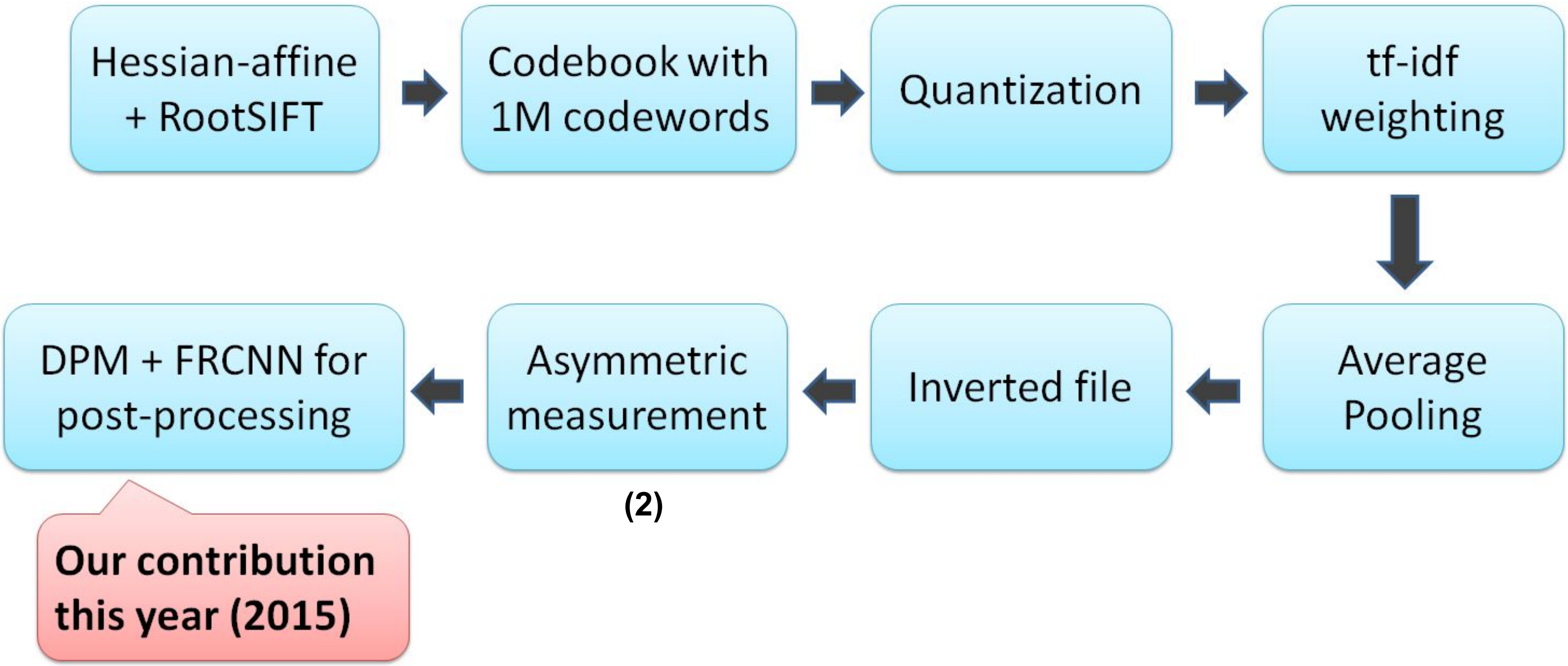
Fast RCNN and DPM As a Combination for Spatial Reranking

Vinh-Tiep Nguyen(2)(3), Duy-Dinh Le(1), Amaia Salvador(3), Caizhi-Zhu(5), Dinh-Luan Nguyen(3), Minh-Triet Tran(3), Thanh Ngo Duc(2), Duc Anh Duong(2), Shin'ichi Satoh(1), Xavier Giro-i-Nieto(4)

(1) National Institute of Informatics, Japan (NII)
(2) VNU-HCMC - University of Information Technology, Vietnam (UIT-HCM)
(3) VNU-HCMC - University of Science, Vietnam (HCMUS-HCM)
(4) Universitat Politecnica de Catalunya (UPC)
(5) Nagoya University , Japan (NU)

