Query Expansion for Visual Search using Data Mining Approach

Ph.D. Defense Presentation

Siriwat Kasamwattanarote
シリワット カセッムワッタナロット
21 January 2016

Department of Informatics (National Institute of Informatics), SOKENDAI (The Graduate University for Advanced Studies), Tokyo, Japan.
Note on major requirements from the previous presentation

**Presentation**

1. Discussing about weakness and limitation of the research. (done)
2. In which cases the method fails (done)
   - Evidences showing good/bad results.
3. Conducting experiments on larger datasets. (done)
   - MVS dataset/Instance search dataset

**Thesis**

1. Intensive literature review. (done)
2. Finishing thesis. (almost done)
Overview

1. Introduction
   • Motivation
   • Baseline problem

2. Contributions list
   • Visual word mining
   • Spatial verification
   • Automatic parameter tuning

3. Proposed methods

4. Experimental results
   • Overall
   • Robustness
   • Time consumption

5. Conclusion
   • Research achievements
   • Pros and Cons
   • Limitation

6. Future work
   • Speed up
   • Binary feature
1. Introduction

Cameras -> Producing -> Internet -> Indexing -> Big images collection -> Retrieving -> Mobile devices
1.1 Motivation

- Big images collection.
- Querying on-the-fly with mobile devices.
- Accuracy issue.

- **State-of-the-art approaches**
  - Bag-of-visual-word (BoVW)
  - Average query expansion (AQE)
1.1.1 Bag-of-Visual-Word (BoVW)\textsuperscript{[1]} (1)

- Image representation using BoVW technique.

\begin{itemize}
  \item Feature extraction, SIFT \textsuperscript{[2,3]}
  \item Clustering, AKM \textsuperscript{[4]}
  \item Quantization, ANN \textsuperscript{[5]}
\end{itemize}

\textbf{BoVW histogram}

- 1M clusters

\begin{itemize}
  \item Image Query
  \item Frequency
  \item Visual words (1M)
\end{itemize}

\textbf{Ref.}

\begin{itemize}
\end{itemize}
1.1.1 Bag-of-Visual-Word (BoVW)\[1\] (2)

- Object-based image retrieval by $\textit{BoVW}$

Ref:

1.1.1.1 Similarity Calculation

\[ \text{sim}(Q, I) = 1 - \left\| \frac{Q}{\|Q\|_1} - \frac{I}{\|I\|_1} \right\|_1 \]

\[ R = \{I_b \in D | I_b \text{ contains object appeared on } Q \} \]

\( Q \) = Query image
\( D \) = Database images
\( R \) = Retrieved images
\( I \) = Reference image
1.1.1.2 BoVW problem

Search

Q

R

Partially matched of an object / visual words on the irrelevant image.

(kin mugi)

(ka wa ru)
1.1.2 Average Query Expansion (AQE) [1]

Ref:
All images will be averaged

$k = \text{Total images}$
AQE

Only verified images and inliers will be averaged

\( Q' \) = verified images

\( R \)

\( Q'' \)
RANSAC spatial verification between images

1.1.2.1 AQE problem (inlier threshold = 4)

**Normal query**
- 1-to-M
- inlier = 10
- inlier = 7
- inlier = 8
- inlier = 7
- inlier = 6
- inlier = 14

**Bad condition query**
- 1-to-M
- inlier = 4
- inlier = 3
- inlier = 2
- inlier = 2
- inlier = 2
- inlier = 10

Too many relevant images were rejected

Self-correspondences without query over-dependency?

Query Bootstrapping!!!
1.1.2.2 Query conditions

On-the-fly image retrieval.

Good query may not be as expected.
1.2 Research objective

- This research aims to relax the over-dependency on query verification.
  - By finding the consistency among highly ranked images, instead.
- We evaluate our methods on several standard datasets.
  - Oxford building 5k, 105k.
  - Paris landmark 6k.
  - Extended distractor with MIR Flickr 1M for (Oxford 1m and Paris 1m)
- Robustness on several query degradation cases.
Where we are?

