Query Expansion for Visual Search
using Data Mining Approach

Siriwat Kasamwattanarote
2 December 2015

Department of Informatics (National Institute of Informatics),
SOKENDAI (The Graduate University for Advanced Studies), Tokyo, Japan.

National University
SOKENDAI
The Graduate University for Advanced Studies

大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

Overview

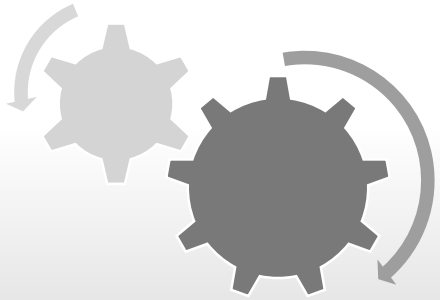
1. Introduction

- Motivation
- Baseline problem

2. Contributions list

- Visual word mining
- Spatial verification
- Automatic parameter tuning

3. Proposed methods



4. Experimental results

- Overall
- Robustness
- Time consumption

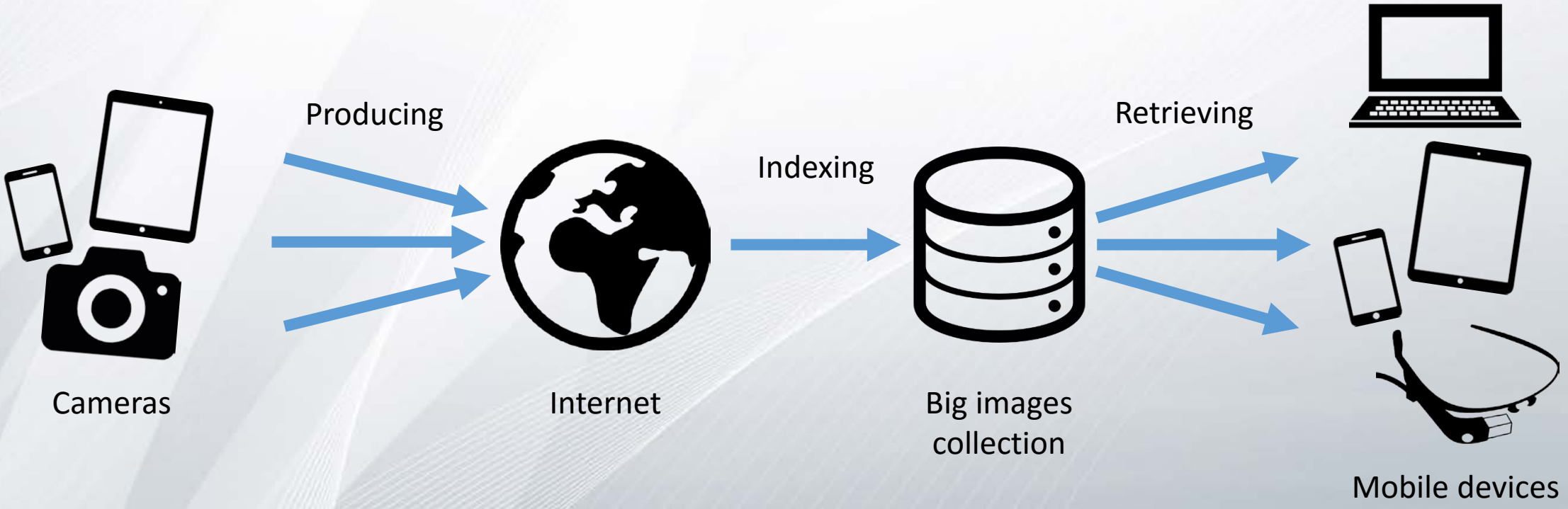
5. Conclusion

- Research achievements
- Pros and Cons

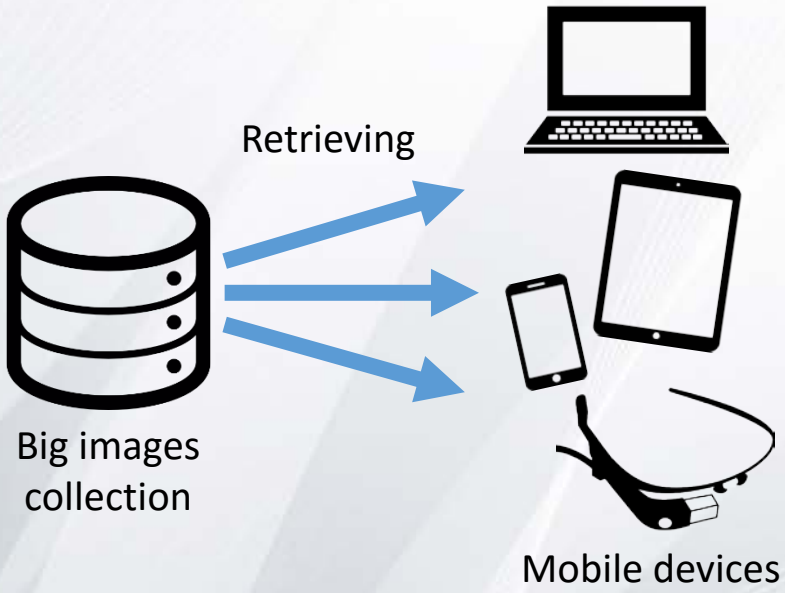
6. Future work

- Speed up
- Binary feature

1. Introduction



1.1 Motivation

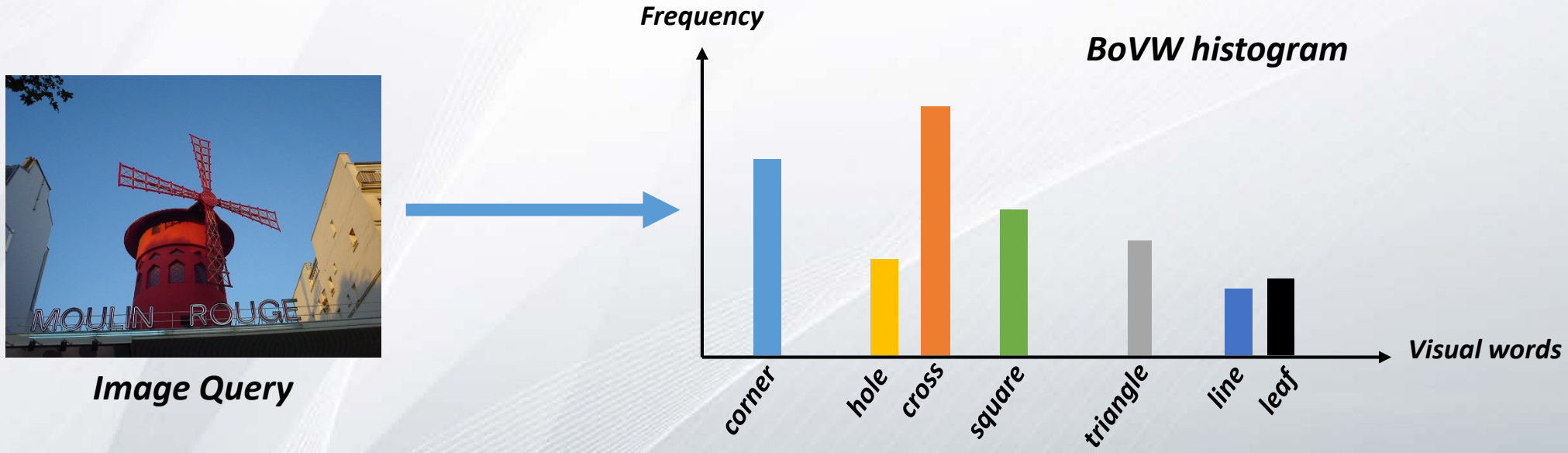


- Big images collection.
- Querying on-the-fly with mobile devices.
- Accuracy issue.

- ***State-of-the-art approaches***
  - Bag-of-visual-word (**BoVW**)
  - Average query expansion (**AQE**)

1.1.1 Bag-of-Visual-Word (BoVW)[1] (1)

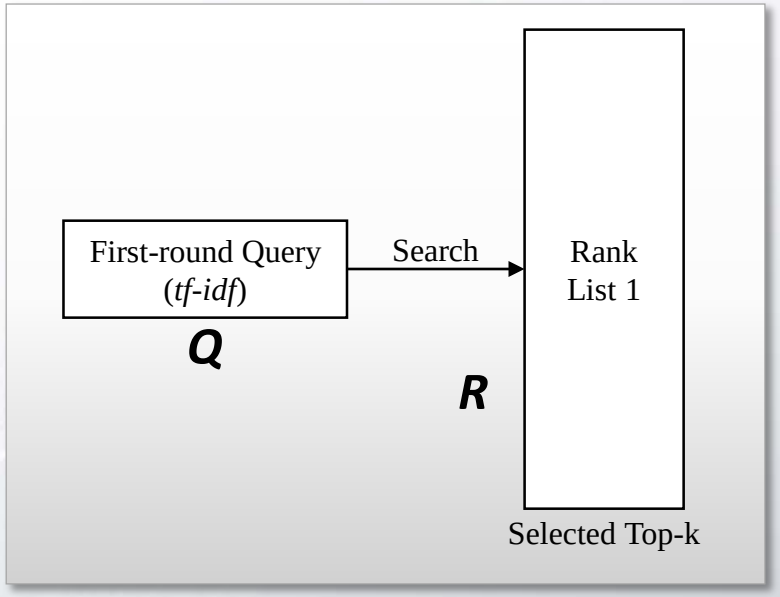
- Image representation of BoVW technique.



Ref:
[1] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," ICCV, pp.1470–1477, 2003.

1.1.1 Bag-of-Visual-Word (BoVW)[1] (2)

- Object-based image retrieval by ***BoVW***



BoVW architecture diagram

Q = Query image
D = Database images
R = Retrieved images

Ref:
[1] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," ICCV, pp.1470–1477, 2003.

1.1.1.1 Similarity Calculation

$$sim(\boldsymbol{Q}, I) = 1 - \left\| \frac{\boldsymbol{Q}}{\|\boldsymbol{Q}\|_1} - \frac{I}{\|I\|_1} \right\|_1$$

$$\boldsymbol{R} = \{I_b \in D | I_b \text{ contains object appeared on } \boldsymbol{Q}\}$$

Q = Query image
D = Database images
R = Retrieved images
I = Reference image

1.1.1.2 BoVW problem

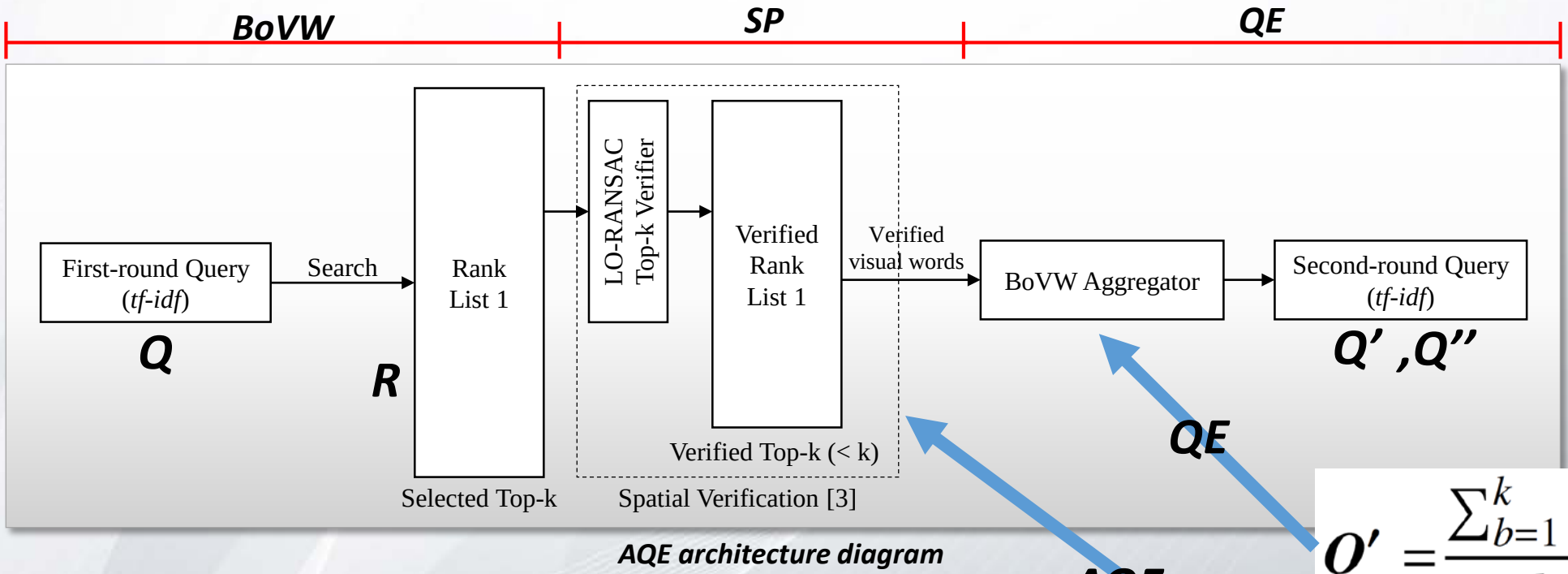
Q

Search

R

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

1.1.2 Average Query Expansion (AQE)[2]



$$Q' = \frac{\sum_{b=1}^{k} R_b}{k}$$

$$Q'' = \frac{Q + \sum_{b=1}^{k'} R_b}{k' + 1}$$

k = Selected top images
k' = Verified images
k' < k

Ref:
[2] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval.," ICCV, pp.1–8, 2007.
[3] K. Lebeda, J. Matas, and O. Chum, "Fixing the locally optimized RANSAC," BMVC, pp.1–11, 2012.

