Supplementary Material for
"Bayes Merging of Multiple Vocabularies for Scalable Image Retrieval"

1. Overview

This document includes supplementary material to "Bayes Merging of Multiple Vocabularies for Scalable Image Retrieval". Included are the fast versions of Bayes merging and some sample retrieval results.

2. Fast Implementation of Bayes Merging

To speed up Bayes merging, besides the offline computation of the cardinality ratio $\frac{Card(\mathcal{A}\cap\mathcal{B})}{Card(\mathcal{A}\cup\mathcal{B})}$, the online process is presented in Algorithm 2 and Algorithm 1 (For one query feature).

These two algorithms correspond to two alternatives of inverted index. In Algorithm 1, the inverted list stores one entry per descriptor, which is required by HE [1, 2]. On the other hand, in Algorithm 2, the inverted index stores one entry per image [4], where the image identifier and the TF value are stored. For "Bayes merging + HE" method, the implementation is essentially built on Algorithm 1; for "Bayes merging" alone, Algorithm 2 is employed.

Here, we illustrate the case of Bayes merging of two vocabularies, because the pseudo-code does not look too long, and because the performance of merging two vocabularies is very close to multiple ones. Note that Bayes merging of multiple ($K \geq 3$) vocabularies shares essentially the same procedure.

Consequently, given two sets of indexed features $\mathcal{A}$ and $\mathcal{B}$, the Bayes merging method has the same computation complexity $\mathcal{O}\left(card(\mathcal{A}) + card(\mathcal{B})\right)$ with baseline B_1. In other words, for each query feature, we only have to traverse the lists of indexed features once.

3. Sample Retrieval Results

In this supplementary material, we also provide some sample retrieval results on the Holidays [1], Oxford [4], and Ukbench datasets [3]. These results are obtained using Bayes merging of two vocabularies of size 20K. Note that HE is not employed here. See Fig. 1, Fig. 2, and Fig. 3, respectively.

Algorithm 1 Bayes merging for one entry per descriptor

Input:

two arrays of the image indices idx_1, idx_2;
Arrays lengths $len(idx_1) = len_1$, $len(idx_2) = len_2$;
Two indicators $i \Leftarrow 0, j \Leftarrow 0$;
Initial scores of the images s;
Bayes weight w;

Iteration:

while $i < len_1$ and $i < len_2$ **do**
  if $idx_1[i] < idx_2[j]$ **then**
    $s\left[idx_1[i]\right] \Leftarrow s[idx_1[i]] + 1$;
    $i \Leftarrow i + 1$;
  else
    if $idx_1[i] > idx_2[j]$ **then**
      $s\left[idx_2[j]\right] \Leftarrow s[idx_2[j]] + 1$;
      $j \Leftarrow j + 1$;
    else
      do
        $s\left[idx_1[i]\right] \Leftarrow s\left[idx_1[i]\right] + w$;
        $i \Leftarrow i + 1$;
      while $i < len_1$ and $idx_1[i] == idx_1[i-1]$
      do
        $s\left[idx_2[j]\right] \Leftarrow s\left[idx_2[j]\right] + w$;
        $j \Leftarrow j + 1$;
      while $j < len_2$ and $idx_2[j] == idx_2[j-1]$
    end if
  end if
end while
while $i < len_1$ **do**
  $s\left[idx_1[i]\right] \Leftarrow s\left[idx_1[i]\right] + w$;
end while
while $j < len_2$ **do**
  $s\left[idx_2[j]\right] \Leftarrow s\left[idx_2[j]\right] + w$;
end while

Output:

The updated score s.

Algorithm 2 Bayes merging for one entry per image

Input:

two arrays of the image indices idx_1, idx_2;
two arrays of the TF values tf_1, tf_2;
Arrays lengths $len(idx_1) = len_1$, $len(idx_2) = len_2$;
Two indicators $i \Leftarrow 0, j \Leftarrow 0$;
Initial scores of the images s;
Bayes weight w;

Iteration:

while $i < len_1$ and $i < len_2$ **do**
  if $idx_1[i] < idx_2[j]$ **then**
    $s\,[idx_1[i]] \Leftarrow s[idx_1[i]] + tf_1[i]$;
    $i \Leftarrow i + 1$;
  else
    if $idx_1[i] > idx_2[j]$ **then**
      $s\,[idx_2[j]] \Leftarrow s[idx_2[j]] + tf_2[j]$;
      $j \Leftarrow j + 1$;
    else
      $s\,[idx_1[i]] \Leftarrow s\,[idx_1[i]] + w \cdot tf_1[i]$;
      $s\,[idx_2[j]] \Leftarrow s\,[idx_2[j]] + w \cdot tf_2[j]$;
      $i \Leftarrow i + 1$;
      $j \Leftarrow j + 1$;
    end if
  end if
end while
while $i < len_1$ **do**
  $s\,[idx_1[i]] \Leftarrow s\,[idx_1[i]] + w \cdot tf_1[i]$;
end while
while $j < len_2$ **do**
  $s\,[idx_2[j]] \Leftarrow s\,[idx_2[j]] + w \cdot tf_2[j]$;
end while

Output:

The updated score s.