Recent Oxford 5k, 105k, and Paris 6k performance

<table>
<thead>
<tr>
<th>Method</th>
<th>Oxford 5k</th>
<th>Oxford 105k</th>
<th>Paris 6k</th>
<th>Oxford 1m</th>
<th>Paris 1m</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoVW [1]</td>
<td>61.20</td>
<td>64.50</td>
<td>78.50</td>
<td>82.70</td>
<td>81.40</td>
</tr>
<tr>
<td>Spatial verification [1]</td>
<td>64.50</td>
<td>65.50</td>
<td>72.00</td>
<td>80.50</td>
<td>78.30</td>
</tr>
<tr>
<td>AQE [2]</td>
<td>78.80</td>
<td>81.40</td>
<td>79.80</td>
<td>82.30</td>
<td>78.20</td>
</tr>
<tr>
<td>Local geometry [3]</td>
<td>82.70</td>
<td>87.60</td>
<td>80.00</td>
<td>89.60</td>
<td>85.60</td>
</tr>
<tr>
<td>Total recall II [4]</td>
<td>81.40</td>
<td>81.80</td>
<td>81.80</td>
<td>89.00</td>
<td>85.60</td>
</tr>
<tr>
<td>Hello neighbors [5]</td>
<td>79.80</td>
<td>80.00</td>
<td>80.00</td>
<td>82.84</td>
<td>76.33</td>
</tr>
<tr>
<td>DQE [6]</td>
<td>82.30</td>
<td>82.84</td>
<td>81.80</td>
<td>88.12</td>
<td>80.44</td>
</tr>
<tr>
<td>AQE [7]</td>
<td>89.60</td>
<td>88.12</td>
<td>89.00</td>
<td>86.41</td>
<td>88.96</td>
</tr>
<tr>
<td>DQE + Boosting (group) [7]</td>
<td>82.84</td>
<td>86.41</td>
<td>87.60</td>
<td>90.36</td>
<td>89.52</td>
</tr>
</tbody>
</table>


Ref:
Result overview

• Overall accuracy improvement
  Normal query + 10-14% (best)

• Higher robustness to low quality queries
  Low resolution / Small object / Blur + ~26% (best)
  Noisy + ~19-26% (best)
Query Expansion for Visual Search using Data Mining Approach

Overview

1. Introduction
   • Motivation
   • Baseline problem

2. Contributions list
   • Visual word mining
   • Spatial verification
   • Automatic parameter tuning

3. Proposed methods

4. Experimental results
   • Overall
   • Robustness
   • Time consumption

5. Conclusion
   • Research achievements
   • Pros and Cons
   • Limitation

6. Future work
   • Speed up
   • Binary feature
2. Contributions list

1. We proposed a “Query Bootstrapping (QB)” as a visual mining for query expansion
   • To discover object consistency among highly ranked images by using Frequent Itemset Mining (FIM)
   • Relaxed a strong constraint between a query image and first-round retrieved list.
   • Gained higher robustness on low quality query.

2. We proposed an “Adaptive Support (ASUP)” tuning algorithm for FIM.
   • To automatically provide an optimal support value (important parameter for FIM).
   • Locally optimize support value for each query, for the best performance of each query.

3. We integrated a LO-RANSAC spatial verification (SP) based method to QB (QB + SP).
   • To verify correspondences between a query and retrieved images.
   • Give a chance for FIM to find correct co-occurrence patterns through the whole of verified images.
     • Less constraint than AQE

4. We proposed an “Adaptive Inlier Threshold (ADINT)” for LO-RANSAC
   • To find an inlier threshold automatically.
   • Good for QB + SP.