QE

Q

R

All images will be averaged

Q'

k = Total images

AQE

Q

R

inlier = 10 ○
inlier = 7 ○
inlier = 8 ○
inlier = 7 ○
inlier = 6 ○
inlier = 14 ○
inlier = 0 ✕
inlier = 0 ✕
inlier = 0 ✕
inlier = 2 ✕
inlier = 3 ○
inlier = 1 ✕
inlier = 2 ✕

Only ***verified images*** and ***inlied visual words*** will be averaged

Q''

k' = verified images

RANSAC spatial verification between images

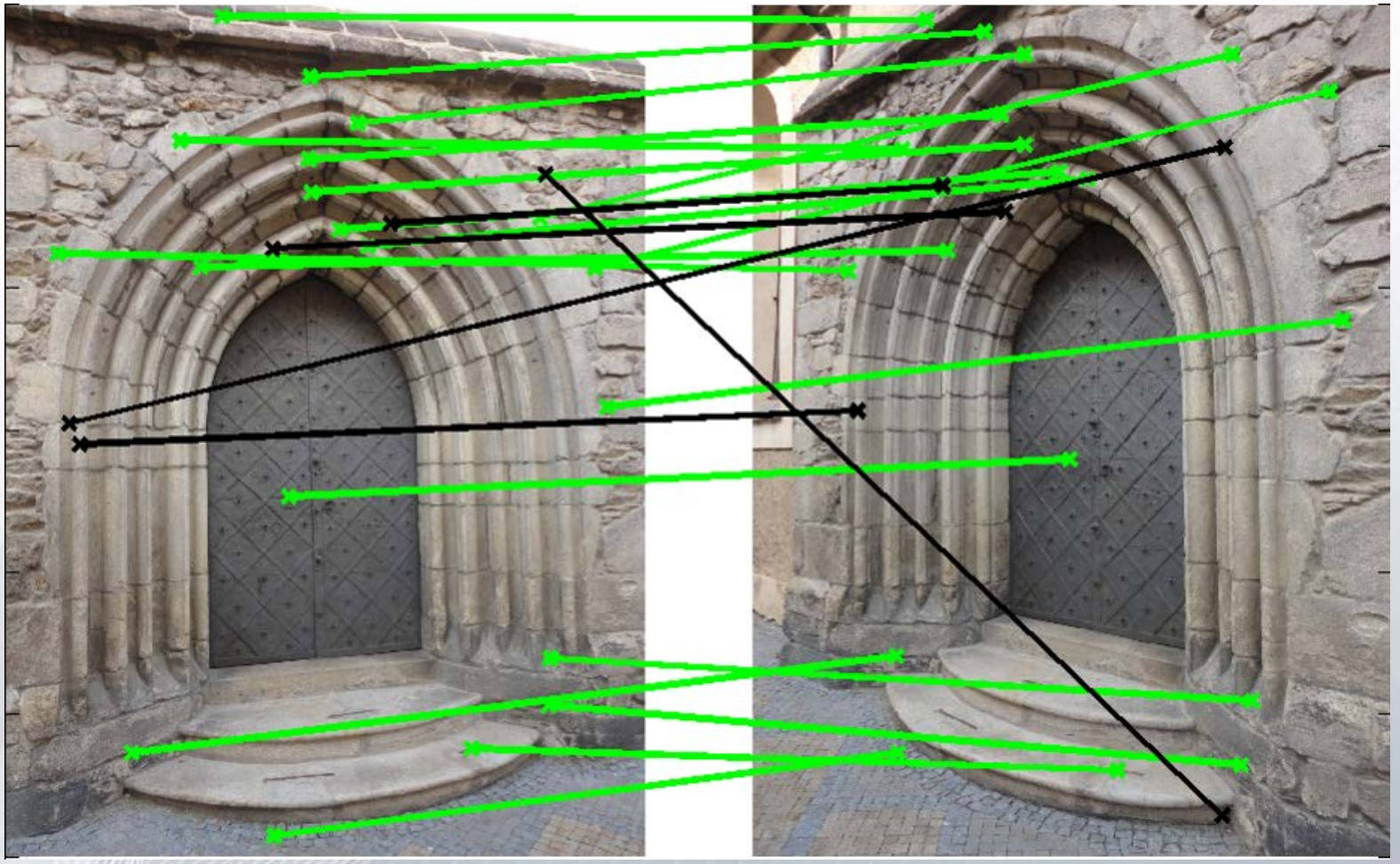


Image reference: https://cyber.felk.cvut.cz/theses/detail.phtml?id=360

1.1.2.1 AQE problem (inlier threshold = 4)

Normal query

inlier = 10 ○
inlier = 7 ○
inlier = 8 ○
inlier = 7 ○
inlier = 6 ○
inlier = 14 ○

...

Low quality query

inlier = 4 ○
inlier = 3 ✗
inlier = 2 ✗
inlier = 2 ✗
inlier = 2 ✗
inlier = 10 ○

...

Too many relevant images were rejected

Self-correspondences without query over-dependency?

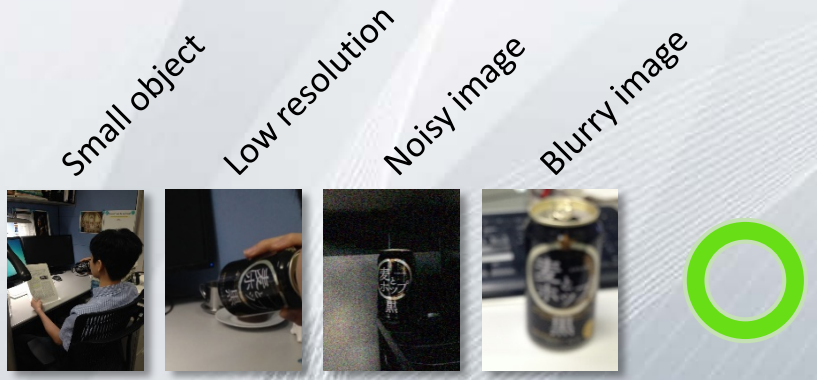
Query Bootstrapping!!!

1.1.2.2 Query quality

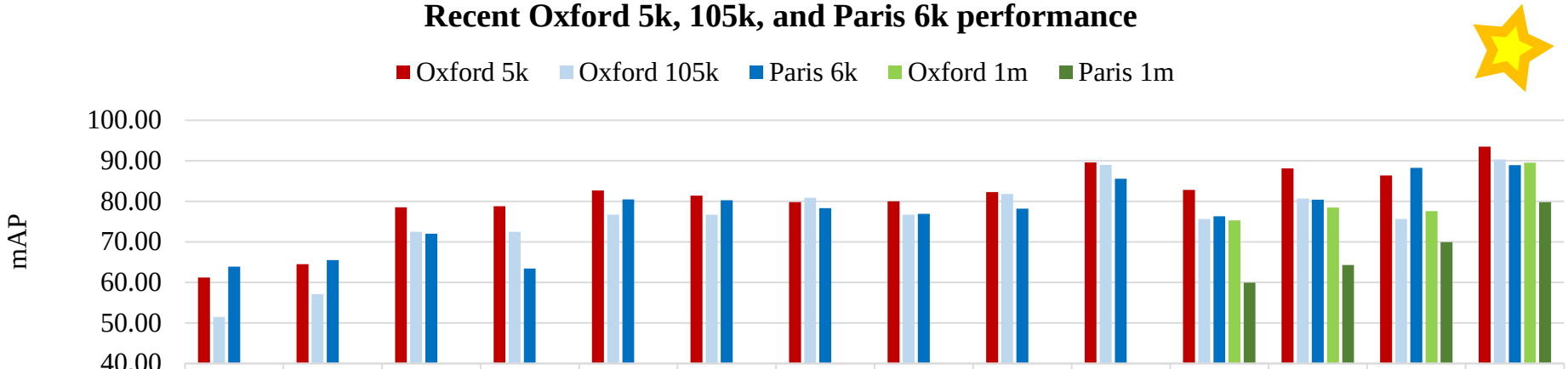
On-the-fly image retrieval..
***Good query** may not be as expected.*

1.2 Research objective

- This research aims to **relax** the **over-dependency** on query checking.
  - By finding the ***consistency among highly ranked images***.
- 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?



	BoVW [1]	Spatial verification [1]	AQE [2]	Local geometry [3]	Total recall II [4]	Hello neighbors [5]	DQE [6]	AQE [7]	DQE + Boosting [7]	DQE + Boosting (group) [7]	BoVW [Our]	AQE [Our]	QB [Our]	QB + SP [Our]
Oxford 5k	61.20	64.50	78.50	78.80	82.70	81.40	79.80	80.00	82.30	89.60	82.84	88.12	86.41	93.49
Oxford 105k	51.50	57.10	72.50	72.50	76.70	76.70	80.90	76.70	81.80	89.00	75.66	80.71	75.67	90.36
Paris 6k	63.90	65.50	72.00	63.40	80.50	80.30	78.30	76.90	78.20	85.60	76.33	80.44	88.28	88.96
Oxford 1m											75.28	78.48	77.56	89.52
Paris 1m											59.95	64.32	69.94	79.81

2007----------2009--2011----------2012---2014----------**2015**---------- つづく

Ref:
[1] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In CVPR, 2007.
[2] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In ICCV, 2007.
[3] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In CVPR, 2009.
[4] O. Chum, A. Mikulik, M. Perdoch, and J. Matas. Total recall II: Query expansion revisited. In CVPR, 2011.
[5] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. J. V. Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In CVPR. IEEE Computer Society, 2011.
[6] R. Arandjelovic. Three things everyone should know to improve object retrieval. In CVPR, 2012.
[7] C. Yanzhi, L. Xi, D. Anthony, and H. Anton van den. Boosting object retrieval with group queries. In SPS, 2014.

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)**

Overview

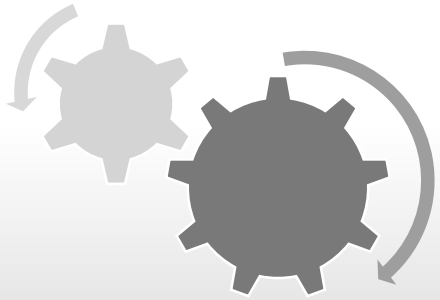
1. Introduction

- 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

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 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

	BoVW	*AQE*
Q4-2013	**+3%**	**-1%**
Q1-2014	**+5%**	**+1%**
Q4-2014	**+12%**	**+7%**
Q1-2015	**+14%**	**+9%**

Overview

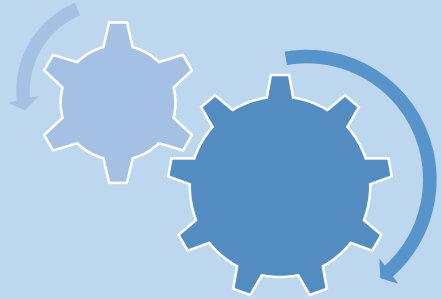
1. Introduction

- 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

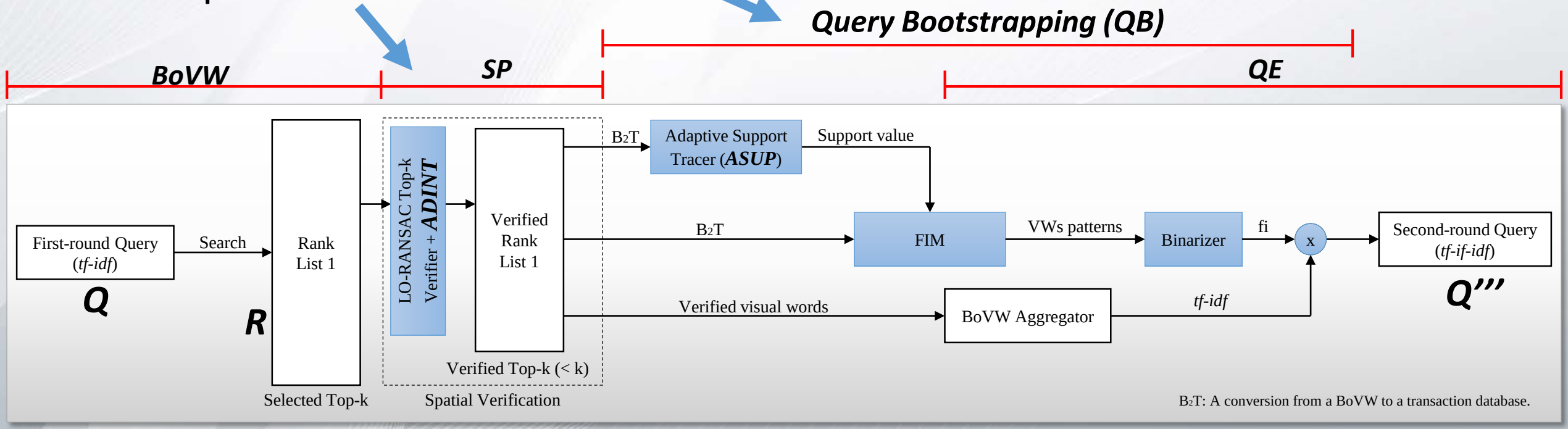
- Research achievements
- Pros and Cons

6. Future work

- Speed up
- Binary feature

3. Proposed methods

1. Visual word mining
2. Spatial verification

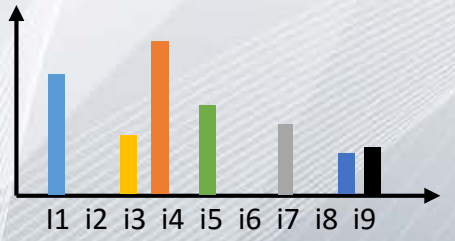


QB / QB + SP architecture diagram

Intro - Frequent Itemset mining (FIM)

