching Method</th>
<th>Voc. Size</th>
<th># Images Registered</th>
<th>Correspondence Search [ms]</th>
<th>RANSAC ok [ms]</th>
<th>RANSAC err [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular SIFT</td>
<td>-</td>
<td>320 (87%)</td>
<td>300.3</td>
<td>0.9</td>
<td>0.0</td>
</tr>
<tr>
<td>Quantized SIFT</td>
<td>100</td>
<td>319 (86%)</td>
<td>14.5</td>
<td>3.1</td>
<td>155.3</td>
</tr>
<tr>
<td>Quantized Hamming (64-bit)</td>
<td>100</td>
<td>307 (83%)</td>
<td>3.6</td>
<td>141.6</td>
<td>2825.0</td>
</tr>
</tbody>
</table>

Median timings per query image - database image pair
### Quantized Matching: Restrict nearest neighbor search to same visual word

<table>
<thead>
<tr>
<th>Matching Method</th>
<th>Voc. Size</th>
<th># Images Registered</th>
<th>Correspondence Search [ms]</th>
<th>RANSAC ok [ms]</th>
<th>err [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular SIFT</td>
<td>-</td>
<td>320 (87%)</td>
<td>300.3</td>
<td>0.9</td>
<td>0.0</td>
</tr>
<tr>
<td>Quantized SIFT</td>
<td>100</td>
<td>319 (86%)</td>
<td>14.5</td>
<td>3.1</td>
<td>155.3</td>
</tr>
<tr>
<td>Quantized Hamming (64-bit)</td>
<td>100</td>
<td>307 (83%)</td>
<td>3.6</td>
<td>141.6</td>
<td>2825.0</td>
</tr>
</tbody>
</table>

Median timings per query image - database image pair
Quantized Matching: Restrict nearest neighbor search to same visual word

<table>
<thead>
<tr>
<th>Matching Method</th>
<th>Voc. Size</th>
<th># Images Registered</th>
<th>Correspondence Search [ms]</th>
<th>RANSAC ok [ms]</th>
<th>err [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular SIFT</td>
<td>-</td>
<td>320 (87%)</td>
<td>300.3</td>
<td>0.9</td>
<td>0.0</td>
</tr>
<tr>
<td>Quantized SIFT</td>
<td>100</td>
<td>319 (86%)</td>
<td>14.5</td>
<td>3.1</td>
<td>155.3</td>
</tr>
<tr>
<td>Quantized Hamming (64-bit)</td>
<td>100</td>
<td>307 (83%)</td>
<td>3.6</td>
<td>141.6</td>
<td>2825.0</td>
</tr>
</tbody>
</table>

median timings per query image - database image pair
## Correspondence Selection

<table>
<thead>
<tr>
<th>Matching Method</th>
<th>Voc. Size</th>
<th># Images Registered</th>
<th>RANSAC ok [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular SIFT</td>
<td>-</td>
<td>320 (87%)</td>
<td>0.9</td>
</tr>
<tr>
<td>Quantized SIFT</td>
<td>100</td>
<td>319 (86%)</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>1k</td>
<td>304 (82%)</td>
<td>17.4</td>
</tr>
<tr>
<td></td>
<td>10k</td>
<td>246 (67%)</td>
<td>10.2</td>
</tr>
<tr>
<td>Quantized Hamming (64-bit)</td>
<td>100</td>
<td>307 (83%)</td>
<td>141.6</td>
</tr>
<tr>
<td></td>
<td>1k</td>
<td>300 (81%)</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>10k</td>
<td>272 (74%)</td>
<td>0.9</td>
</tr>
</tbody>
</table>
### Correspondence Selection

<table>
<thead>
<tr>
<th>Matching Method</th>
<th>Voc. Size</th>
<th># Images Registered</th>
<th>RANSAC ok [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular SIFT</td>
<td>-</td>
<td>320 (87%)</td>
<td>0.9</td>
</tr>
<tr>
<td>Quantized SIFT</td>
<td>100</td>
<td>319 (86%)</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>1k</td>
<td>304 (82%)</td>
<td>17.4</td>
</tr>
<tr>
<td></td>
<td>10k</td>
<td>246 (67%)</td>
<td>10.2</td>
</tr>
<tr>
<td>Quantized Hamming</td>
<td>100</td>
<td>307 (83%)</td>
<td>141.6</td>
</tr>
<tr>
<td>(64-bit)</td>
<td>1k</td>
<td>300 (81%)</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>10k</td>
<td>272 (74%)</td>
<td>0.9</td>
</tr>
</tbody>
</table>
## Correspondence Selection