Average improvement over the state-of-the-arts

<table>
<thead>
<tr>
<th></th>
<th>BoVW</th>
<th>AQE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q4.2013</td>
<td>+3%</td>
<td>-1%</td>
</tr>
<tr>
<td>Q1.2014</td>
<td>+5%</td>
<td>+1%</td>
</tr>
<tr>
<td>Q4.2014</td>
<td>+12%</td>
<td>+7%</td>
</tr>
<tr>
<td>Q1.2015</td>
<td>+14%</td>
<td>+9%</td>
</tr>
</tbody>
</table>
Query Expansion for Visual Search using Data Mining Approach

Overview

1. Introduction
• Motivation
• Baseline problem

2. Contributions list
• Visual word mining
• Spatial verification
• Automatic parameter tuning

3. Proposed methods

4. Experimental results
• Overall
• Robustness
• Time consumption

5. Conclusion
• Research achievements
• Pros and Cons
• Limitation

6. Future work
• Speed up
• Binary feature
3. Proposed methods

1. Visual word mining
2. Spatial verification

Query Bootstrapping (QB)

QB / QB + SP architecture diagram
Intro - Frequent Itemset mining (FIM)

Frequent Itemset mining (FIM)

<table>
<thead>
<tr>
<th>Img. $I_k$</th>
<th>Trans. $t_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_1$</td>
<td>$t_1 = {i_1, i_2, i_4, i_6}$</td>
</tr>
<tr>
<td>$I_2$</td>
<td>$t_2 = {i_2, i_5, i_8}$</td>
</tr>
<tr>
<td>$I_3$</td>
<td>$t_3 = {i_2, i_3, i_9}$</td>
</tr>
<tr>
<td>$I_4$</td>
<td>$t_4 = {i_1, i_2, i_4, i_7}$</td>
</tr>
<tr>
<td>$I_5$</td>
<td>$t_5 = {i_2, i_3, i_8}$</td>
</tr>
</tbody>
</table>

Pattern | support |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>${i_2}$</td>
<td>60%</td>
</tr>
<tr>
<td>${i_3}$</td>
<td>40%</td>
</tr>
<tr>
<td>${i_8}$</td>
<td>40%</td>
</tr>
<tr>
<td>${i_1, i_4}$</td>
<td>40%</td>
</tr>
<tr>
<td>${i_3, i_8}$</td>
<td>20%</td>
</tr>
<tr>
<td>${i_1, i_4, i_7}$</td>
<td>20%</td>
</tr>
<tr>
<td>${i_2, i_3, i_9}$</td>
<td>20%</td>
</tr>
<tr>
<td>${i_2, i_5, i_8}$</td>
<td>20%</td>
</tr>
<tr>
<td>${i_1, i_2, i_4, i_6}$</td>
<td>20%</td>
</tr>
</tbody>
</table>
Related works that applied FIM

• Video mining [1]
  • Mining visual word motions into groups.

• Visual phrase mining [2]
  • Finding visual phrase lexicon.
  • Separating object out of background.

• Mining multiple queries [3]
  • Mining query patterns to better focus of targeted object.

• Mining for re-ranking and classification [4]
  • Voting image score by counting FIM patterns.

Our work closed to
  • But we are on the result side.
  • But we feed back result as AQE.

Non of them work directly on
FIM for Query expansion!

Ref:
3.1 Contribution 1 - QB

- Mining co-occurrence visual words among highly ranked images.
  - FIM returns frequent patterns \((fi)\).
- Constructing a new query \((Q''')\)
  - We regard \(fi\) is a representative form of the occurrences of visual words.
  - Considering a new term \(fi\) into a standard BoVW term \((tf-idf)\)
  - Named as \(tf-fi-idf\) (or \(fi \times tf-idf\))
3.1 QB problem 1 (1)

- FIM is designed for
  - Many transactions, Less items (n).
  - Total possible patterns \( \approx 2^n \)
- BoVW size up to 1 million, **slow down** FIM.
  - Less images, many words (n).

\[ n = \text{total non-zero visual words} \]
3.1 QB problem 1 (2)

• Helped by
  • Transaction transposition [1-3].

\[ n = \text{total top-k images} \]

\[ << n \]

Transactions

\[ \text{Transaction DB}^T \]

FIM

Patterns

\[ 2^{<<n} \]

Transitions

Ref:
3.1 QB problem 2

What if we set support individually? Is it better to set it locally?