T

Img. I_k	Trans. t_k
I_1	$t_1 = \{i_1, i_2, i_4, i_6\}$
I_2	$t_2 = \{i_2, i_5, i_8\}$
I_3	$t_3 = \{i_2, i_3, i_9\}$
I_4	$t_4 = \{i_1, i_2, i_4, i_7\}$
I_5	$t_5 = \{i_2, i_3, i_8\}$



FIM

Pattern	*support*
$\{i_2\}$	60%
$\{i_3\}$	40%
$\{i_8\}$	40%
$\{i_1, i_4\}$	40%
$\{i_3, i_8\}$	20%
$\{i_1, i_4, i_7\}$	20%
$\{i_2, i_3, i_9\}$	20%
$\{i_2, i_5, i_8\}$	20%
$\{i_1, i_2, i_4, i_6\}$	20%

P

Intro - Existing works

- Video mining [6]
  - Mining visual word motions into groups.
- Visual phrase mining [7]
  - Finding visual phrase lexicon.
  - Separating object out of background.
- Mining multiple queries [8]
  - Mining query patterns to better focus of targeted object.
- Mining for re-ranking and classification [9]
  - Voting image score by counting FIM patterns.

Our work closed to

[8] FIM for multiple images.
- But we are on the **result side.**

[9] FIM on result images.
- But we feed **back result** as AQE.

Non of them work directly on FIM for Query expansion!

Ref:
[6] T. Quack, V. Ferrari, and L.J.V. Gool, “Video mining with frequent itemset configurations.,” FIMI, pp.360–369, 2006.
[7] J. Yuan, Y. Wu, and M. Yang, “Discovery of collocation patterns: from visual words to visual phrases,” CVPR, pp.1–8, 2007.
[8] B. Fernando and T. Tuytelaars, “Mining multiple queries for image retrieval: On-the-fly learning of an object-specific mid-level representation,” ICCV, pp.2544–2551, 2013.
[9] W. Voravuthikunchai, B. Cr´emilleux, and F. Jurie, “Image re-ranking based on statistics of frequent patterns,” ICMR, pp.129–136, 2014.

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*** x ***tf-idf)***

R

FIM

Q'''

Back-projected visualization

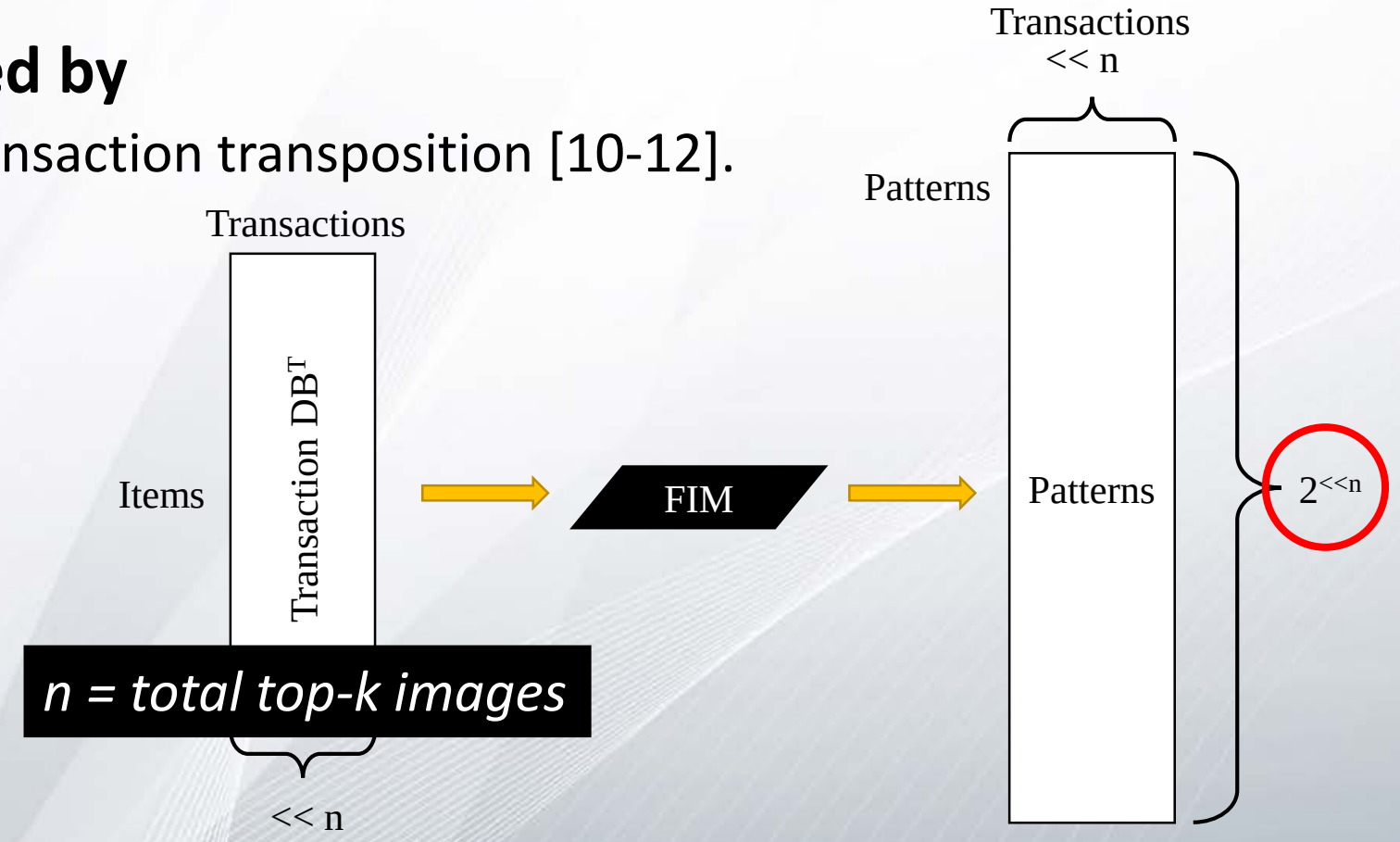
3.1 QB problem 1 (1)

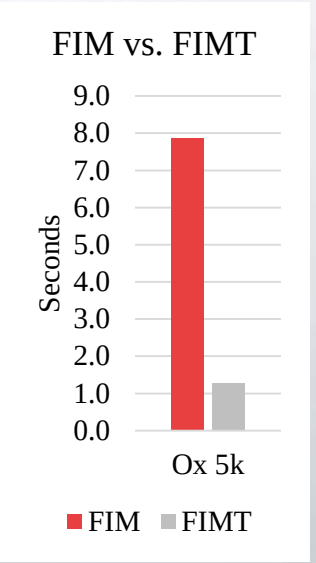
- FIM is designed for
  - Many transactions, Less items (n).
  - Total possible patterns **≈2^n**
- BoVW size up to 1 million, **slow down** FIM.
  - Less images, many words (n).

n = total non-zero visual words

3.1 QB problem 1 (2)

- **Helped by**
  - Transaction transposition [10-12].

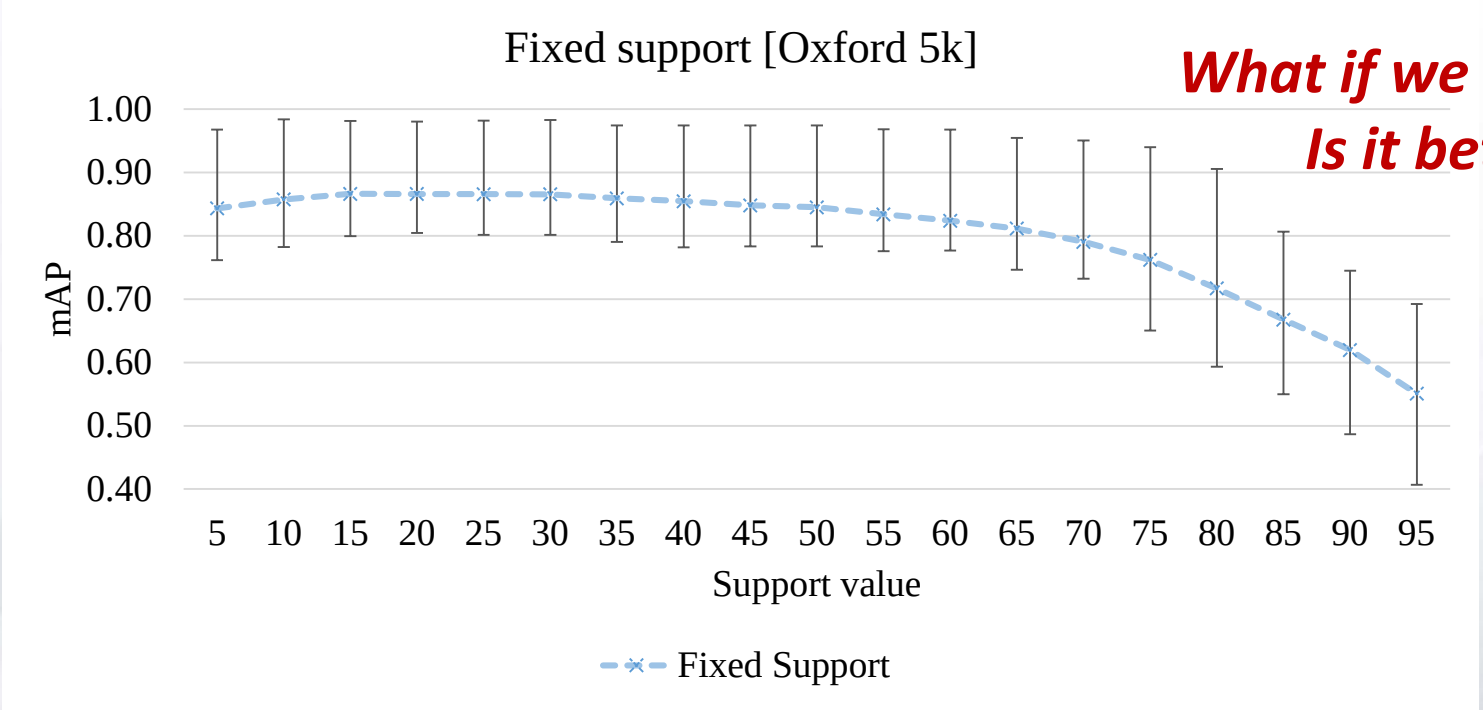




Faster!!

Ref:
[10] F. Rioult, J.F. Boulicaut, B. Cr´emilleux, and J. Besson, “Using transposition for pattern discovery from microarray data,” DMKD, pp.73–79, 2003.
[11] F. Rioult, “Mining strong emerging patterns in wide sage data,” 2004.
[12] F. Domenach and M. Koda, “Mining association rules using lattice theory (6th workshop on stochastic numerics),” 2004.

3.1 QB problem 2



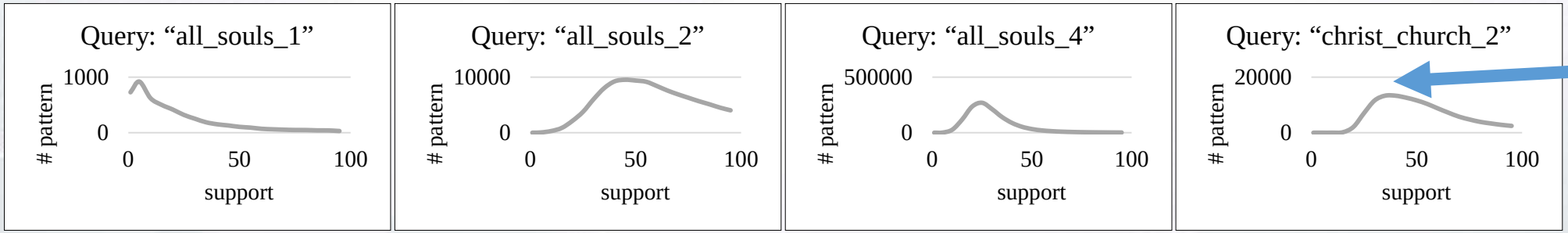
Fixed support value and its performance

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.