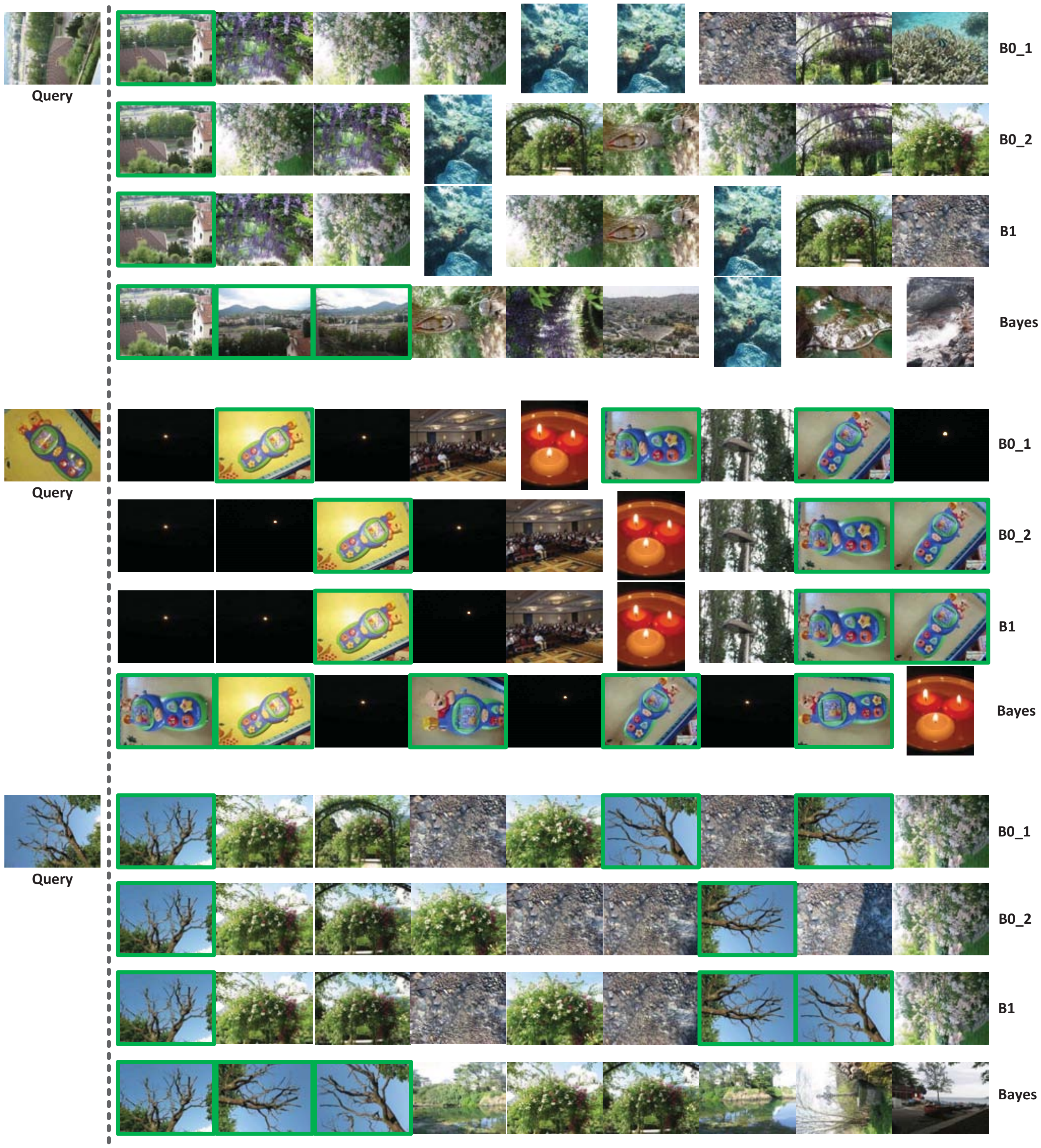

Figure 1. Sample retrieval results on Holidays dataset. For each query (**left**), retrieval results of four methods are presented in each row, *i.e.*, baseline B_0 using vocabulary 1 and 2 (**B0_1** and **B0_2**, respectively), baseline **B1**, and the proposed Bayes merging (**Bayes**). The images start from the second one in the rank list. The ground truth images are in green boxes.

