Mobile Product Image Search by Automatic Query Object Extraction

Xiaohui Shen[1], Zhe Lin[2], Jonathan Brandt[2], and Ying Wu[1]

[1] Northwestern University, Evanston, IL 60208, USA
{xsh835,yingwu}@eecs.northwestern.edu
[2] Advanced Technology Labs, Adobe, San Jose, CA 95110, USA
{zlin,jbrandt}@adobe.com

Abstract. Mobile product image search aims at identifying a product, or retrieving similar products from a database based on a photo captured from a mobile phone camera. Application of traditional image retrieval methods (e.g. bag-of-words) to mobile visual search has been shown to be effective in identifying duplicate/near-duplicate photos, near-planar and textured objects such as landmarks, books/cd covers. However, retrieving more general product categories is still a challenging research problem due to variations in viewpoint, illumination, scale, the existence of blur and background clutter in the query image, etc. In this paper, we propose a new approach that can simultaneously extract the product instance from the query, identify the instance, and retrieve visually similar product images. Based on the observation that good query segmentation helps improve retrieval accuracy and good search results provide good priors for segmentation, we formulate our approach in an iterative scheme to improve both query segmentation and retrieval accuracy. To this end, a weighted object mask voting algorithm is proposed based on a spatially-constrained model, which allows robust localization and segmentation of the query object, and achieves significantly better retrieval accuracy than previous methods. We show the effectiveness of our approach by applying it to a large, real-world product image dataset and a new object category dataset.

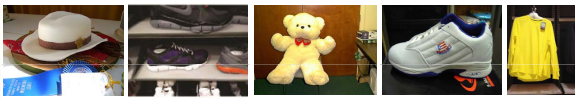
1 Introduction

Mobile product image search has recently become an interesting research topic due to the unprecedented development of smart phones and applications along with the increasing popularity of online shopping. In an ideal scenario, a user can simply take a picture of a product using a mobile phone to promptly identify the product and/or retrieve visually similar products from the database.

Traditional image retrieval methods typically adopt the bag-of-words model initially introduced in [1]. In this model, local features such as SIFT [2] are extracted from the query image and assigned to their closest visual words in a visual vocabulary. The query image is accordingly represented by a global histogram of visual words, and matched with database images by *tf-idf* weighting using inverted files[3,4].

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part IV, LNCS 7575, pp. 114–127, 2012.

(a) Examples of database images, with aligned products and clean background.



(b) Examples of query images taken by mobile phones, with different backgrounds, viewpoint and illumination.

Fig. 1. Examples of database images and query images in mobile product image search

The bag-of-words model works well for retrieving duplicate/near-duplicate images and near-planar/highly-textured objects. However, its performance is generally poor when directly applied to mobile product image search. The mobile product image search problem has the following distinct characteristics compared to general image search:

1. The products in the database images are mostly well aligned and captured in studio environments with controlled lighting. The background is often clean, and the texture details are clear. See Fig. 1(a) for an example.
2. Mobile query images are usually taken under very different lighting conditions with cluttered background. There may exist large viewpoint variations between the query and database images. Moreover, motion blur and out-of-focus blur are very common in images captured by mobile phones. Fig. 1(b) shows some examples of mobile query images.
3. Product instances are often non-planar (e.g. shoes) and/or less textured (e.g. clothes), hence the standard RANSAC-based verification can easily fail.
4. Some product instances (e.g. shoes) are visually very similar to each other, and only a small portion of visual features can discriminate them, so we need a fine-grained discrimination strategy for correct identification.

As a result, when we use the bag-of-words model to perform mobile product image search, the results may be largely affected by the features extracted from the background of the query images. Even when we specify the location of the product in the query image, the features around the occluding boundaries of the product may still be largely different from those extracted from clean background. One can segment the query object by manual labeling. However, simple labeling (e.g., specifying the object by a bounding rectangle, as in [5]) does not necessarily guarantee accurate segmentation results, while extensive and careful labeling largely increases the burden for the users.

To this end, in this paper, a new approach is proposed to simultaneously retrieve visually similar product images, and localize/identify the product instance in the query image. In our approach, each retrieved database image predicts a location and an outline shape (or mask) for the query object. The center location and the support region of the query object can then be inferred by a weighted object mask voting and aggregation scheme while removing the outliers. Based on that, the query object is automatically segmented and filled with clean background, which is used to refine the search results in the next round. Since better search results yield better query object extraction, and vice versa, the above two procedures are performed in an iterative and interleaved way, hence forming a closed-loop adaptation between query object extraction and object retrieval.