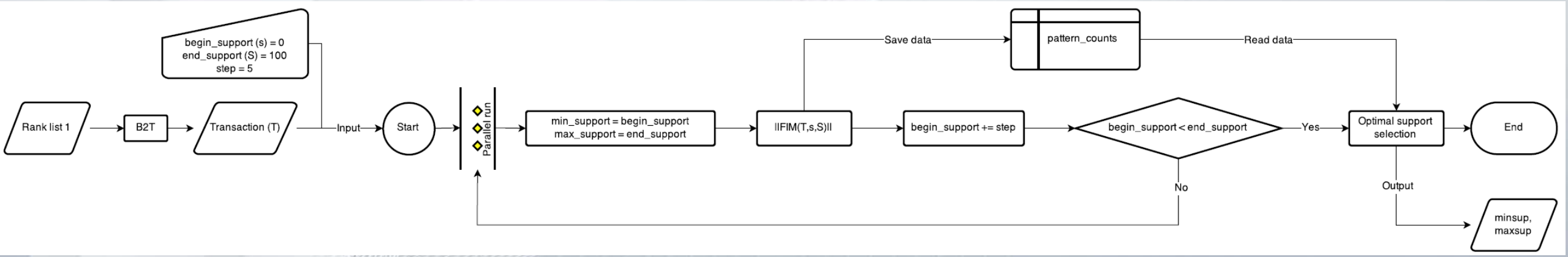
3.2 Contribution 2 - ASUP

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

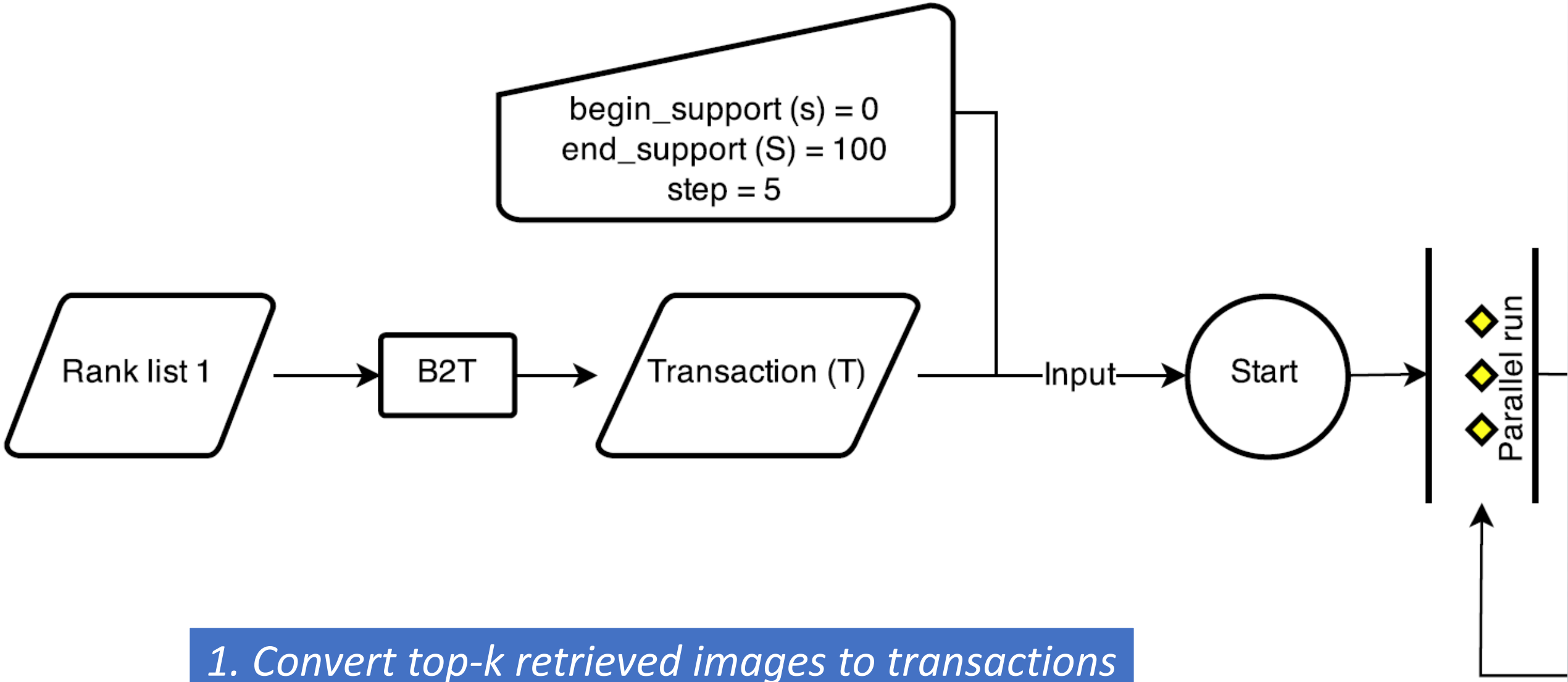


As we observed..
The optimal support is at the highest frequent patterns.

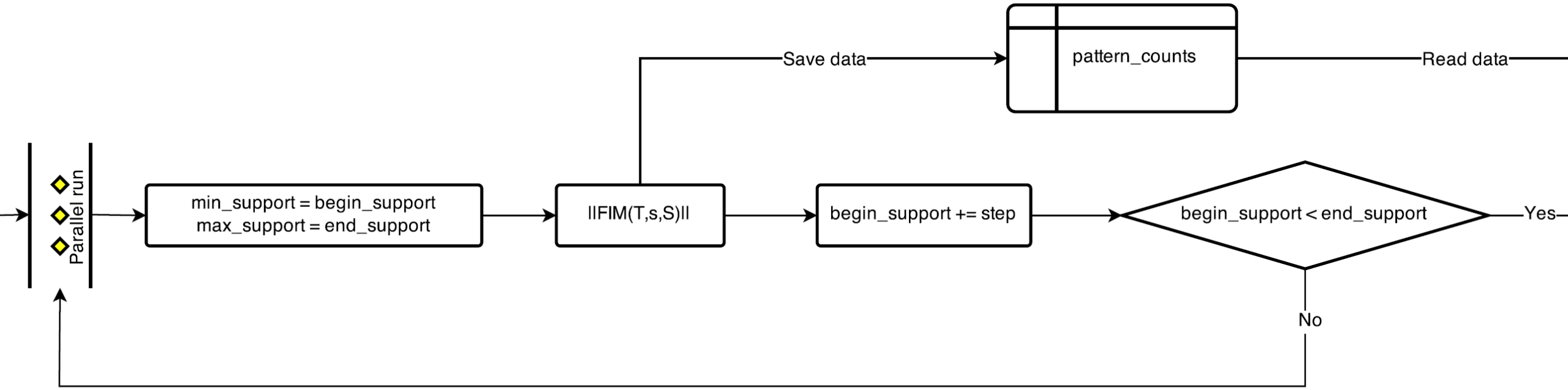
Pattern amount at each specific support range



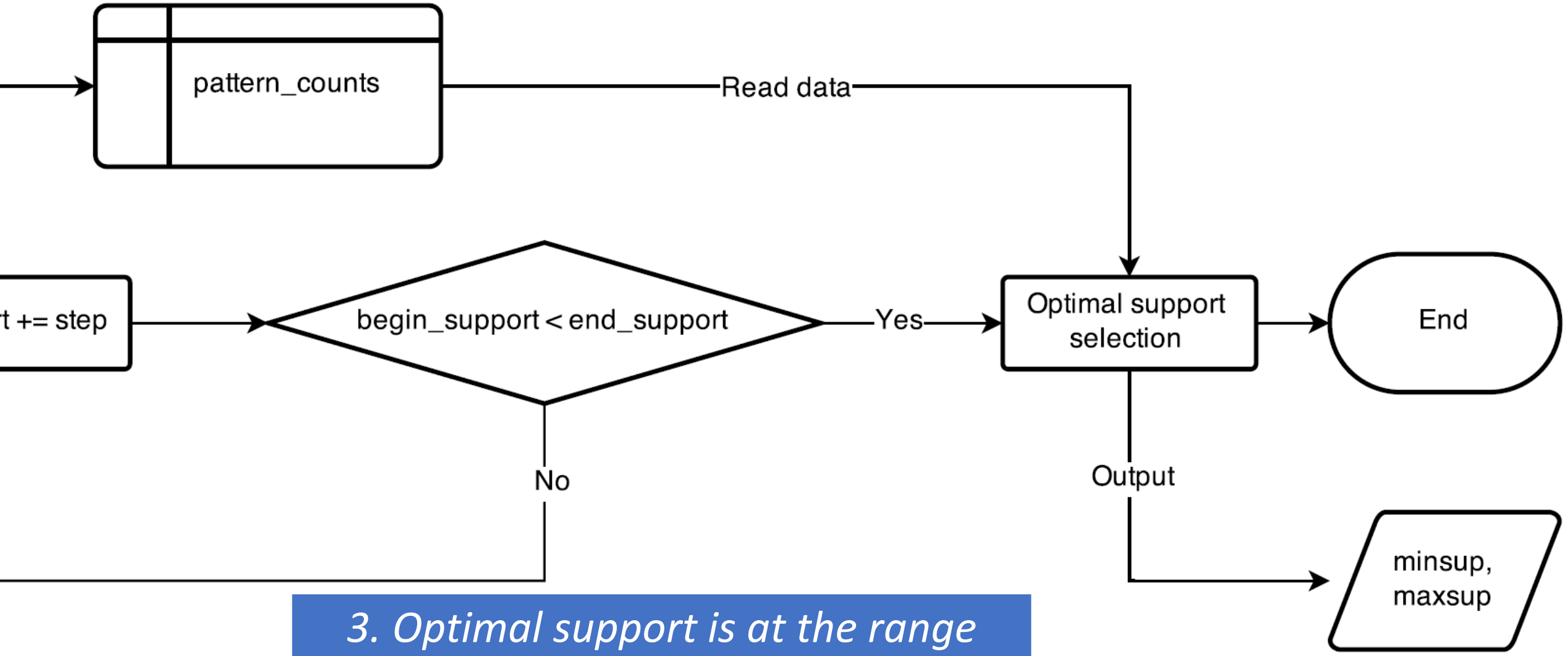
ASUP algorithm



1. Convert top-k retrieved images to transactions



2. Parallelizing FIM processes to get patterns at each support range (min to max support).



3. Optimal support is at the range which has the most frequent patterns.

3.2 ASUP problem

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

The rest of images are mostly a branches and a tree →

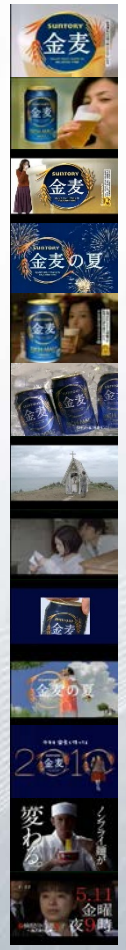
Round1 ***R*** (BoVW)

Round2 ***R*** (QB)

3.3 Contribution 3 - QB + SP

Q

R



Low threshold ***High threshold***

Accepting relevant images is fine!

Problem

How much inlier threshold should be set?

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

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.

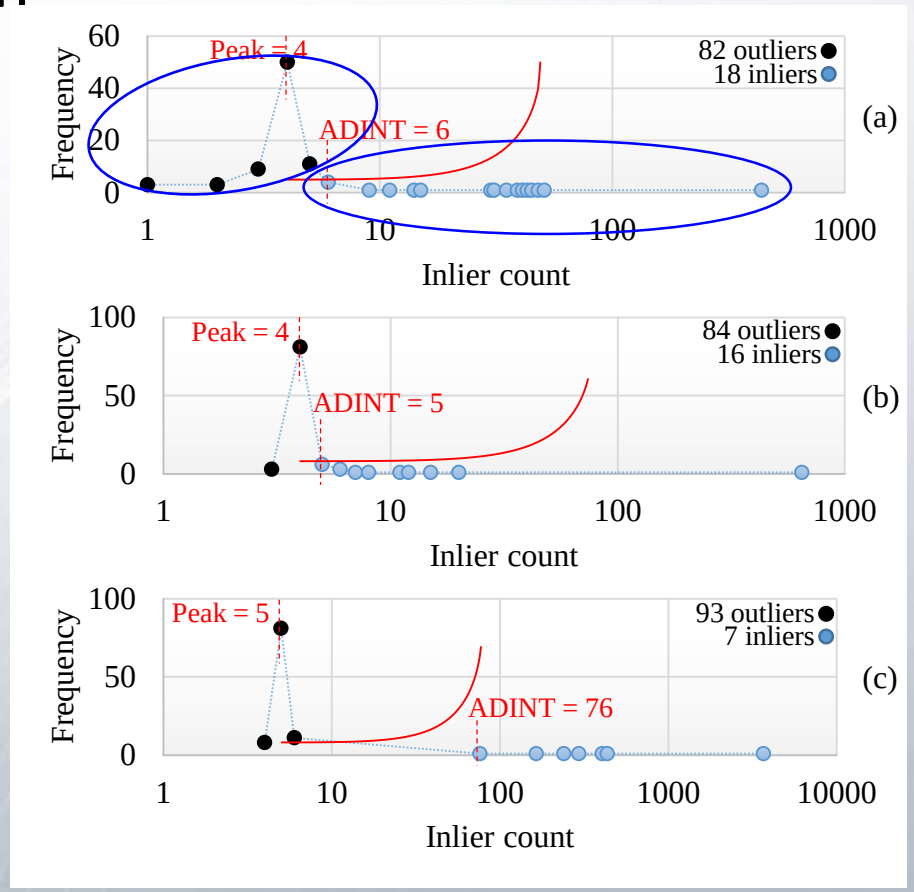
Inlier count histogram

Horizontal axis

Inlier count value provided by LO-RANSAC.