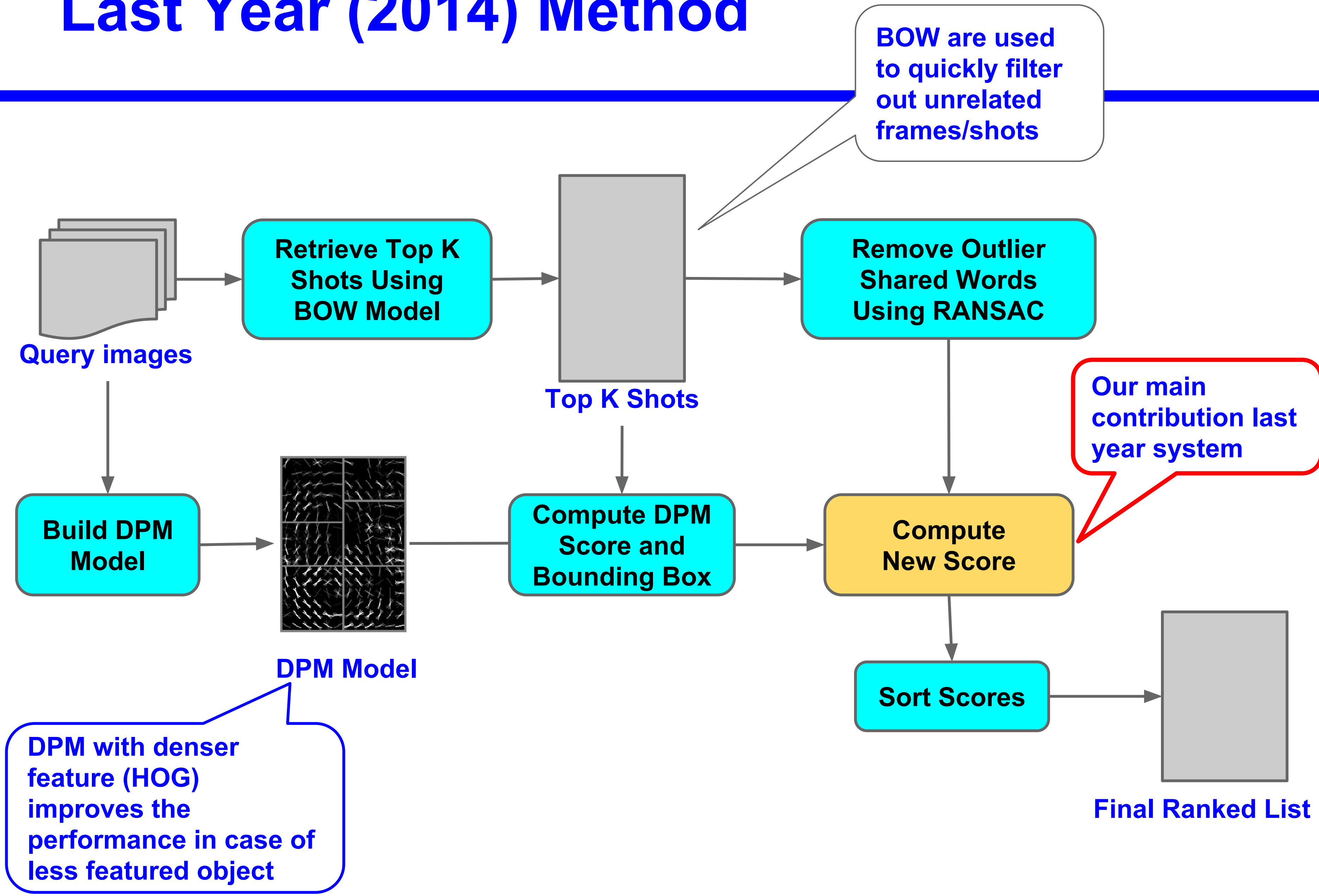
General Instance Search Framework [1]



(1) ***Three things everyone should know to improve object retrieval,*** R. Arandjelović, A. Zisserman, CVPR 2012

(2) ***Query-adaptive asymmetrical dissimilarities for visual object retrieval,*** Cai-Zhi Zhu, Hervé Jégou, Shin'Ichi Satoh, ICCV 2013.