We collected two datasets for experimental validation: a large, real-world product image database for identical object retrieval, and a new object category dataset sampled from Caltech256 [6] for object category retrieval[1]. Experimental results on these two datasets show that our automatic query extraction yields even better results than manual segmentation with a bounding rectangle as initialization in retrieval accuracy, while our iterative approach significantly outperforms previous methods.

2 Related Work

Previous image retrieval research mostly focuses on duplicate image, or near-planar and textured object retrieval with applications to web image search and personal photo management. The standard bag-of-words model [1] is heavily explored for these tasks, and many of its variations are introduced to further improve the performance. They either encode spatial information[4,7,8,9,10], use better feature quantization[11,12,13,14], or better vocabularies [15] to refine the search results. Query expansion [16,17] is a common post-processing technique to increase the recall while improving the retrieval precision.

While general image search has been well-studied, research efforts devoted to mobile product image search are still limited. Some search engines for product images have recently been developed [18,19,20]. However, in these works, the query images are very similar to the database images (i.e., captured in the same settings). Google Goggles[2] and Amazon Flow[3] are well-known commercial mobile product image search engines, but are working robustly only for a few near-planar, textured object categories such as logos/trademarks, books/CD covers, landmarks, artworks, text, etc. In [21], a new database for mobile visual search is proposed, in which the objects are still limited to planar categories such as books and CD covers. Retrieving more general object categories (either severely non-planar, non-rigid, or less-textured objects) from mobile phones is still an open research question, and a search engine specifically designed for mobile product image search for more challenging object categories is highly demanded.

[1] We will make both of our datasets publicly available

[2] `www.google.com/mobile/goggles/`

[3] `http://flow.a9.com`

On the other hand, object segmentation [22] is integrated with other vision problems such as object detection, categorization and recognition. Reference images are used in [23,24] to perform co-segmentation. In [25,26], poselet and image contours are aggregated to segment the object. [27,28] propose simultaneous object detection and segmentation, while [29,30] introduce algorithms for concurrent object recognition and segmentation. However, they all assume the example images are of known category labels, and/or are not addressed in the context of image retrieval where the database consists of thousands to millions of unlabeled images. In the context of image retrieval, some approaches have been proposed to localize the object in the database images, either by sub-image search [31,7] or by generalized Hough voting [10]. However, to the best of our knowledge, there is no previous work to simultaneously localize, identify and automatically segment the object from the query image during large-scale search. In [17], the failure cases in query expansion are automatically recovered by removing background confusers from the top retrieval results, but the method assumes the confuser textures coexist in many database images, which is not valid in our case; also, no clear object boundary can be easily obtained using their method, which is critical for mobile product image search.

3 Formulation

In this section, we present our simultaneous query object extraction and retrieval algorithm. The query object location and its support map is estimated by aggregating votes from the top-retrieved database images. The estimated object support map is then used to generate a trimap for GrabCut [22], by which the query object is segmented.

3.1 Query Object Localization from Database Images

In [10], a spatially-constrained similarity measure is proposed to simultaneously retrieve and localize the objects in the database images, in which the object in the query image is manually specified by a bounding rectangle. In this paper, we propose that when the object location, scale and/or pose in the query image is unknown, the similarity measure can be further extended to localize the query object with the help of the top-retrieved database images. Robust query object localization serves as a good prior for segmentation, and good object segmentation allows more accurate retrieval by using the spatially constrained model and reducing the influence of background clutter.