• How much support value is appropriate?
  • Too low support give too much patterns.
  • Too high support might give nothing.
3.2 Contribution 2 - ASUP

- **Adaptive Support** tuning algorithm for *individual query*.

As we observed.. The optimal support is at the highest frequent patterns.

*Pattern amount at each specific support range*
3.2 Contribution 2 – ASUP (2)

- ASUP algorithm

Ref:

Optimal!!

minsupt = 30

maxsup = 50
3.2 ASUP problem (1)

- BoVW result ($R$) may be dominated by irrelevant images.

Top 10 images example.
The rest of images are mostly a branches and a tree →

Round 1 $R$ (BoVW)

Round 2 $R$ (QB)
3.2 ASUP problem (2)

- The performance is decreasing when the number of top-k is increasing.

<table>
<thead>
<tr>
<th>top-k</th>
<th>AQE</th>
<th>QB</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>87.73</td>
<td>86.41</td>
</tr>
<tr>
<td>50</td>
<td>88.01</td>
<td>83.20</td>
</tr>
<tr>
<td>75</td>
<td>87.99</td>
<td>79.12</td>
</tr>
<tr>
<td>100</td>
<td>88.11</td>
<td>74.23</td>
</tr>
</tbody>
</table>
3.3 Contribution 3 - QB + SP (1)

- Spatial verification is back
  - Properly for QB.
  - To give hints of verify *images*.
  - Mining will be more focused.
3.3 Contribution 3 - QB + SP (2)

- Too low filtering nothing.
- Too high filtering everything.

Problem

Accepting relevant images is fine!

Accepting irrelevant images leads high noise to FIM!
3.4 Contribution 4 – ADINT (1)

• Adaptive Inlier Threshold (ADINT) algorithm
  1. Feed top-k to LO-RANSAC
  2. Constructing the inlier count histogram.
  3. Select a pivot on a peak.
  4. Sweeping clockwise from a pivot with a radius of 0.9 (ADINT ratio)
  5. The first point that cut histogram will be an Adaptive Inlier Threshold.
3.4 Contribution 4 – ADINT (2)

• Why ADINT ratio = 0.9?

ADINT ratio \( \sim 0.9 \)
Always gives the best
ADINT performance
3.4 Contribution 4 – ADINT (3)

- **ADINT** thresholding result

**Color code**
- (blue) Inlier count from LO-RANSAC
- (red) ADINT threshold
- (orange) Automated selected relevant images
- (gray) Ground truth
Query Expansion for Visual Search using Data Mining Approach

Overview

1. Introduction
   • Motivation
   • Baseline problem

2. Contributions list
   • Visual word mining
   • Spatial verification
   • Automatic parameter tuning

3. Proposed methods

4. Experimental results
   • Overall
   • Robustness
   • Time consumption

5. Conclusion
   • Research achievements
   • Pros and Cons
   • Limitation

6. Future work
   • Speed up
   • Binary feature
4. Experimental results (1)

• **Standard dataset**
  - Oxford building 5k and 105k.
  - Paris 6k.
  - Total 55 queries on each dataset.
    - 11 landmarks and locations (topic).
    - 5 different views on each topic.

• **Extra 1 million distractor dataset images**
  - MIR Flickr 1m to make Oxford building 1m and Paris 1m.

• **Evaluation protocol**
  - We use mean average precision (mAP) as an evaluation metric.
  - And ground truth files obtained from the dataset provider.

Ref:
Oxford dataset: [http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/](http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/)
Paris dataset: [http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/](http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/)
4. Experimental results (2)

- Dataset examples

Paris landmarks

Oxford buildings
4. Experimental results (3)

1. Overall retrieval performance
2. Contributions comparison
3. Impact of Top-$k$ retrieval images
4. Automatic parameter evaluation
5. Impact of varies quality query
6. Time consumption
4.1 Overall retrieval performance