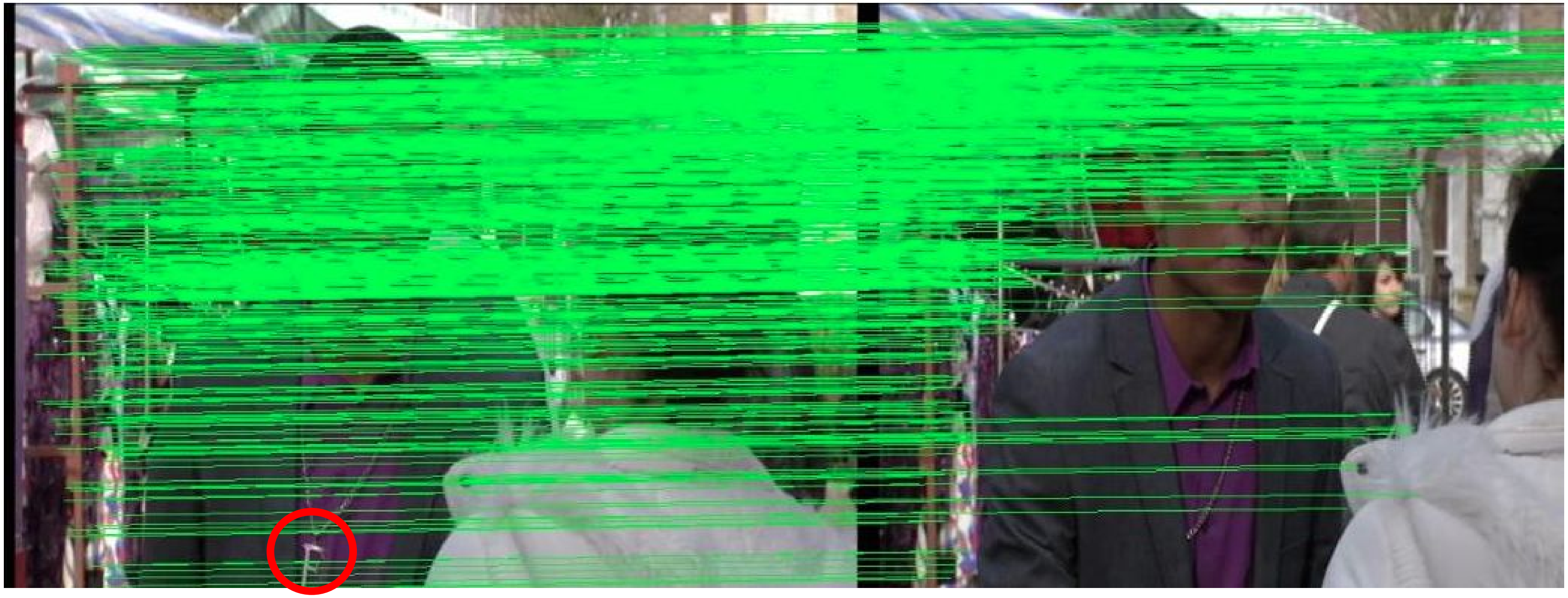
Last Year (2014) Method



BOW is Good for Rich Featured Objects

But … Not for Less Textured Objects

- Small objects

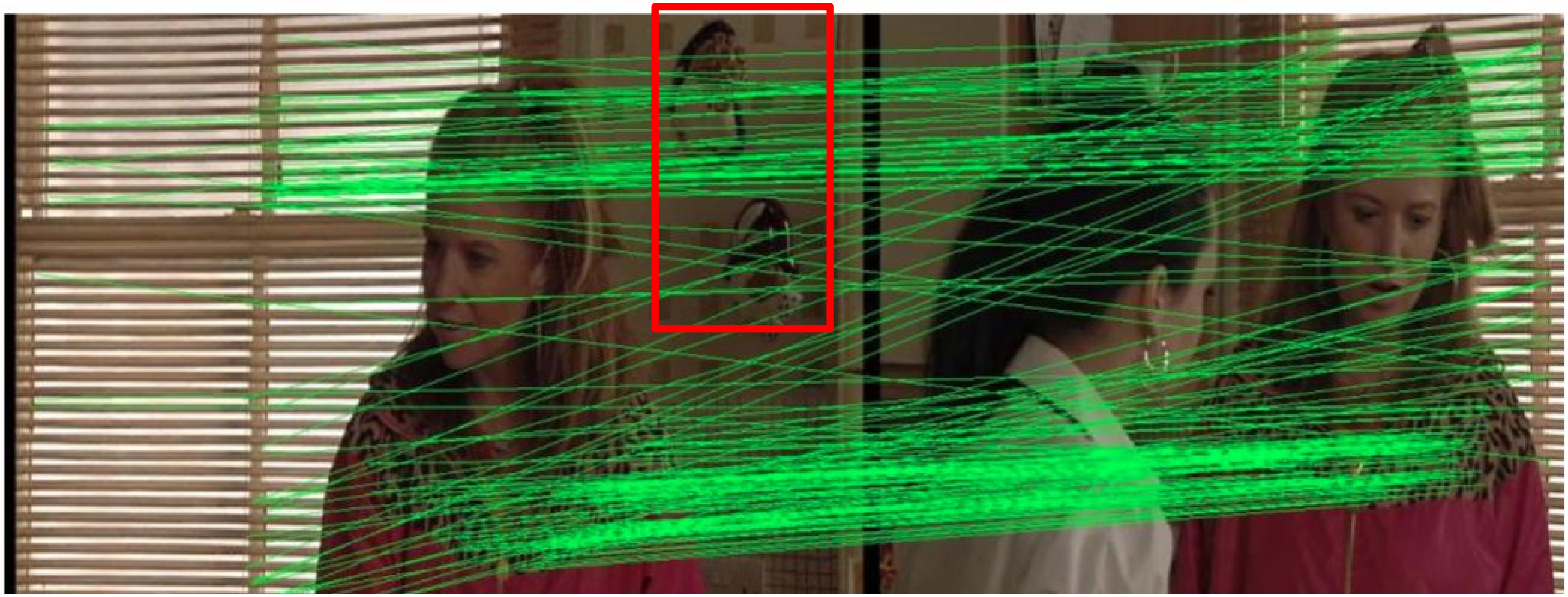


Query

Background Dominated Query Object

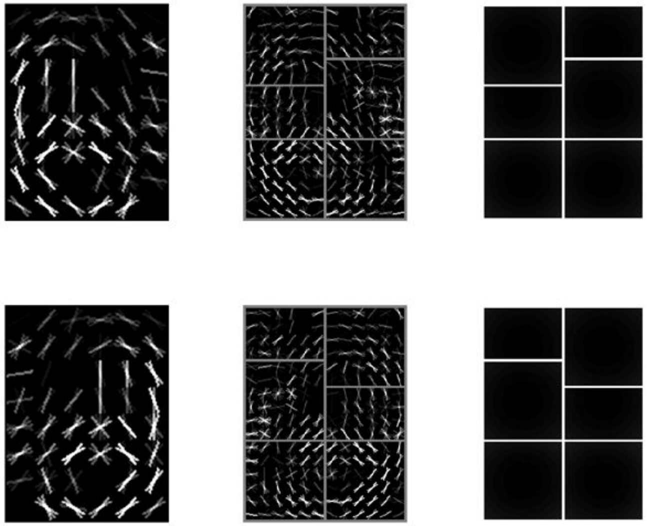