<table>
<thead>
<tr>
<th>Matching Method</th>
<th>Voc. Size</th>
<th># Images Registered</th>
<th>RANSAC ok [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular SIFT</td>
<td>-</td>
<td>320 (87%)</td>
<td>0.9</td>
</tr>
<tr>
<td>Quantized SIFT</td>
<td>100</td>
<td>319 (86%)</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>1k</td>
<td>304 (82%)</td>
<td>17.4</td>
</tr>
<tr>
<td></td>
<td>10k</td>
<td>246 (67%)</td>
<td>10.2</td>
</tr>
<tr>
<td>Quantized Hamming (64-bit)</td>
<td>100</td>
<td>307 (83%)</td>
<td>141.6</td>
</tr>
<tr>
<td></td>
<td>1k</td>
<td>300 (81%)</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>10k</td>
<td>272 (74%)</td>
<td>0.9</td>
</tr>
</tbody>
</table>
### Correspondence Selection

<table>
<thead>
<tr>
<th>Matching Method</th>
<th>Voc. Size</th>
<th># Images Registered</th>
<th>RANSAC ok [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular SIFT</td>
<td>-</td>
<td>320 (87%)</td>
<td>0.9</td>
</tr>
<tr>
<td>Quantized SIFT</td>
<td>100</td>
<td>319 (86%)</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>1k</td>
<td>304 (82%)</td>
<td>17.4</td>
</tr>
<tr>
<td></td>
<td>10k</td>
<td>246 (67%)</td>
<td>10.2</td>
</tr>
<tr>
<td>Quantized Hamming</td>
<td>100</td>
<td>307 (83%)</td>
<td>141.6</td>
</tr>
<tr>
<td>(64-bit)</td>
<td>1k</td>
<td>300 (81%)</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>10k</td>
<td>272 (74%)</td>
<td>0.9</td>
</tr>
</tbody>
</table>
### Correspondence Selection

<table>
<thead>
<tr>
<th>Matching Method</th>
<th>Voc. Size</th>
<th># Images Registered</th>
<th>RANSAC ok [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular SIFT</td>
<td>-</td>
<td>320 (87%)</td>
<td>0.9</td>
</tr>
<tr>
<td>Quantized SIFT 100</td>
<td>100</td>
<td>319 (86%)</td>
<td>3.1</td>
</tr>
<tr>
<td>Quantized SIFT 1k</td>
<td>1k</td>
<td>304 (82%)</td>
<td>17.4</td>
</tr>
<tr>
<td>Quantized SIFT 10k</td>
<td>10k</td>
<td>246 (67%)</td>
<td>10.2</td>
</tr>
<tr>
<td>Quantized Hamming 100</td>
<td>100</td>
<td>307 (83%)</td>
<td>141.6</td>
</tr>
<tr>
<td>Quantized Hamming 64-bit</td>
<td>1k</td>
<td>300 (81%)</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>10k</td>
<td>272 (74%)</td>
<td>0.9</td>
</tr>
</tbody>
</table>
Correspondence Selection

<table>
<thead>
<tr>
<th>Matching Method</th>
<th>Voc.</th>
<th>Size</th>
<th># Images</th>
<th>RANSAC ok [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular SIFT</td>
<td>-</td>
<td>320</td>
<td>(87%)</td>
<td>0.9</td>
</tr>
<tr>
<td>Quantized SIFT</td>
<td>100</td>
<td>319</td>
<td>(86%)</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>1k</td>
<td>304</td>
<td>(82%)</td>
<td>17.4</td>
</tr>
<tr>
<td></td>
<td>10k</td>
<td>246</td>
<td>(67%)</td>
<td>10.2</td>
</tr>
<tr>
<td>Quantized Hamming (64-bit)</td>
<td>100</td>
<td>307</td>
<td>(83%)</td>
<td>141.6</td>
</tr>
<tr>
<td></td>
<td>1k</td>
<td>300</td>
<td>(81%)</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td>10k</td>
<td>272</td>
<td>(74%)</td>
<td>0.9</td>
</tr>
</tbody>
</table>
Conclusion

- Incorrect votes are a major source of error for image retrieval-based localization.
- **Hamming voting** avoids most incorrect votes at little computation and memory overhead.
- Image retrieval with Hamming voting yields **scalable image-based localization**.
- **Correspondence selection** can be accelerated using quantized matching.