![Bar chart showing mAP for each method and dataset for different datasets: BoVW, AQE [39], AQE, QB, QB + SP.]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BoVW</th>
<th>AQE</th>
<th>AQE [39]</th>
<th>QB</th>
<th>QB + SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ox 5k</td>
<td>82.84</td>
<td>78.50</td>
<td>78.12</td>
<td>86.41</td>
<td>93.49</td>
</tr>
<tr>
<td>Ox 105k</td>
<td>75.66</td>
<td>72.50</td>
<td>80.71</td>
<td>75.67</td>
<td>90.36</td>
</tr>
<tr>
<td>Ox 1m</td>
<td>75.28</td>
<td>72.00</td>
<td>78.48</td>
<td>77.56</td>
<td>89.52</td>
</tr>
<tr>
<td>Paris 6k</td>
<td>76.33</td>
<td>80.44</td>
<td>80.44</td>
<td>88.28</td>
<td>88.96</td>
</tr>
<tr>
<td>Paris 1m</td>
<td>59.95</td>
<td>64.32</td>
<td>69.94</td>
<td>69.94</td>
<td>79.81</td>
</tr>
</tbody>
</table>

Ref:
4.2 Contributions comparison

• Notation of our proposed methods
  • QB = (QB + ASUP)
  • QB + SP = (QB + ASUP) + (SP + ADINT)

<table>
<thead>
<tr>
<th></th>
<th>Ox 5k</th>
<th>Ox 105k</th>
<th>Paris 6k</th>
</tr>
</thead>
<tbody>
<tr>
<td>QB + FSUP</td>
<td>83.52</td>
<td>74.43</td>
<td>84.77</td>
</tr>
<tr>
<td>QB + ASUP</td>
<td>86.41</td>
<td>75.67</td>
<td>88.28</td>
</tr>
<tr>
<td>QB + ASUP + SP + FINT</td>
<td>92.48</td>
<td>89.31</td>
<td>87.76</td>
</tr>
<tr>
<td>QB + ASUP + SP + ADINT</td>
<td>93.49</td>
<td>90.36</td>
<td>88.96</td>
</tr>
</tbody>
</table>

The performance comparison between our contributions
4.3 Impact of Top-k relevant images

**Result:**

- Higher top-k is **good** for spatial verification based methods.
  - Some relevant images can be found in lower ranked images.
  - AQE, QB + SP
- Higher top-k is **bad** for greedy methods.
  - Too many irrelevant images were added during aggregation.
  - QE, QB

```
Why QE/QB did not fail on Paris6k?
Because of the number of true positive images.
Paris6k has avg.~163 (51-289) positive images.
Oxford has avg.~51 (6-221) positive images.
```
4.4.1 Adaptive support (ASUP)

- Experiment for FIM based methods (run with QB + SP)
- Comparison of
  - mAP of a **fixed minimum support** of 5 to 95
  - and **adaptive support** (ASUP)

---

Best performance — Achieved by **ASUP**, which also has much lower variances.
4.4.2 Adaptive inlier threshold (ADINT)

- Experiment for AQE, QB + SP
- Comparison on mAP of
  - Fixed inlier threshold (FINT) of 3, 5, 7, 9, 11 and
  - Adaptive inlier threshold (ADINT) or A

\[ \Delta(\text{min, A}) \] is how much ADINT better than a minimum of FINT.

\[ \Delta(\text{max, A}) \] is how much ADINT better than a maximum of FINT.

**Result:**
- ADINT better than FINT in most cases of QB + SP.
- ADINT does not improve much on AQE, but at least it’s automated!!

<table>
<thead>
<tr>
<th>Inlier Threshold</th>
<th>AQE (mAP %)</th>
<th>QB + SP (mAP %)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ox5k</td>
<td>Ox105k</td>
</tr>
<tr>
<td>3</td>
<td>88.11</td>
<td>80.44</td>
</tr>
<tr>
<td>5</td>
<td>88.60</td>
<td>80.13</td>
</tr>
<tr>
<td>7</td>
<td>87.87</td>
<td>79.19</td>
</tr>
<tr>
<td>9</td>
<td>87.32</td>
<td>78.87</td>
</tr>
<tr>
<td>11</td>
<td>87.13</td>
<td>78.70</td>
</tr>
<tr>
<td>A</td>
<td>87.88</td>
<td>81.85</td>
</tr>
<tr>
<td>( \Delta(\text{min, A}) )</td>
<td>0.75</td>
<td>2.16</td>
</tr>
<tr>
<td>( \Delta(\text{max, A}) )</td>
<td>-0.72</td>
<td>-0.01</td>
</tr>
</tbody>
</table>