- Burstiness

Query



DPM-based Object Localizer

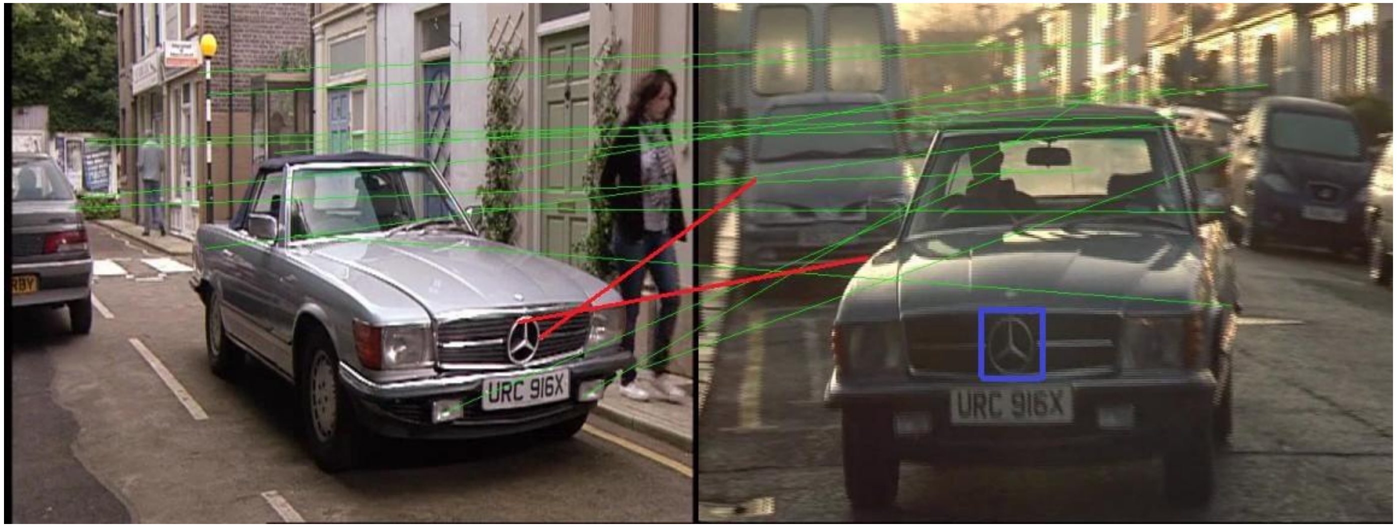
Query 9109



Visualization of DPM model for query 9109

- Benefit:
  - Model query object as a shape structure.
  - Work well with small and texture-less object.
  - Augment bounding box information.