Vertical axis

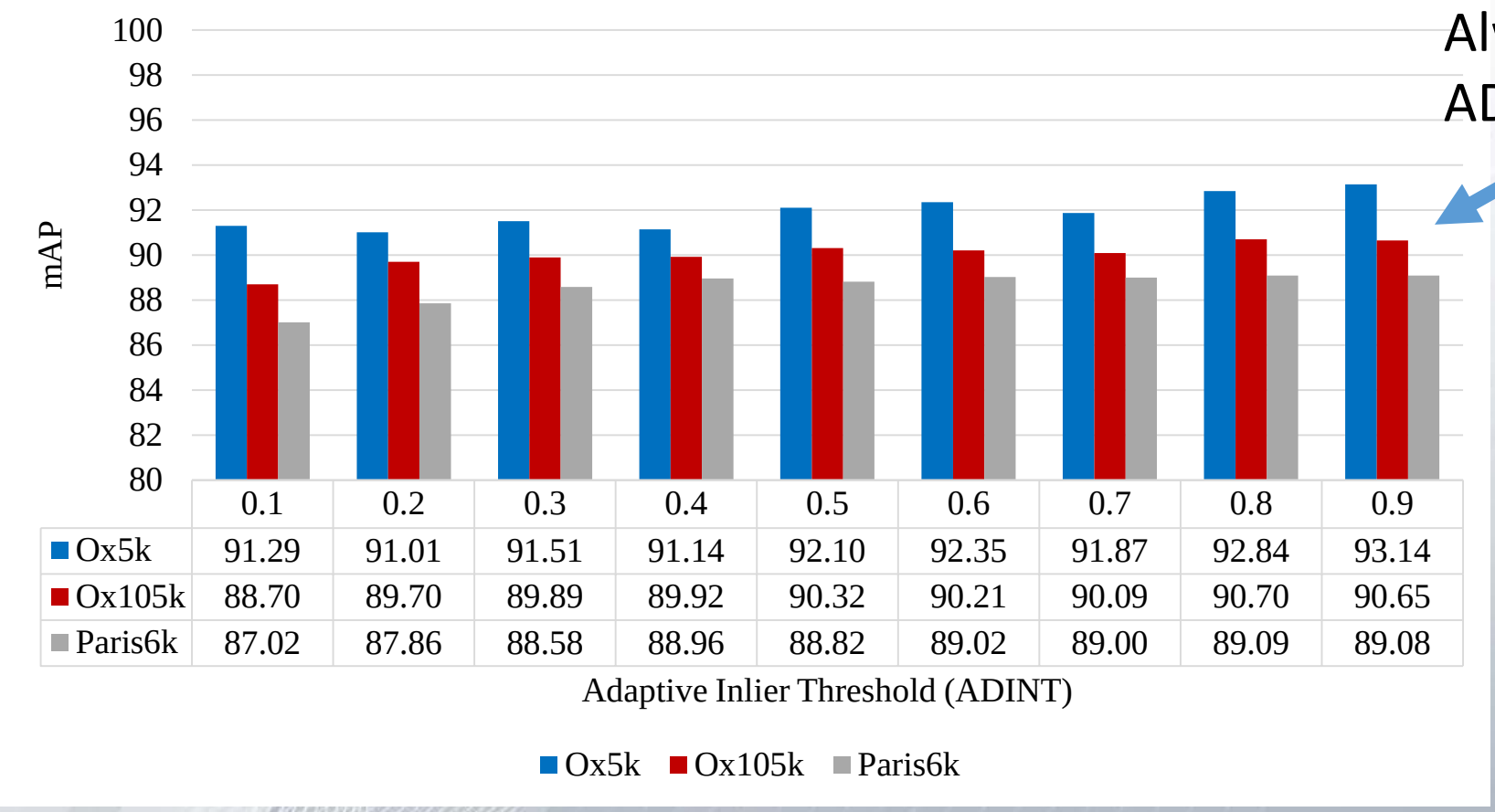
Total number of images.



Inlier count histogram

3.4 Contribution 4 – ADINT (2)

- Why ADINT ratio = 0.9?



	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Ox5k	91.29	91.01	91.51	91.14	92.10	92.35	91.87	92.84	93.14
Ox105k	88.70	89.70	89.89	89.92	90.32	90.21	90.09	90.70	90.65
Paris6k	87.02	87.86	88.58	88.96	88.82	89.02	89.00	89.09	89.08

ADINT ratio ~0.9
Always gives the best ADINT performance

3.4 Contribution 4 – ADINT (3)

- **ADINT** thresholding result

ADINT thresholding result

Color code

(blue)	Inlier count from LO-RANSAC
(red)	ADINT threshold
(orange)	Automated selected relevant images
(gray)	Ground truth

Overview

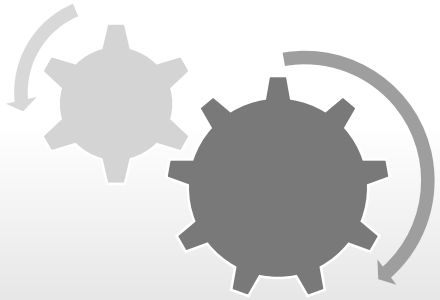
1. Introduction

- 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

6. Future work

- Speed up
- Binary feature

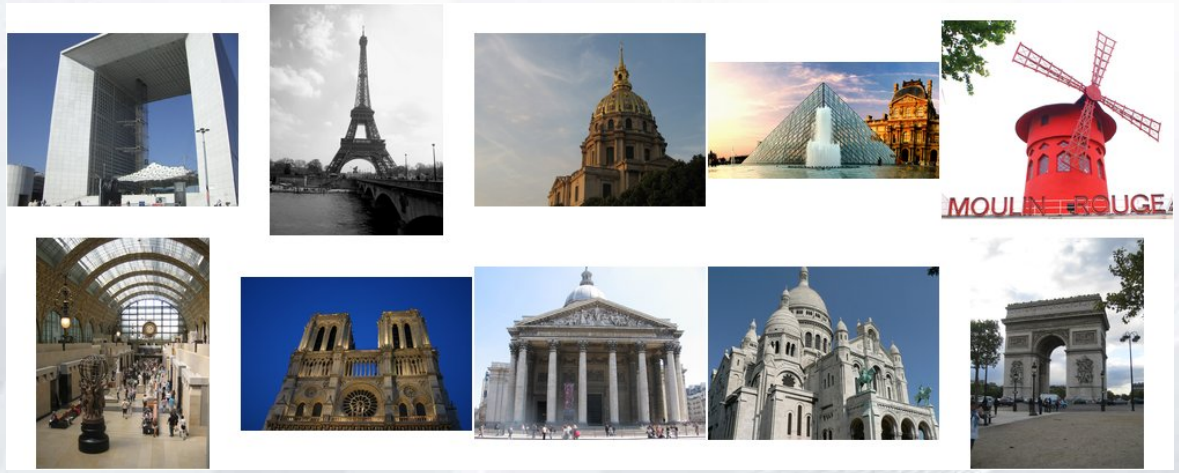
4. Experimental results (1)

- **Standard dataset**
  - Oxford building 5k and 105k.
  - Paris 6k.
  - 11 landmarks and locations (topic).
  - 5 different views on each topic.
  - Total 55 queries on each dataset.
- **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 matric.
  - And ground truth files obtained from the dataset provider.

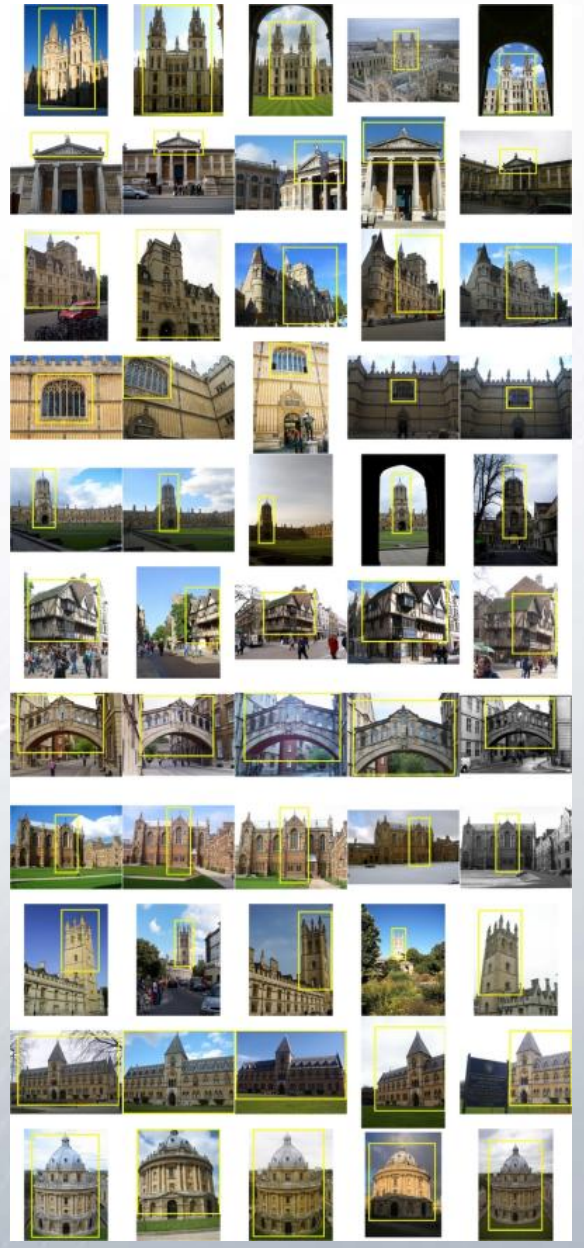
Ref:
Oxford dataset: http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/
Paris dataset: http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/
MIRFlickr1M dataset: http://press.liacs.nl/mirflickr/mirdownload.html

4. Experimental results (2)

- Dataset examples



Paris landmarks

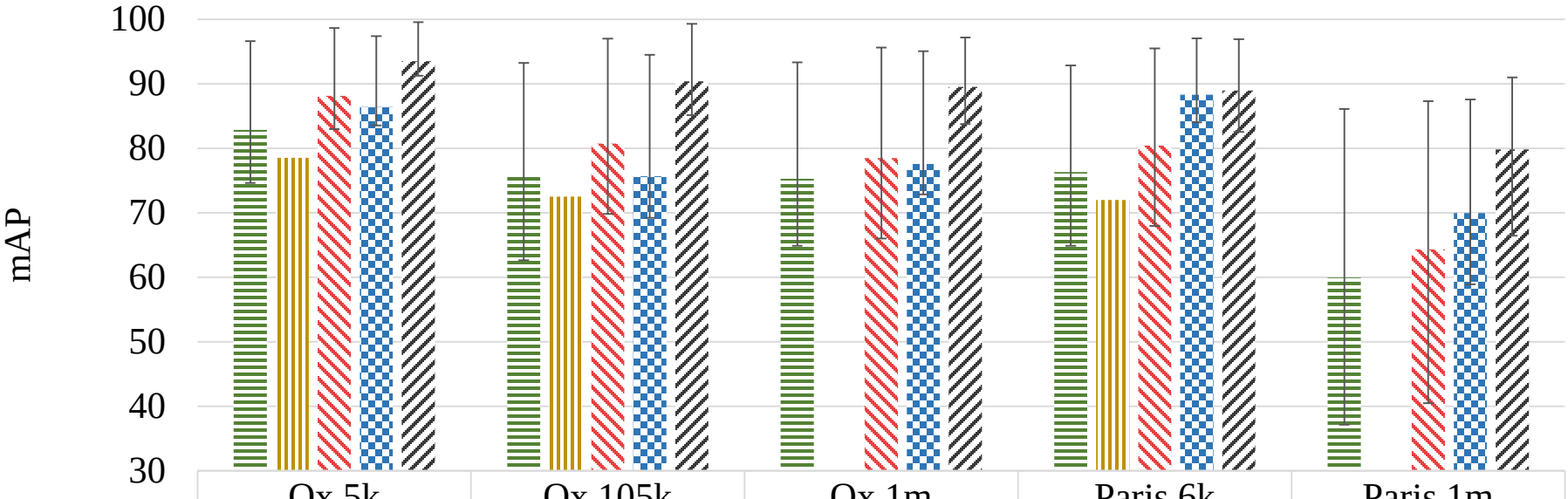


Oxford buildings

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



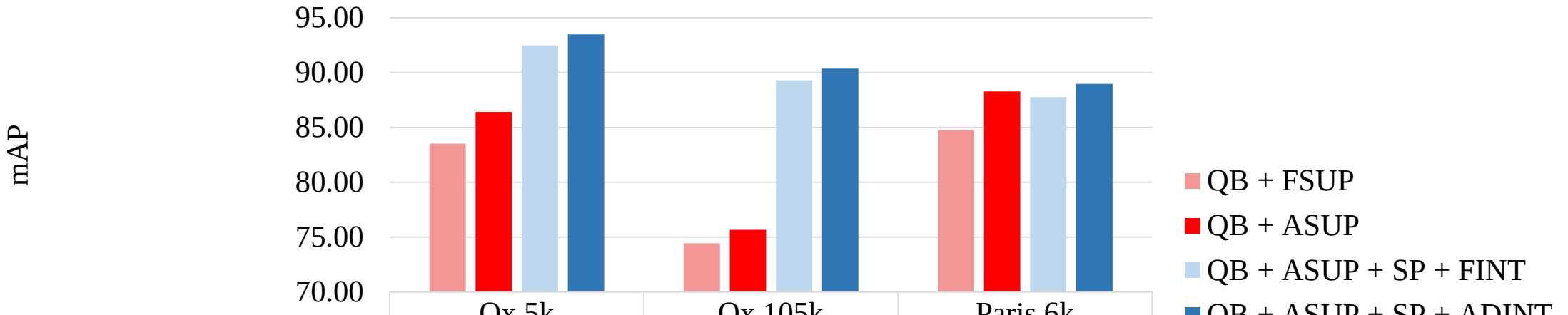
	Ox 5k	Ox 105k	Ox 1m	Paris 6k	Paris 1m
BoVW	82.84	75.66	75.28	76.33	59.95
AQE [28]	78.50	72.50		72.00	
AQE	88.12	80.71	78.48	80.44	64.32
QB	86.41	75.67	77.56	88.28	69.94
QB + SP	93.49	90.36	89.52	88.96	79.81

mAP for each method and dataset

Ref:
[28] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, "Total recall II: Query expansion revisited," CVPR, pp.889–896, 2011.