**ADINT vs. FINT performance**
4.5 Impact of a noisy query

Sample query image with noise @sigma = 2.0

<table>
<thead>
<tr>
<th>Gaussian sigma (σ)</th>
<th>mAP</th>
<th>w/o</th>
<th>1.0</th>
<th>1.5</th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oxford 5k mAP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>82.84</td>
<td>80.17</td>
<td>73.32</td>
<td>62.28</td>
<td></td>
</tr>
<tr>
<td>AQE</td>
<td>88.12</td>
<td>88.24</td>
<td>86.43</td>
<td>82.02</td>
<td></td>
</tr>
<tr>
<td>QB</td>
<td>86.41</td>
<td>79.94</td>
<td>66.29</td>
<td>51.18</td>
<td></td>
</tr>
<tr>
<td>QB + SP</td>
<td>93.49</td>
<td>92.15</td>
<td>90.71</td>
<td>89.03</td>
<td></td>
</tr>
</tbody>
</table>

| Oxford 105k mAP    |     |     |     |     |     |
| Baseline           | 75.66 | 71.25 | 62.45 | 49.36 |
| AQE                | 80.71 | 80.92 | 76.25 | 67.92 |
| QB                 | 75.67 | 63.49 | 46.02 | 35.18 |
| QB + SP            | 90.36 | 88.48 | 84.60 | 75.92 |

| Paris 6k mAP       |     |     |     |     |     |
| Baseline           | 76.33 | 72.82 | 66.21 | 57.72 |
| AQE                | 80.44 | 77.14 | 75.77 | 74.05 |
| QB                 | 88.28 | 85.01 | 83.77 | 77.70 |
| QB + SP            | 88.96 | 87.11 | 86.61 | 84.64 |

mAP vs. noise level
4.5 Impact of a low resolution query

Sample query image with scale of 20% of original

<table>
<thead>
<tr>
<th>Query scale (%)</th>
<th>Oxford 5k mAP</th>
<th>Oxford 105k mAP</th>
<th>Paris 6k mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>80</td>
<td>82.84</td>
<td>82.29</td>
<td>79.89</td>
</tr>
<tr>
<td>60</td>
<td>82.25</td>
<td>82.25</td>
<td>79.89</td>
</tr>
<tr>
<td>40</td>
<td>79.89</td>
<td>79.89</td>
<td>79.89</td>
</tr>
<tr>
<td>20</td>
<td>66.47</td>
<td>66.47</td>
<td>66.47</td>
</tr>
</tbody>
</table>

- Baseline
- AQE
- QB
- QB + SP

mAP vs. image scale
4.6 Time consumption

• **Overall time consumption**
  • *Fast* with BoVW, and AQE
  • *Slow* with QB, and QB + SP
4.6 Time consumption - breakdown

- FIM-based methods are **QB** and **QB + SP**
- **Result:**
  - FIM is the most *slowest part*, why?
### 4.6.1 Colossal pattern [1]

#### Lower number of pattern
BoVW not really good
our QB + SP gives it big improvement
Query class: Easy (to be improved)

#### Higher number of pattern
BoVW already good
our QB + SP gives a small improvement
Query class: Hard (to be improved)

<table>
<thead>
<tr>
<th>Type</th>
<th>#Topics</th>
<th>BoVW</th>
<th>QB</th>
<th>QB+SP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>mAP(%)</td>
<td>mAP(%)</td>
<td>mAP(%)</td>
</tr>
<tr>
<td>Ox 5k</td>
<td>Easy</td>
<td>40</td>
<td>81.26</td>
<td>0.075</td>
</tr>
<tr>
<td></td>
<td>Hard</td>
<td>15</td>
<td>87.06</td>
<td>4.471</td>
</tr>
<tr>
<td>Ox 105k</td>
<td>Easy</td>
<td>40</td>
<td>73.94</td>
<td>0.011</td>
</tr>
<tr>
<td></td>
<td>Hard</td>
<td>15</td>
<td>80.24</td>
<td>0.109</td>
</tr>
<tr>
<td>Paris 6k</td>
<td>Easy</td>
<td>25</td>
<td>71.09</td>
<td>0.922</td>
</tr>
<tr>
<td></td>
<td>Hard</td>
<td>30</td>
<td>80.69</td>
<td>21.475</td>
</tr>
</tbody>
</table>