Our retrieval framework falls under the category of approaches using local feature, visual vocabulary and inverted file. We denote the query image by Q and a database image by D, respectively. Let $\{f_1, f_2, \cdots, f_m\}$ be the local features extracted from Q, and $\{g_1, g_2, \cdots, g_n\}$ be the local features extracted from D. In order to encode relative feature locations in the image similarity, we use the spatially-constrained similarity measure defined in [10]:

$$S(Q,D|\mathbf{T}) = \sum_{k=1}^{N} \sum_{\substack{(f_i,g_j) \\ f_i \in Q, g_j \in D \\ w(f_i)=w(g_j)=k \\ ||\mathbf{T}(L(f_i))-L(g_j)||<\varepsilon}} \frac{\mathrm{idf}^2(k)}{\mathrm{tf}_Q(k) \cdot \mathrm{tf}_D(k)} \quad (1)$$

where k denotes the k-th visual word in the vocabulary. $w(f_i) = w(g_j) = k$ means that f_i and g_j are both assigned to visual word k. $\mathrm{idf}(k)$ is the inverse document frequency of k, $\mathrm{tf}_Q(k)$ and $\mathrm{tf}_D(k)$ are the term frequencies (i.e., number of occurrence) of k in Q and D respectively. $L(f) = (x_f, y_f)$ is the 2D image location of f. The spatial constraint $||\mathbf{T}(L(f_i)) - L(g_j)|| < \varepsilon$ means that the locations of two matched features should be sufficiently close under a certain transformation[4]. Therefore, all the matched feature pairs that violate that transformation would be filtered out.

The approximate optimal transformation $\mathbf{T}^*$ (maximizing the score in Eqn. 1) between Q and D is obtained by generalized Hough voting[10], while the database images are simultaneously ranked by the maximum scores in Eqn. 1.

Similar to [10], we use the generalized Hough voting algorithm to localize the object in the query. The spatial constraint $||\mathbf{T}(L(f_i)) - L(g_j)|| < \varepsilon$ is equivalent to $||(L(f_i)) - \mathbf{T}^{-1}(L(g_j))|| < \varepsilon$. To localize the object in the query image, we need to first find the optimal $\mathbf{T}^{*-1}$. We decompose $\mathbf{T}^{-1}$ to rotation angle, scale factor and translation. For simplicity of illustration, we ignore the rotation angle in the following description. The scale factor is uniformly quantized to 4 bins in the range of 1/2 and 2, and a voting map indicating the probability of the object support pixels is generated for each of the quantized scale factors.

Our object extraction process is illustrated in Fig. 2. Suppose that $w(f_i) = w(g_i)$ in Fig. 2(a) and (b). We assume that the product objects in the database images are mostly around the image center. Therefore, the image center is also considered as the object center c in D. Since $w(f_i) = w(g_i)$, given a certain scale factor s, if (f_i, g_i) obeys the transformation $\mathbf{T}^{-1}$, the object center in Q would be $L(f_i) + s \cdot \overrightarrow{L(g_i)c}$, where $\overrightarrow{L(g_i)c}$ denotes the vector from $L(g_i)$ to c in D. Therefore, we cast a vote for each matched feature pair on the corresponding center location in Q, with voting score $\frac{\mathrm{idf}^2(k)}{\mathrm{tf}_Q(k) \cdot \mathrm{tf}_D(k)}$. If all the (f_i, g_i) pairs obey the same transformation, the voted object center would be very consistent, see $(f_i, g_i)(i = 1, 2, 3)$ for examples. On the contrary, if a feature pair is not spatially consistent with others, it will vote for a different location ((f_4, g_4) and (f_5, g_5)). After voting from all matched feature pairs, we choose the location with the maximum score as the best estimated object center in Q. It is straightforward to verify that the maximum score at the estimated location is exactly the similarity measure defined in Eqn. 1 given the pre-quantized scale factor s. To choose the

[4] We only consider scale change and translation for simplicity of illustration, but rotation can be easily handled by max pooling on retrieval scores of multiple rotated versions of the query as in [10].

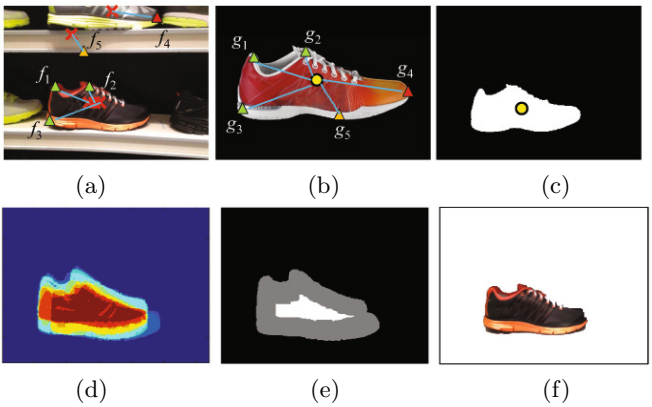


Fig. 2. Illustration of query object localization and extraction. (a) query image Q, (b) database image D, (c) voted mask of D on the object support map of Q, (d) query object support map by aggregating the voted masks of the top retrieved database images, (e) generated trimap based on the support map, (f) segmentation result using GrabCut with the trimap in (e).

best scale factor, we only need to select the scale corresponding to the voting map which generates the largest maximum score.