4.2 Contributions comparison

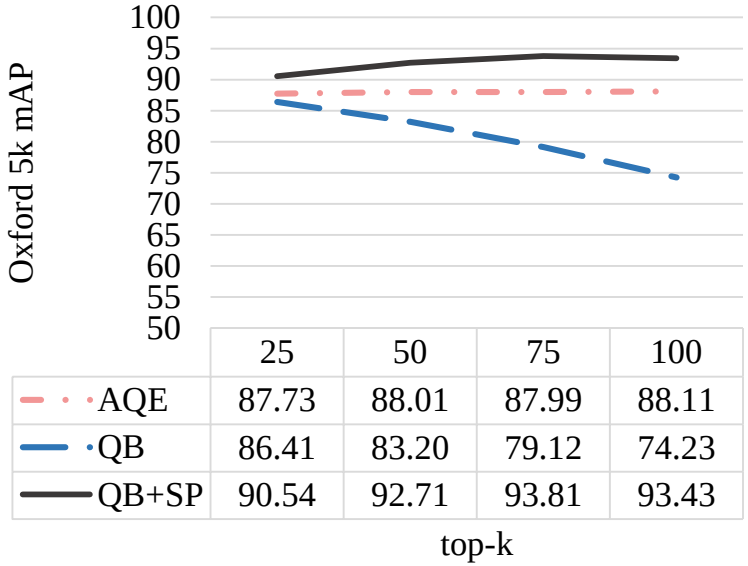
- Notation of our proposed methods
  - QB = (QB + ASUP)
  - QB + SP = (QB + ASUP) + (SP + ADINT)



	Ox 5k	Ox 105k	Paris 6k
QB + FSUP	83.52	74.43	84.77
QB + ASUP	86.41	75.67	88.28
QB + ASUP + SP + FINT	92.48	89.31	87.76
QB + ASUP + SP + ADINT	93.49	90.36	88.96

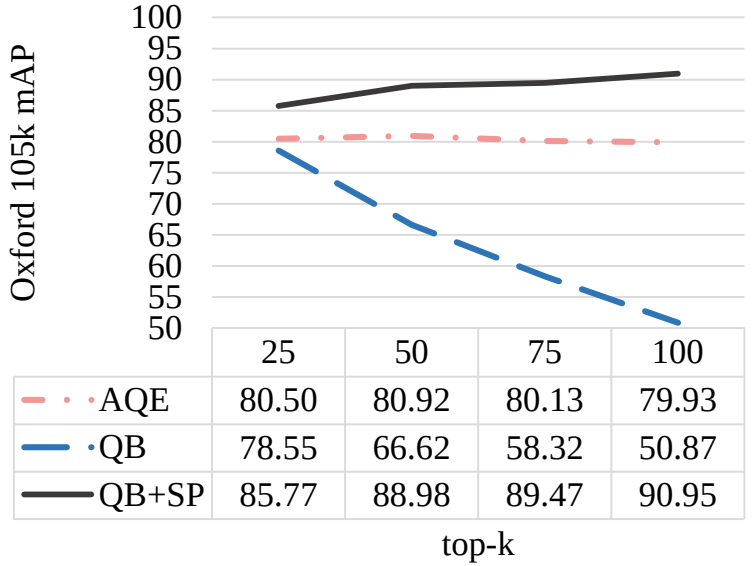
The performance comparison between our contributions

4.3 Impact of Top-k relevant images



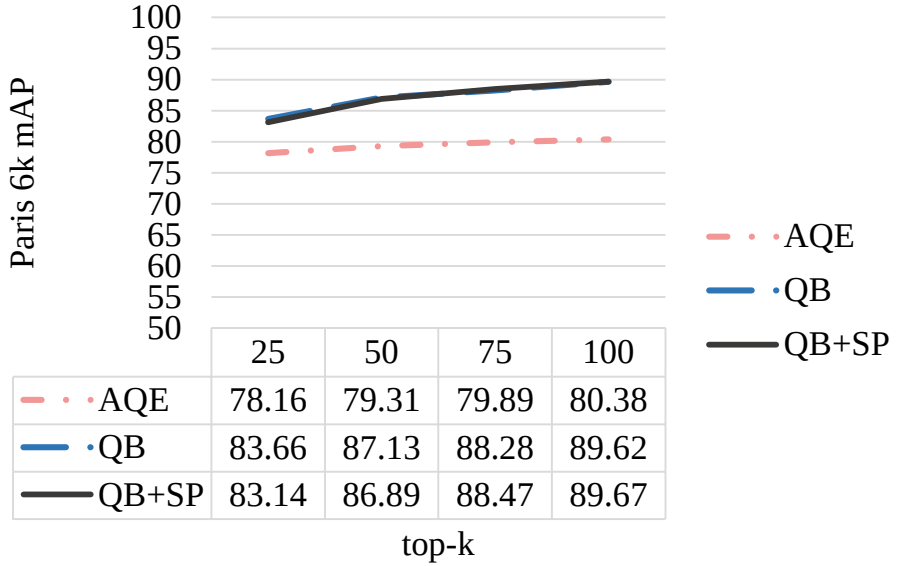
	25	50	75	100
AQE	87.73	88.01	87.99	88.11
QB	86.41	83.20	79.12	74.23
QB+SP	90.54	92.71	93.81	93.43

top-k



	25	50	75	100
AQE	80.50	80.92	80.13	79.93
QB	78.55	66.62	58.32	50.87
QB+SP	85.77	88.98	89.47	90.95

top-k



	25	50	75	100
AQE	78.16	79.31	79.89	80.38
QB	83.66	87.13	88.28	89.62
QB+SP	83.14	86.89	88.47	89.67

top-k

mAP vs. total number of retrieved 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

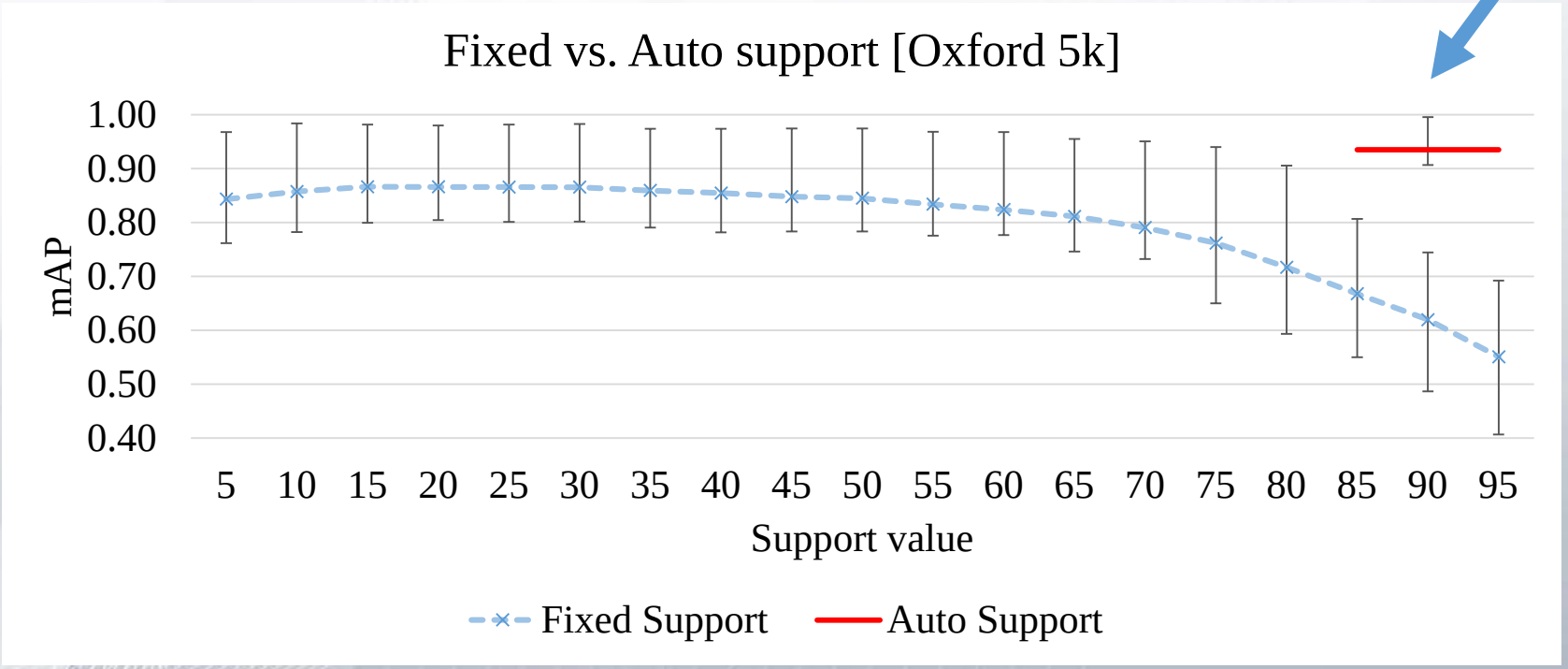
Why QE/QB did not fail on Paris6k?

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**

$\Delta(\min, \mathbf{A})$ is

how much **ADINT** better than a minimum of FINT.

$\Delta(\max, \mathbf{A})$ is

how much **ADINT** better than a maximum of FINT.

Inlier Threshold	AQE (mAP %)			QB + SP (mAP %)		
	Ox5k	Ox105k	Paris6k	Ox5k	Ox105k	Paris6k
3	88.11	79.69	80.44	74.39	50.95	89.66
5	88.60	80.72	80.13	85.47	68.44	89.32
7	87.87	81.86	79.19	92.48	89.31	87.76
9	87.32	81.15	78.87	91.64	88.28	86.62
11	87.13	80.85	78.70	90.77	87.56	85.88
A	**87.88**	**81.85**	**78.70**	**93.49**	**90.36**	**88.96**
$\Delta(\min, \mathbf{A})$	0.75	2.16	0.00	19.10	39.41	3.08
$\Delta(\max, \mathbf{A})$	-0.72	-0.01	-1.74	1.01	1.05	-0.70

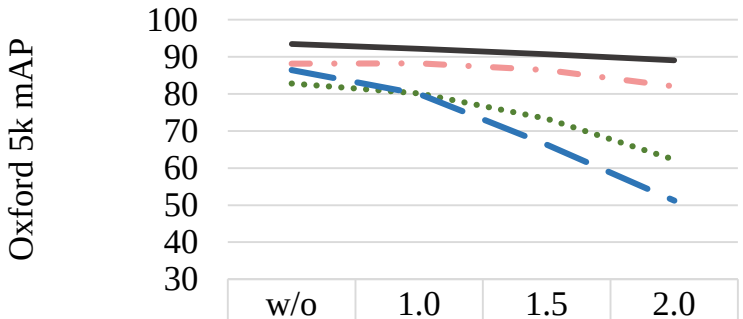
ADINT vs. FINT performance

Result:

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

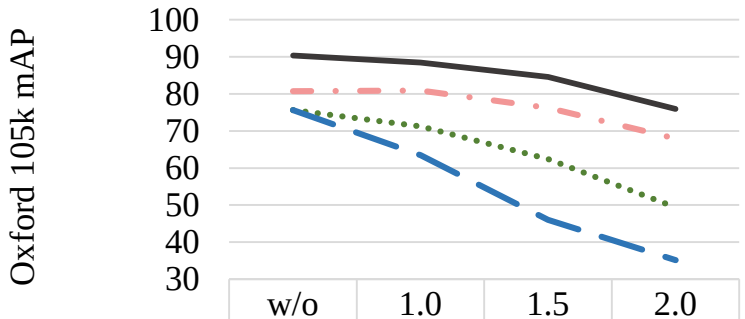
4.5 Impact of low quality query (noise test)