4.7 Result

BoVW
Baseline

AQE
More relevant to query ROI

QB + SP
Relevant to each others
4.7 Result

- **BoVW**: Baseline
- **AQE**: More relevant to query ROI
- **QB + SP**: Relevant to each other
Overview

1. Introduction
   • Motivation
   • Baseline problem

2. Contributions list
   • Visual word mining
   • Spatial verification
   • Automatic parameter tuning

3. Proposed methods

4. Experimental results
   • Overall
   • Robustness
   • Time consumption

5. Conclusion
   • Research achievements
   • Pros and Cons
   • Limitation

6. Future work
   • Speed up
   • Binary feature
5. Conclusion

• **We proposed**
  • “Query Bootstrapping (QB)” as visual mining technique for query expansion.
  • The way to integrate “Spatial Verification (SP)” for such mining.

• **The important parameters are automatically determined.**
  • Adaptive support (ASUP) for FIM.
  • Adaptive inlier threshold (ADINT) for LO-RANSAC.

• **Achievements**
  • Our methods reach the highest performance on all datasets.
  • Very high robustness on difficult cases of query quality are proved.
5.1 Benefits of using QB

• *To help understand more on the target object and its context.*
  • Context can also be learned.
  • Hidden visual words from other view angles can be learned.

• *QB can be used to reject irrelevant visual words.*
  • Object occlusions.
  • Misleading visual words.
  • Not useful visual words, not clearly related to the object.
5.1.1 Context discovery example (1)

- Query topic: defense_2
5.1.1 Context discovery example (2)

- Co-occurrences between top-1 and top-2
5.1.1 Context discovery example (3)

• Learned object contexts that help describing a target object.
5.1.1 Context discovery example (4)

- **AQE** result of “defense_2” on Paris 1M, AP = **27.04%**
5.1.1 Context discovery example (5)

- **QB** result of “defense_2” on Paris 1M, AP = 71.35%
5.1.1 Context discovery example (6)

- **AQE** result of “moulinrouge_1” on Paris 1M, AP = 28.86%
5.1.1 Context discovery example (7)

- **QB** result of “moulinrouge_1” on Paris 1M, AP = **83.52%**
5.1.2 Hidden visual words discovery (1)

• One query image may have limited visual contents
5.1.2 Hidden visual words discovery (2)

- **QB** finds hidden visual words within the target object
  - Using relevance images.
5.1.2 Hidden visual words discovery (3)

• **AQE** Result (AP 23.67%)

• **QB** Result (AP 44.77%)
5.1.3 Irrelevant visual word identification (1)

• Misleading visual words in AQE matching.
5.1.3 Irrelevant visual word identification (2)

- QB can identify and reject those visual words.
5.1.3 Irrelevant visual word identification (3)

- Misleading visual words in AQE matching.
5.1.3 Irrelevant visual word identification (4)

- **QB** can identify and reject those visual words.
5.2 QB limitations

• Experiments with the other datasets
  • Mobile visual search
  • Instance Search

• Target dataset characteristics

• Weakness summarization
5.2.1 Experiments with the other datasets (1)

- **Stanford Mobile Visual Search**
  - Book covers
  - Business cards
  - CD covers
  - DVD covers
  - Landmarks
  - Museum paintings
  - Prints
  - Video frames

Only one reference image is available. **No more consistency among the retrieved images.**
5.2.1 Experiments with the other datasets (2)

• Instance Search 2011, 2013

• MAX Late fusion
5.2.1 Experiments with the other datasets (3)