DPM Is Good for Less Textured Objects

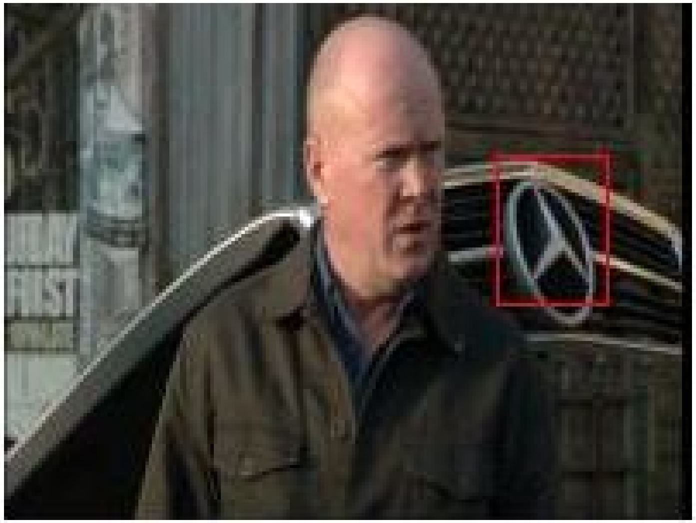


Wrong shared words case

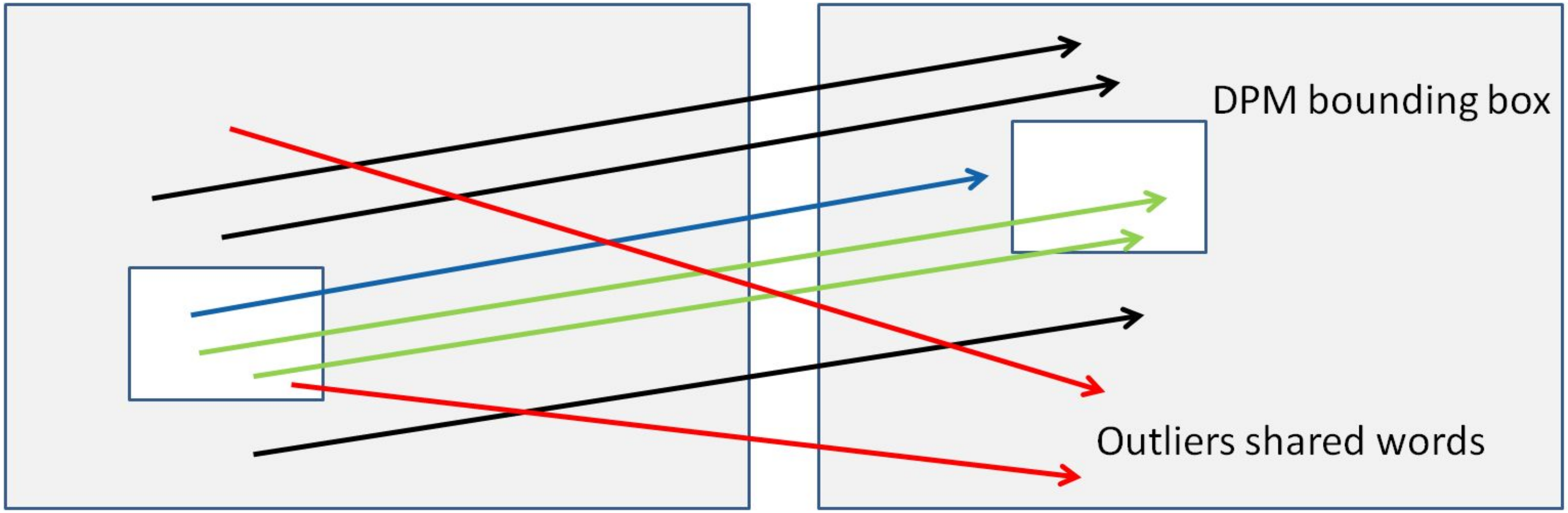
No shared word case

DPM: The Good and The Bad

- DPM is based on gray scale feature




Re-Scoring Method
→ Our Main Contribution in 2014



$$S_{new} = (1+N_d)^2 (1+N_{fg}-N_d) \log_2 (1+N_{bg}) (w_1 S^*_{BOW} + w_2 S^*_{DPM})$$

where:

N_d : number of shared words of foreground inside bounding box (green lines)
N_{fg} : number of shared word of foreground (both blue and green lines)
N_{bg} : number of shared word of background (black lines)
w_1 : weight of BOW score
w_2 : weight of DPM score

However

- How to weight score of BOW and DPM?
- How to handle more highly deformable and rich colored texture objects?

⇒ This year, we tried two methods.

Query Adaptive Fusion

- Instead of using average approach (w1=w2), we proposed an adaptive way of fusion.
- A neural network is used to automatically estimate weights of combining the two scores of BOW and DPM.

Query Adaptive Fusion

- Input of the network are features derived from:
  - average ratio of object area to image area
  - average number of keypoints inside query mask
  - number of shared visual words between two query examples
- Output of the network is weight of BOW and DPM derived from last years dataset
- Adaptive fusion score (*NII_HITACHI_UIT_1*):

$$S_{new} = (1 + N_d)^2 (1 + N_{fg} - N_d) \log_2 (1 + N_{bg}) (w_1^* S_{BOW}^* + w_2^* S_{DPM}^*)$$

Combination with RCNN Based Object Detector

- DPM are good, but it:
  - does not take into account color information
  - has not enough training data and hard negatives
  - still bad at too much deformable object (with occlusion)
- RCNN based object detector are current SOA
  - uses color information to compute similarity score
  - trained on a lot of data
  - retrained on specific query object
  - still not good at finding bounding box

⇒ We combine these methods together

Final Score Based on Fast RCNN and DPM

- The final score of our proposed method is given as following (*NII_HITACHI_UIT_3*):

$$S_{new} = \left(1+N_d^{DPM}\right)^2 \left(1+N_{fg}^{DPM} - N_d^{DPM}\right) \log_2\left(1+N_{bg}^{DPM}\right)\left(w_1 S^*_{BOW} + w_2 S^*_{FRCNN}\right)$$

where,

- Bounding box is kept as last year (returned from DPM), 3 types of shared points are computed the same
- Normalized score of Fast RCNN are used to compute base score

Experiments

Run ID	Description	MAP
F_A_NII_Hitachi_UIT_1	Query adaptive fusion	40.11%
F_A_NII_Hitachi_UIT_2	Last year config with w1=w2=0.5	41.76%
F_A_NII_Hitachi_UIT_3	Late fusion of DPM and Fast RCNN	42.42%
F_A_NII_Hitachi_UIT_4	Last year config with w1=0.67, w2=0.33	41.53%

Results - Good

- We got max perf on 8/30 queries from our 4 submitted runs.
- *Object query (9145 → this jukebox wall unit)*

- *Object query (9146 → this change machine)*

Results - Good

- Consistently good for logo query (2014 & 2015)
- *(9137 → a Ford script logo)*

3.

[shot160_453-1850.894336]

4.

[shot135_95-177.919637]

Results - Bad

- Small objects *(9129 → this silver necklace)*

9.

[shot218_1765-0.125700]

10.

[shot49_171-0.124500]

Results - Bad

- Texture, illumination *(9139 → this shaggy dog (Genghis))*

1\.

[shot194_1104-0.211400]

2\.

[shot206_381-0.208600]

Results - Bad

- Color information is important *(9136 → this yellow VW beetle with roofrack)*

4. 

[shot135_1383-0.176500]

5. 

[shot128_2066-0.137300]

Results - Bad

- Context *(9155 → this dart board)*

7. 

[shot6_111-0.593400] 

8. 

[shot4_977-0.558200] 

Conclusions

- The first time we use a RCNN in our system and it improves pretty much compared to two baselines (41.76% $\rightarrow$ 42.42%)
  - take into account pretrained network.
  - take advantage of color information.
- We tried to improve the adaptive weighting and it works on previous datasets, but unsuccessful in this year (40.11% vs 41.76%)
- There still have unsolved problems:
  - Too small objects (with no texture).
  - Too flexible query instances: persons, animals.

Best Run NII_Hitachi_UIT_3 (42.42%)