Sample query image with noise @sigma = 2.0



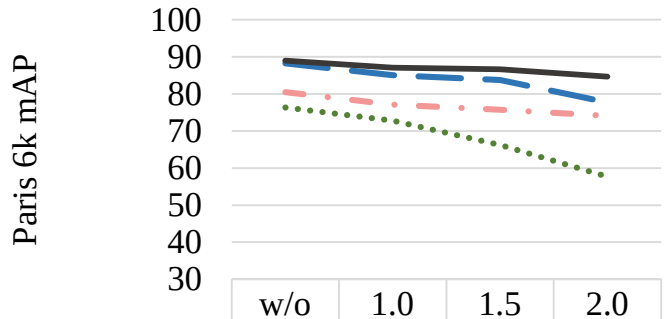
	w/o	1.0	1.5	2.0
Baseline	82.84	80.17	73.32	62.28
AQE	88.12	88.24	86.43	82.02
QB	86.41	79.94	66.29	51.18
QB + SP	93.49	92.15	90.71	89.03

Gaussian sigma (σ)



	w/o	1.0	1.5	2.0
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

Gaussian sigma (σ)



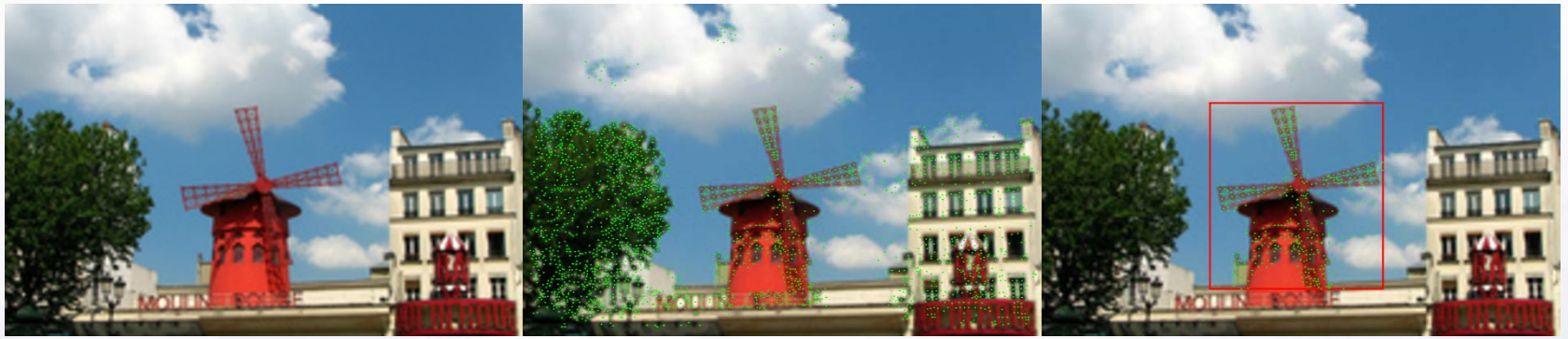
	w/o	1.0	1.5	2.0
Baseline	76.33	72.82	66.21	57.72
AQE	80.44	77.14	75.77	74.05
QB	88.28	85.01	83.77	77.70
QB + SP	88.96	87.11	86.61	84.64

Gaussian sigma (σ)

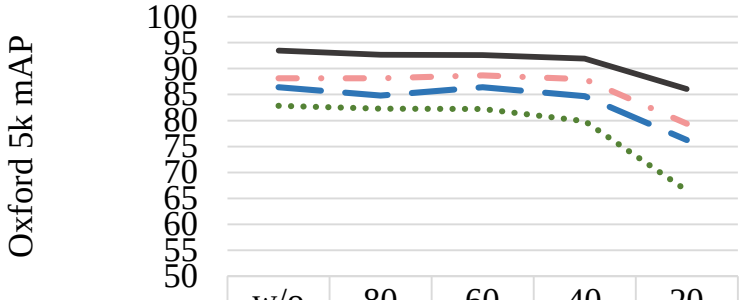
Baseline
AQE
QB
QB + SP

mAP vs. noise level

4.5 Impact of low quality query (scale test)

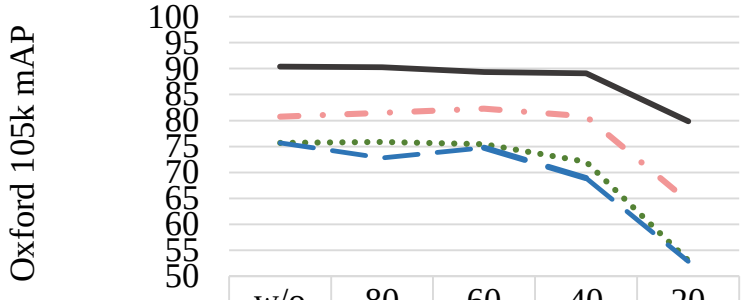


Sample query image with scale of 20% of original



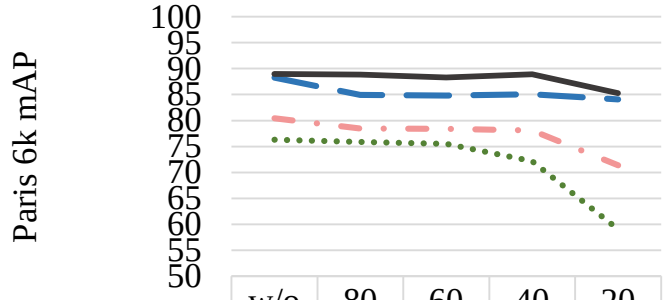
	w/o	80	60	40	20
Baseline	82.84	82.29	82.25	79.89	66.47
AQE	88.12	88.14	88.70	87.93	79.37
QB	86.41	84.78	86.39	84.69	76.22
QB + SP	93.49	92.68	92.58	91.92	86.07

Query scale (%)



	w/o	80	60	40	20
Baseline	75.66	75.85	75.45	72.04	53.07
AQE	80.71	81.51	82.28	80.80	64.46
QB	75.67	72.77	74.74	68.93	52.86
QB + SP	90.36	90.28	89.31	89.12	79.82

Query scale (%)



	w/o	80	60	40	20
Baseline	76.33	75.90	75.47	72.17	59.05
AQE	80.44	78.46	78.38	78.09	71.40
QB	88.28	84.91	84.81	85.04	84.05
QB + SP	88.96	88.84	88.31	88.93	85.29

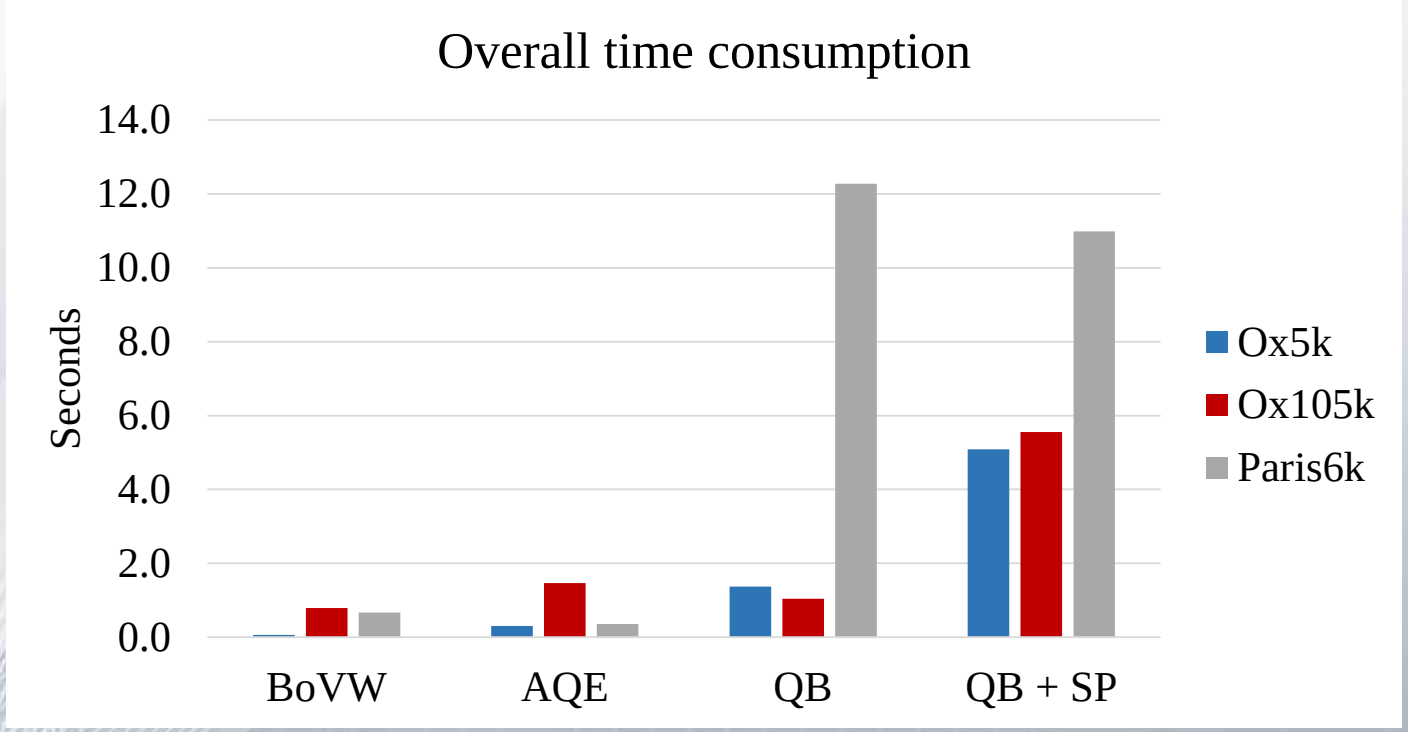
Query scale (%)

Baseline
AQE
QB
QB + SP

mAP vs. image scale

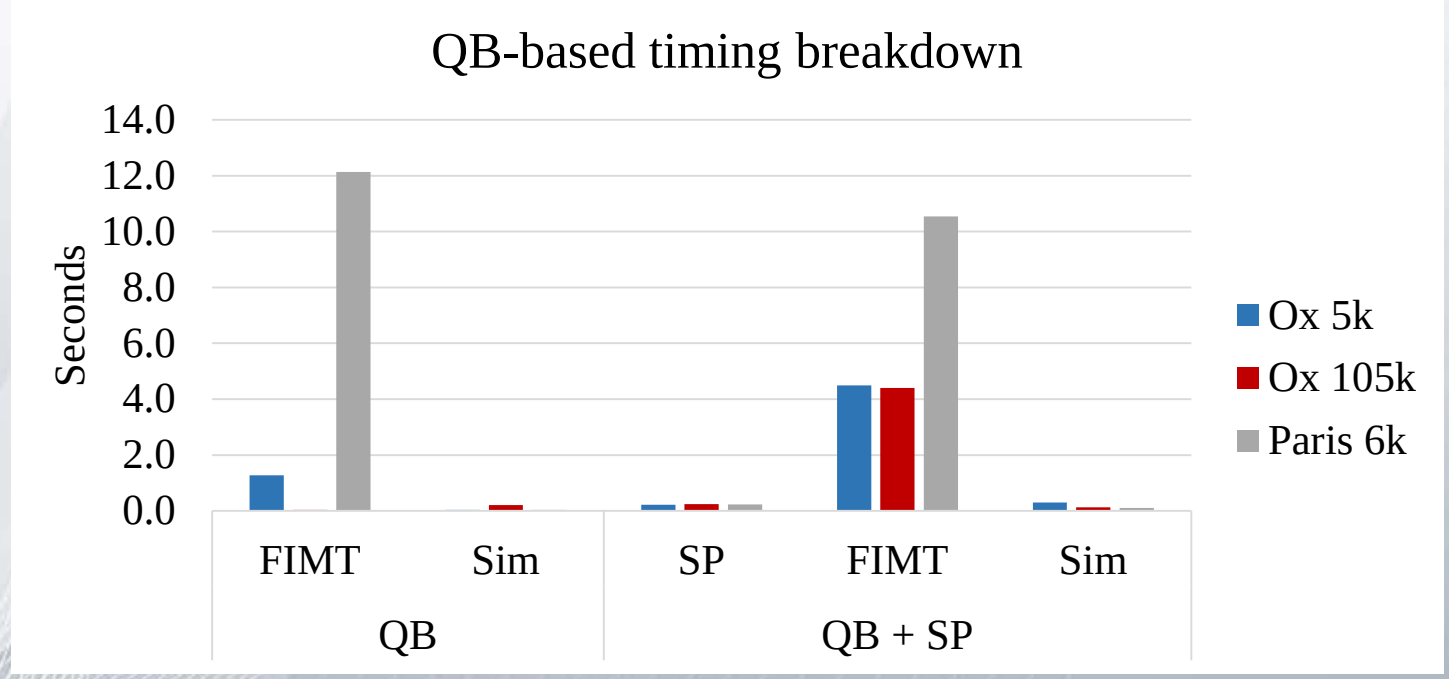
4.6 Time consumption

- **Overall time consumption**
  - Fast with BoVW, and AQE
  - Slow with QB, and QB + SP