- Instance Search performance evaluation

<table>
<thead>
<tr>
<th>Methods</th>
<th>Instance Search 2011</th>
<th>Instance Search 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoVW</td>
<td>48.61</td>
<td>21.82</td>
</tr>
<tr>
<td>AQE</td>
<td>41.87</td>
<td>18.41</td>
</tr>
<tr>
<td>QB</td>
<td>46.54</td>
<td></td>
</tr>
<tr>
<td>QBSP</td>
<td>39.28</td>
<td></td>
</tr>
</tbody>
</table>
5.2.1 Experiments with the other datasets (4)

• QB works well with some query e.g. “9028”

• BoVW – Result consisted with several big enough airplanes. (AP = 52.14%)

• QBSP – Mining pattern focused on an airplane (AP = 80.98%)
5.2.1 Experiments with the other datasets (5)

• QB works well with some query e.g. “9029”

• BoVW – This room (AP = 51.26%)

• QBSP – This room (AP = 64.12%)
5.2.1 Experiments with the other datasets (6)

- QB *works* well with some query e.g. “9037”

- BoVW – A back balloon (AP = 40.07%)

- QBSP – A back balloon helped by in front balloon (AP = 47.61%)
5.2.1 Experiments with the other datasets (7)

- QB **do not works** in the most cases e.g.

  - BoVW – A back balloon (AP = **18.72%**)

  ![BoVW result image]

  - QBSP – A back balloon helped by in front balloon (AP = **3.85%**)

  ![QBSP result image]
5.2.2 Target dataset characteristics

• QB will work perfectly when
  • Original BoVW provides **good enough result**, then QB will boost its performance.
  • QB help improving the performance by **using context**, e.g. Finding an **object that does not move**, or **finding a landmark**.
5.2.3 Weakness

- QB will not work if
  - Only one true positive is provided, so no more consistency can be discovered, e.g. MVS dataset.
  - To search for a deformable object, e.g. Cloth, animal, texture less object, etc. (mostly are the characteristic of INS dataset)

- Results of QB are narrow
  - QB try to find thing that similar to each others out of the relevancies.
6. Future work

• This research can be extended
  • Detect the possibility of colossal pattern.
  • Let AQE handle the task of "Hard" query.
  • Result to reduce overall time consumption taken by our QB.
6. Future work

- We also did experiments on binary feature.
- ORB feature
6. Future work

- ORB experiments on MVS dataset

<table>
<thead>
<tr>
<th>Query topics</th>
<th>SIFT</th>
<th>ORB</th>
</tr>
</thead>
<tbody>
<tr>
<td>book covers</td>
<td>61.21</td>
<td>97.79</td>
</tr>
<tr>
<td>business cards</td>
<td>86.33</td>
<td>88.74</td>
</tr>
<tr>
<td>cd covers</td>
<td>61.10</td>
<td>95.61</td>
</tr>
<tr>
<td>dvd covers</td>
<td>65.51</td>
<td>99.08</td>
</tr>
<tr>
<td>landmarks</td>
<td>77.52</td>
<td>44.15</td>
</tr>
<tr>
<td>museum painting</td>
<td>94.50</td>
<td>86.17</td>
</tr>
<tr>
<td>print</td>
<td>82.99</td>
<td>79.29</td>
</tr>
<tr>
<td>video frames</td>
<td>97.08</td>
<td>99.35</td>
</tr>
<tr>
<td>average</td>
<td>78.28</td>
<td>86.27</td>
</tr>
</tbody>
</table>

**MVS dataset**

SIFT wins! ORB wins! Par
Query Expansion for Visual Search using Data Mining Approach

Overview and Q/A

1. Introduction
   • Motivation
   • Baseline problem

2. Contributions list
   • Visual word mining
   • Spatial verification
   • Automatic parameter tuning

3. Proposed methods

4. Experimental results
   • Overall
   • Robustness
   • Time consumption

5. Conclusion
   • Research achievements
   • Pros and Cons

6. Future work
   • Speed up
   • Binary feature