Based on the above process, each D has a prediction of the object location in the query image, which can be characterized by a vector $[x_c, y_c, s]^T$, where, (x_c, y_c) is the location of the object center in the query, and s is the relative scale factor between the query object compared with the object in D.

3.2 Query Object Extraction and Retrieval

In product image search, the database images mostly have clean background. The background color can be easily identified by finding the peak of the color histogram built upon the entire image, and the mask of the object can be accordingly obtained by comparing with the background color. Once we have the mask of the object in D as well as the estimated object location $[x_c, y_c, s]^T$, a transformed object mask can be voted at the estimated query location (x_c, y_c) with scale factor s, see Fig. 2(c) for example.

However, not all the top retrieved images can correctly localize the query object, especially when irrelevant objects are retrieved. Therefore, the outliers need to be excluded. Although sophisticated outlier removal methods such as spatial verification using RANSAC can be adopted here, the computational cost of these methods is typically high, and RANSAC does not handle non-planar, non-rigid, and less textured objects very well. Therefore, we only use their location predictions $[x_c, y_c, s]^T$ to effectively remove the outliers.

Consider that top N retrieved images are used to localize the query object, we get N location predictions $[x_c^i, y_c^i, s^i]^T (i = 1 \cdots N)$. Let $[\bar{x}_c, \bar{y}_c, \bar{s}]^T$ be the median values of all the predictions. For each $[x_c^i, y_c^i, s^i]^T$, if the squared distance

(a) (b) (c) (d) (e) (f) (g)

Fig. 3. Our query object localization method is robust to retrieved irrelevant objects. (a) Query images, (b)-(f) top 5 retrieved images, (g) voted object support maps.

$$D = (x_c^i - \bar{x}_c)^2 + (y_c^i - \bar{y}_c)^2 + \lambda(s^i - \bar{s})^2 > \tau \tag{2}$$

the corresponding database images will be removed from localization. In Eqn. 2 τ is a predefined threshold, and λ is a weight parameter.

We iterate this outlier removal and vote aggregation process multiple times to refine the object location, in which the median values $[\bar{x}_c, \bar{y}_c, \bar{s}]^T$ are updated after removing some outliers in each iteration. Once the outliers are removed, each inlier database image accumulates a mask at the estimated location with a weight. The weight can be determined as square root of the inverse of the rank, to assign more confidence on votes from higher ranked images. This process generates a soft map indicating the query object support region (Fig. 2(d)).

This algorithm is very simple, but can very effectively localize the object in the query image. See Fig. 3 for an example, even when irrelevant objects are retrieved, the location map can still accurately localize the object.

Once the object support map is generated, we use it to generate a trimap for GrabCut [22]. We first normalize the support map to a gray-scale image and perform dilation on the map. The pixels below a threshold (< 50) are set as background. Erosion is also performed, and the pixels above a high threshold (> 200) are set as foreground. All the other regions are labeled as uncertain. See Fig.2(e) for an example, the black regions represent the background, and the white and gray regions indicate the foreground and uncertain areas, respectively. In more challenging retrieval tasks (e.g., retrieving objects of the same semantic category but with large appearance changes, see Fig. 3), since shape information is not obvious in the estimated support map, to avoid false foreground labeling, only background and uncertain regions are labeled. Such a trimap is used as an input for GrabCut, and the final segmentation result is obtained as shown in Fig.2(f). Experimental results show that the overall segmentation results using our trimap are better than GrabCut with manual initialization.

We then extract the query object, fill the query image with a clean background and re-extract features from the new query image, in order to obtain better feature consistency across object boundaries, which are then used to perform search using Eqn. 1 in the next round. By reducing the background influence, the retrieval performance is dramatically improved. Therefore we can further use the refined search results to update the query object localization and segmentation.

(a) Examples of database images.

(a) Examples of query images.