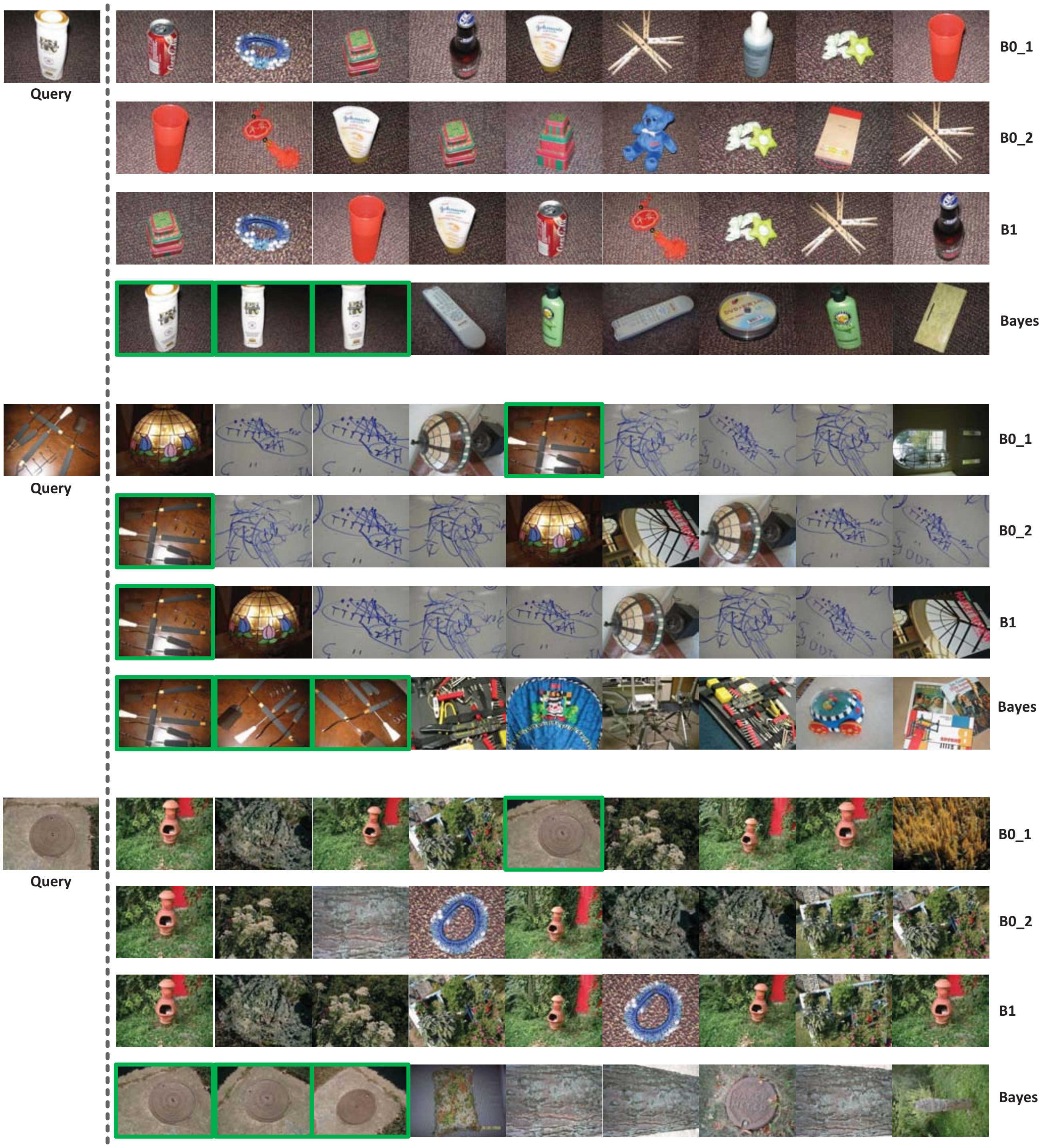

Figure 2. Sample retrieval results on Ukbench dataset. For each query (**left**), retrieval results of four methods are presented in each row, *i.e.*, baseline B_0 using vocabulary 1 and 2 (**B0_1** and **B0_2**, respectively), baseline **B1**, and the proposed Bayes merging (**Bayes**). The images start from the second one in the rank list. The ground truth images are in green boxes.