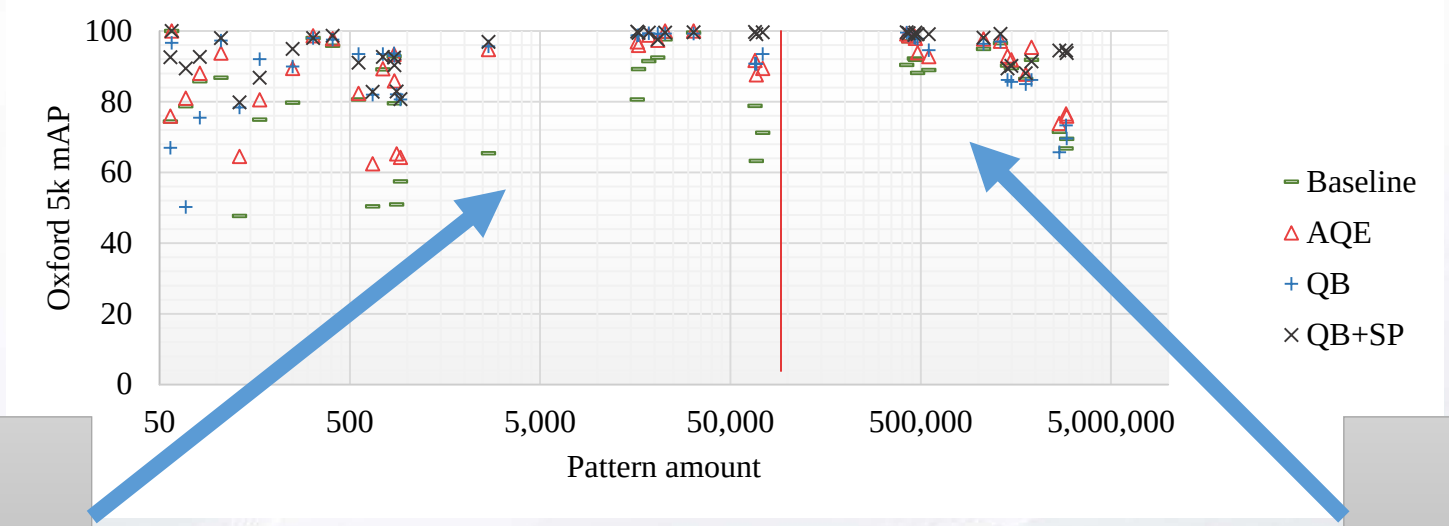


4.6 Time consumption - breakdown

- FIM-based methods are **QB** and **QB + SP**
- **Result:**
  - FIM is the most slowest part, why?



4.6.1 Colossal pattern[13]



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)

	Type	#Topics	BoVW	QB				QB+SP			
				FIM^T(s)	Precision(%)			FIM^T(s)	Precision(%)		
			mAP(%)		mAP(%)	SD(±%)	mAP+(%)		mAP(%)	SD(±%)	mAP+(%)
Ox 5k	Easy	40	81.26	0.075	85.51	21.02	4.25	0.166	92.69	14.25	**11.43**
	Hard	15	87.06	4.471	88.79	10.97	1.72	16.037	95.64	4.07	**8.58**
Ox 105k	Easy	40	73.94	0.011	73.99	29.94	0.05	0.066	90.77	15.95	**16.83**
	Hard	15	80.24	0.109	80.13	13.81	-0.11	15.949	89.28	9.19	**9.04**
Paris 6k	Easy	25	71.09	0.922	86.53	9.23	**15.44**	0.363	86.17	9.39	15.08
	Hard	30	80.69	21.475	89.74	15.37	9.05	19.030	91.28	12.28	**10.59**

QB + SP improve "**Easy**" query very well. And FIMT time usage on "**Easy**" is not much.

Ref:
[13] F. Zhu, X. Yan, J. Han, P.S. Yu, and H. Cheng, "Mining colossal frequent patterns by core pattern fusion," ICDE, pp.706–715, 2007.

4.7 Result

(a) Query (b) BoVW results.

(c) AQE results.

(d) QB + SP results.

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 others

Overview

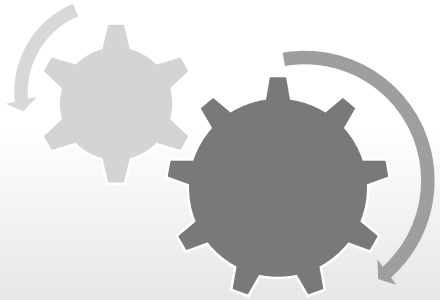
1. Introduction

- 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

6. Future work

- Speed up
- Binary feature

5. Conclusion

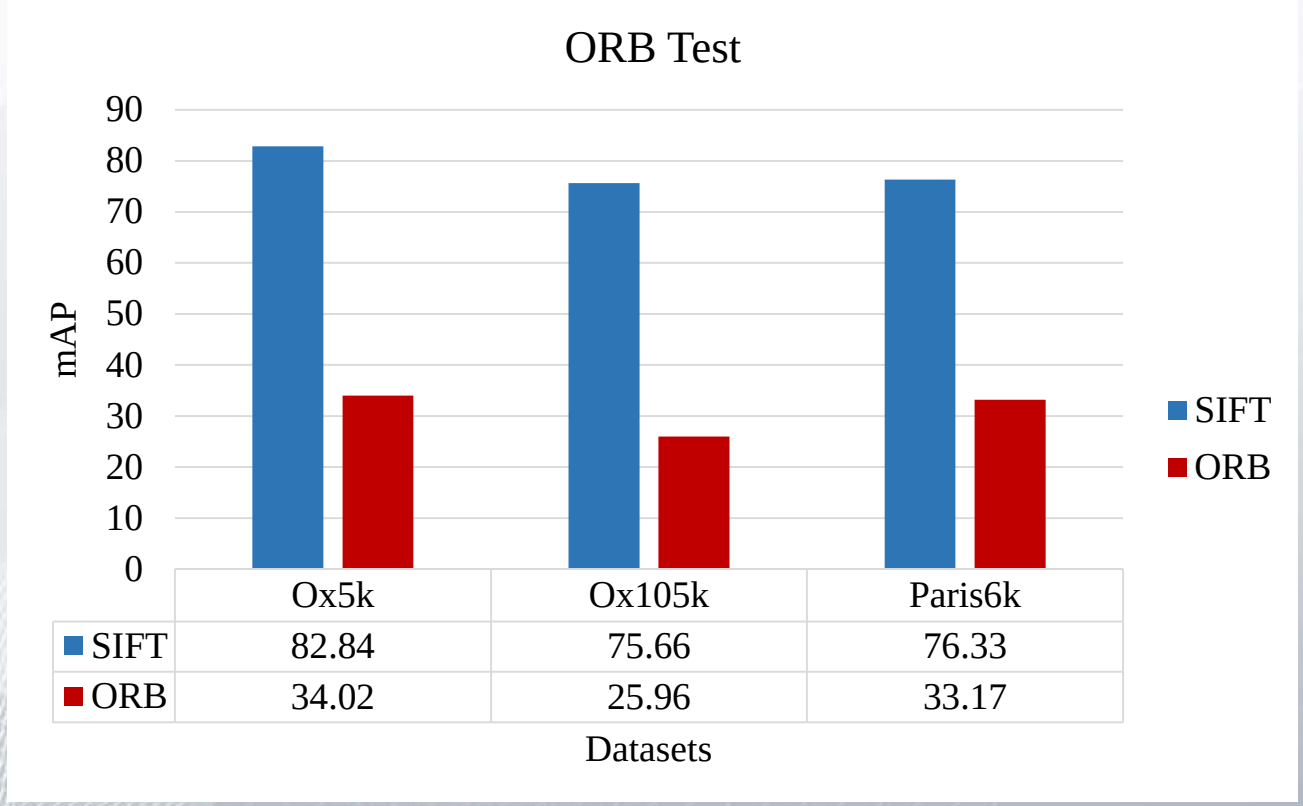
- ***We proposed***
  - "Query Bootstrapping (QB)" as visual mining technique for query expansion.
  - The way to integrate "Spatial Verification (SP)" for such mining.
- ***The important parameters are automatically determined.***
  - Adaptive support (ASUP) for FIM.
  - Adaptive inlier threshold (ADINT) for LO-RANSAC.
- ***Achievements***
  - Our methods reach the highest performance on all datasets.
  - Very high robustness on difficult cases of query quality are proved.
- ***Slowness of FIM***
  - Too many relevant images create too many patterns.
  - Switch back to AQE, if the high performance result is detected.

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

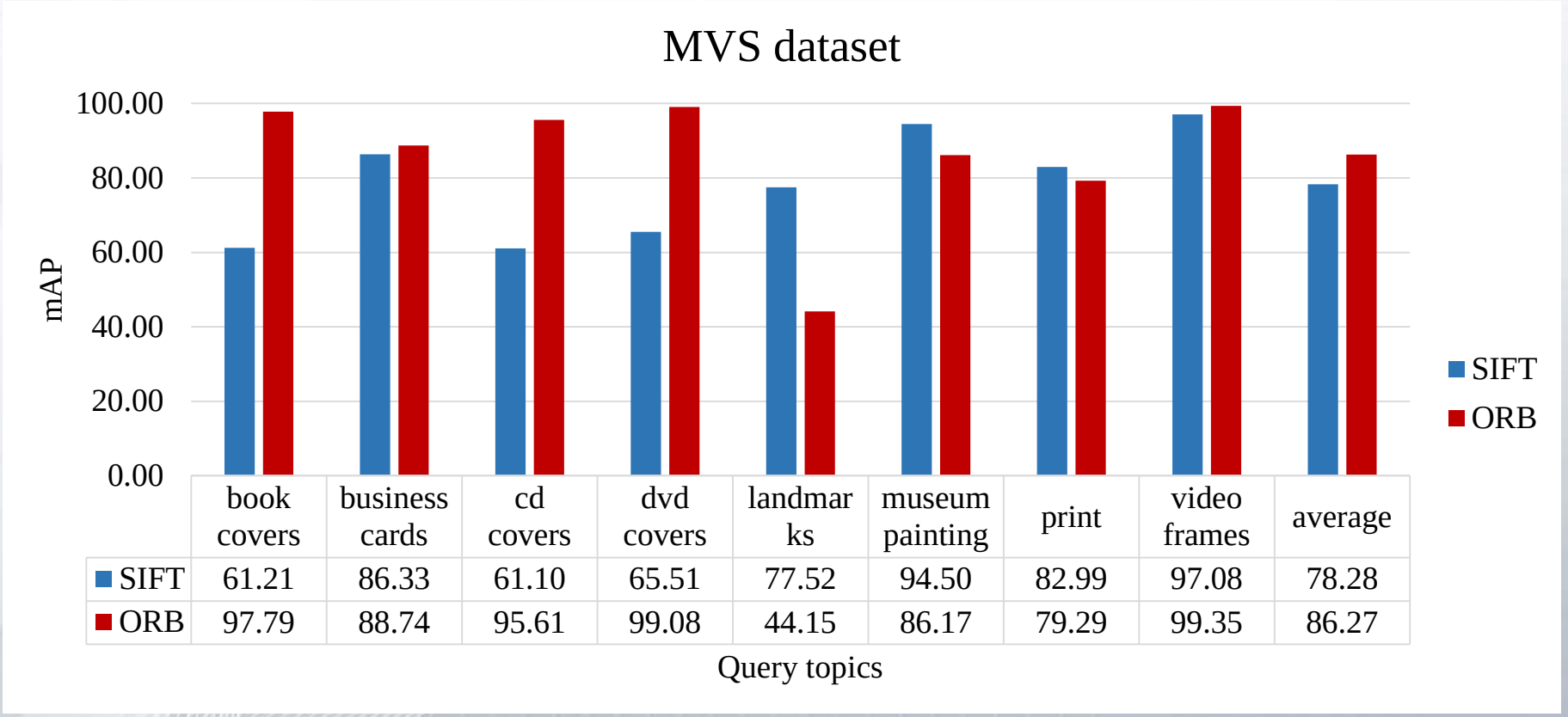
- We also did experiments on binary feature.
  - ORB feature



	Ox5k	Ox105k	Paris6k
SIFT	82.84	75.66	76.33
ORB	34.02	25.96	33.17

6. Future work

- ORB experiments on MVS dataset



	book covers	business cards	cd covers	dvd covers	landmarks	museum painting	print	video frames	average
SIFT	61.21	86.33	61.10	65.51	77.52	94.50	82.99	97.08	78.28
ORB	97.79	88.74	95.61	99.08	44.15	86.17	79.29	99.35	86.27

Overview and Q/A

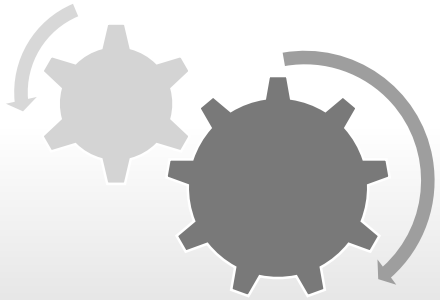
1. Introduction

- 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