Fig. 4. Example images in the sports product image dataset

By performing query object extraction and object search in an iterative way, the results of localization, segmentation and retrieval are simultaneously boosted. We stop the iteration when the difference between the segmentation masks of two consecutive iterations is smaller than a certain threshold. We found that in many cases, the differences of the segmented masks at the first two iteration steps are already small enough to stop the search. And most of the segmented results remain stable beyond the third iteration.

4 Experiments

We evaluated our method on two product image datasets, and compared it with the baseline bag-of-words retrieval method, the state-of-the-art spatial model as well as the query extraction method by GrabCut with manual initialization in terms of both segmentation and retrieval accuracy.

4.1 Datasets

We collected two datasets for product image search. The first one is a real-world sports product image (SPI) dataset, with 10 categories (hats, shirts, trousers, shoes, socks, gloves, balls, bags, neckerchief and bands) and 43953 catalog images. The objects in the database images are all well aligned, with clean background. See Fig. 4 for some examples. We also collected 67 query images captured with a mobile phone in local stores under various backgrounds, illumination and viewpoints. The objects in the query images are all shoes, and each has one exact same instance in the database, while there are totally 5925 catalog images in the shoe category. The task hence is to retrieve the same product from the database

(a) Examples of database images.

(a) Examples of query images.

Fig. 5. Example images in the object category search dataset

images. Cumulative Match Characteristic Curve (CMC) is used for performance evaluation, since it is equivalent to a 1:1 identification problem.

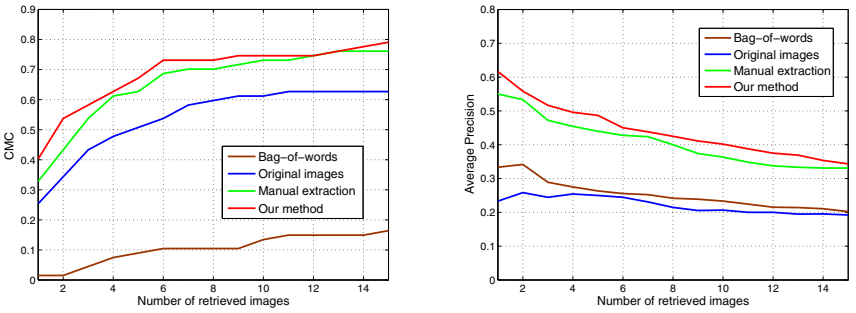
The second dataset is an object category search (OCS) dataset. Given a single query object, objects with the same semantic category need to be retrieved from the database. We collected 868 product images from Caltech 256 [6], in which the objects are positioned at the image center, with clean background. We also collected 60 query images for 6 categories from internet (each category has 10 queries). The query images contain background clutter, and the objects have large appearance differences, which makes it a very challenging task for object retrieval. See Fig. 5 for some examples. The number of relevant database images for the 6 categories ranges from 18 to 53. Average precision at rank k , i.e., the percentage of relevant images in the top-k retrieved images, is used to evaluate the performance on this dataset.

4.2 Results on the Sports Product Image Dataset

We use combined sparse and dense SIFT descriptors [2] as features[5] and hierarchical k-means clustering [3] to build the vocabulary. SIFT descriptors are computed with the "gravity" constraint [32]. The vocabulary on this dataset has 10580 visual words, which is used throughout all the experimental evaluations. Top 10 retrieved database images are used for query object localization.

We compared our method with the baseline bag-of-words method, and the spatially-constrained model with original query images [10]. We also manually segment the query object using GrabCut with a bounding rectangle as initialization, and then use the extracted object to perform search. Fig. 6(a) shows

[5] Sparse SIFT features are computed from DoG interest regions, and dense SIFT features are computed from a densely sampled regions in multiple scales across the image frame. Dense features are very useful for handling non-textured products.