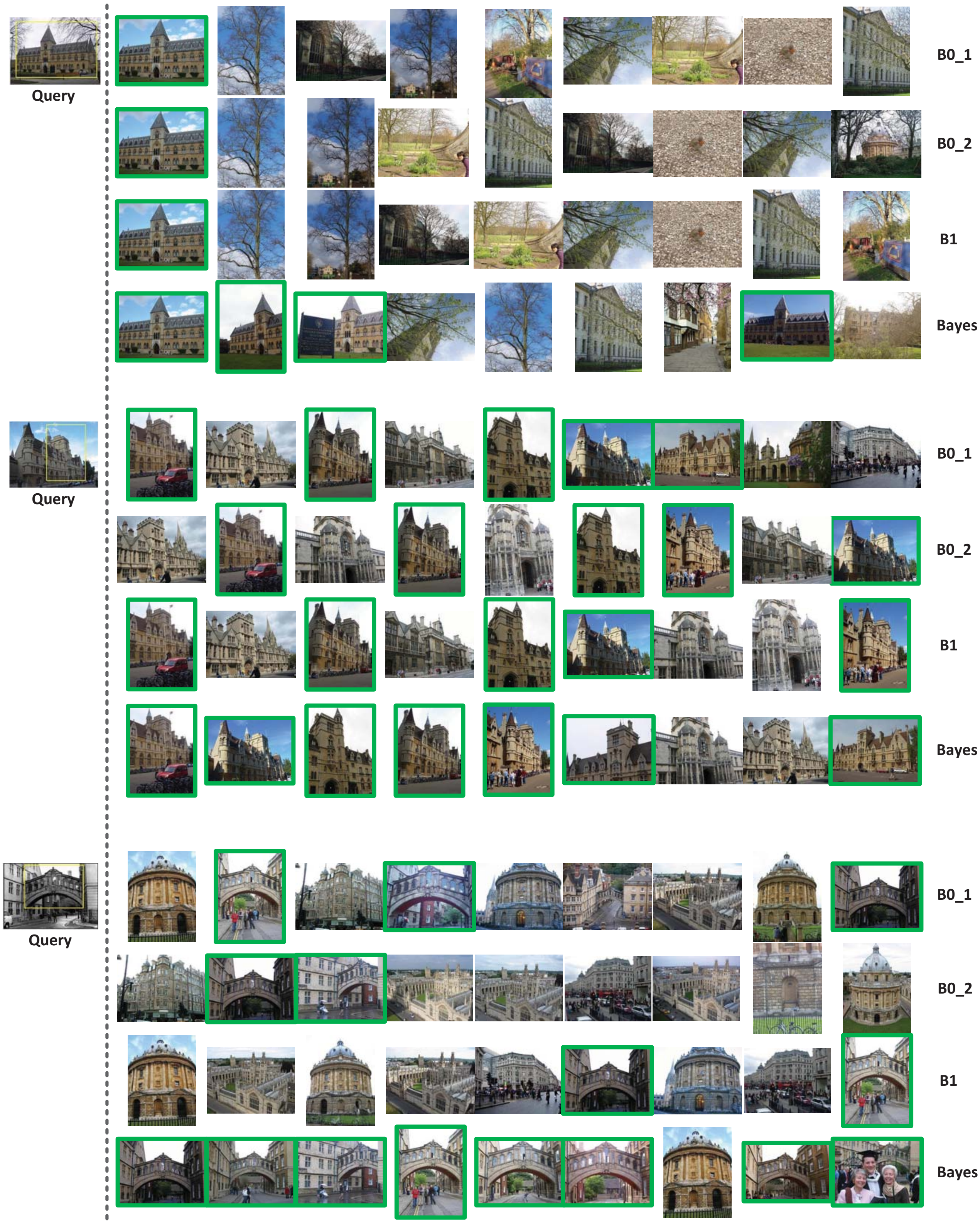

Figure 3. Sample retrieval results on Oxford dataset. For each query (**left**), retrieval results of four methods are presented in each row, *i.e.*, baseline B_0 using vocabulary 1 and 2 (**B0_1** and **B0_2**, respectively), baseline **B1**, and the proposed Bayes merging (**Bayes**). For the first and second queries, the images start from the second one in the rank list; for the third query, the images start from the 11th in the rank list. The ground truth images are in green boxes.

References

[1] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008.

[2] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *IJCV*, 2010.

[3] D. Niester and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[4] J. Philbin, O. Chum, M. Isard, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.