(a) (b)

Fig. 6. Performance evaluation on the two mobile product image dataset. "Original Images" refers to the method in [10] using the original query image as a whole. (a)CMC curve on the sports product image dataset. Our method obtained significantly better performance than other methods. (b)Average precision at rank k on the object category search dataset. Our method consistently yields better precision.

the CMC for all the methods, in which the x-axis indicates the number of retrieved images k, and the y-axis indicates the probabilities that the correct catalog object appears in the top k retrieved images. It shows that the standard bag-of-words model cannot retrieve the correct object well for mobile product images. The spatially-constrained model removes some falsely matched features by more precise spatial matching, therefore largely improves the performance. However, it is still severely affected by the features extracted from the background and the object/background boundaries. Our method, by automatically extracting the query object, further improves the performance, and even outperforms the retrieve approach with manually initialized query object segmentation. In our method, 40% of the query images rank the correct catalog object at top 1, while the percentages for manual extraction and using original images are 32.8% and 25.3% respectively. When we consider the top 6 retrieved images, 73% of the query images have their correct catalog object ranked in top 6 with our method. The CMC curve only shows the results for top 15, as images with low ranks are far less important in most applications. There are still 20% of queries that cannot retrieve their relevant images in the top 15. This is because the viewpoint, lighting condition and image resolution are too different between the query and the database images. Further study can be conducted for these cases, e.g. investigating viewpoint or illumination robust features for product images.

Fig. 7 shows some examples of our query object extraction. We can see our object support maps accurately indicate the object regions, even when there are irrelevant objects in the top retrieved list (see the second row for an example). As a result, we can accurately extract the query object, and in many cases achieve more accurate performance than manually initialized segmentation.

Fig. 7. Examples of query object extraction on the sports product image dataset. (a) original query images, (b) object support maps, indicating the object regions, (c) automatic object cut using the support maps in (b), (d) GrabCut with manual initialization. Since the trimaps provided by our method are more accurate, the segmentation results are even better than manual extraction.

4.3 Results on the Object Category Search Dataset

The implementation on this dataset is the same as the first dataset. Fig. 6(b) shows the average precision at rank k, i.e., the average percentage of relevant objects appearing in the top-k retrieved images, for all the four methods.

In this dataset, the appearance variation is very large within one category. As a result, the spatially-constrained model, which is mainly targeted for instance retrieval instead of object category retrieval, is not sufficient. We can see that the performance of this spatial model is slightly worse than the bag-of-words model. The average precision at rank k for these two methods remains 20% to 30%, which indicates that the retrieval task for this dataset is quite difficult.

By using our simultaneous segmentation and retrieval method, the average precision is dramatically improved, as shown in Fig. 6(b). Similar to the sports product image search dataset, our method still produces better retrieval performance than manual query object extraction, which demonstrates the effectiveness of our method on this challenging task.

Some examples of query object extraction are provided in Fig. 8. We can see that, when the object appearance does not change significantly within the semantic category, our object support map can accurately estimate the query object regions (the top two rows). Meanwhile, when the object appearance variation is large and the initial search results are noisy, our filtering process using Eqn. 2 can remove some irrelevant objects that incorrectly localize the query

Fig. 8. Examples of query object extraction on the object category search dataset. (a) original images, (b) object support maps, indicating the object regions, (c) automatic cut using the support maps in (b), (d) GrabCut with manual initialization. Even when the appearance variation is very large where many irrelevant objects are retrieved, our method can still successfully localize and extract the query object.

object, and the query object location can still be accurately estimated (the bottom two rows). As a result, we can still get comparable segmentation results as the manually initialized extraction method.

4.4 Complexity

Compared with the bag-of-words model, the additional storage in the indexing file is the location for each feature, which can be encoded by a 1-byte integer as in [10]. Therefore the additional storage for a database with 45k images is less than 2 MB. The additional memory cost in the retrieval process is the voting maps for each database image when optimizing $\mathbf{T}^{-1}$, which has the size of 16×16 with floating values. When we use 4 scale bins, i.e., generating 4 voting maps for each database image, the additional memory cost for the 45k-image dataset is much less than the size of the inverted file. Since we need to perform one or two iterations of search, the retrieval time would be multiple times of the initial search time, but the absolute retrieval time is still very short. The most time-consuming step of our method is the GrabCut segmentation. Excluding Grabcut, with 3.4G CPU, the search procedure for one iteration step takes 0.380s on average on the

sports product image database with 45k images, and the whole process for each query can be performed within 3 seconds without code optimization.

5 Conclusions

We proposed a simple yet effective method to automatically extract the query object for mobile product image search. The top-retrieved images are used to localize the object in the query image with a spatially-constrained model. By extracting the query object, the influence of background clutter on visual features and retrieval accuracy is removed, and the retrieval performance is significantly improved. Experiments show that our method achieves more than 200% improvement over the baseline bag-of-words model, and even outperforms the method with manually initialized query object extraction.

Besides background clutter and small intra-class difference, there are still other issues in mobile product image search, such as the existence of image blur, and large viewpoint variation, image resolution and lighting condition. To improve the performance and make the product image search system practical, more research will be conducted to address these issues in our future work.

Acknowledgements. This work is partially supported by Adobe Systems Incorporated, and in part by National Science Foundation grant IIS- 0347877, IIS-0916607, US Army Research Laboratory and the US Army Research Office under grant ARO W911NF- 08-1-0504, and DARPA Award FA 8650-11-1-7149.

References

1. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV (2003)
2. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
3. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: CVPR (2006)
4. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)
5. He, J., Lin, T.H., Feng, J., Chang, S.F.: Mobile product search with bag of hash bits. In: ACM MM (2011)
6. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology (2007)
7. Lin, Z., Brandt, J.: A Local Bag-of-Features Model for Large-Scale Object Retrieval. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 294–308. Springer, Heidelberg (2010)
8. Zhang, Y., Jia, Z., Chen, T.: Image retrieval with geometry-preserving visual phrases. In: CVPR (2011)
9. Cao, Y., Wang, C., Li, Z., Zhang, L., Zhang, L.: Spatial-bag-of-features. In: CVPR (2010)
10. Shen, X., Lin, Z., Brandt, J., Avidan, S., Wu, Y.: Object retrieval and localization with spatially-constrained similarity measure and k-nn reranking. In: CVPR (2012)

11. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR (2008)
12. Wang, X., Yang, M., Cour, T., Zhu, S., Yu, K., Han, T.X.: Contextual weighting for vocabulary tree based image retrieval. In: ICCV (2011)
13. Jégou, H., Harzallah, H., Schmid, C.: A contextual dissimilarity measure for accurate and efficient image search. In: CVPR (2007)
14. Philbin, J., Isard, M., Sivic, J., Zisserman, A.: Descriptor Learning for Efficient Retrieval. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 677–691. Springer, Heidelberg (2010)
15. Mikulík, A., Perdoch, M., Chum, O., Matas, J.: Learning a Fine Vocabulary. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 1–14. Springer, Heidelberg (2010)
16. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: ICCV (2007)
17. Chum, O., Mikulík, A., Perd'och, M., Matas, J.: Total recall II: Query expansion revisited. In: CVPR (2011)
18. Jing, Y., Baluja, S.: Pagerank for product image search. In: WWW (2008)
19. Lin, X., Gokturk, B., Sumengen, B., Vu, D.: Visual search engine for product images. In: Multimedia Content Access: Algorithms and Systems II (2008)
20. Girod, B., Chandrasekhar, V., Chen, D., Cheung, N.M., Grzeszczuk, R., Reznik, Y., Takacs, G., Tsai, S., Vedantham, R.: Mobile visual search. IEEE Signal Processing Magazine 28 (2011)
21. Chandrasekhar, V., Chen, D., Tsai, S., Cheung, N.M., Chen, H., Takacs, G., Reznik, Y., Vedantham, R., Grzeszczuk, R., Bach, J., Girod, B.: The stanford mobile visual search dataset. In: ACM Multimedia Systems Conference (2011)
22. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In: SIGGRAPH (2004)
23. Batra, D., Kowdle, A., Parikh, D., Luo, J., Chen, T.: icoseg: Interactive co-segmentation with intelligent scribble guidance. In: CVPR (2010)
24. Rother, C., Kolmogorov, V., Minka, T., Blake, A.: Cosegmentation of image pairs by histogram matching-incorporating a global constraint into MRFs. In: CVPR (2006)
25. Bourdev, L.D., Malik, J.: Poselets: Body part detectors trained using 3D human pose annotations. In: ICCV (2009)
26. Brox, T., Bourdev, L.D., Maji, S., Malik, J.: Object segmentation by alignment of poselet activations to image contours. In: CVPR (2011)
27. Wu, B., Nevatia, R.: Simultaneous object detection and segmentation by boosting local shape feature based classifier. In: CVPR (2007)
28. Opelt, A., Pinz, A., Zisserman, A.: A Boundary-Fragment-Model for Object Detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 575–588. Springer, Heidelberg (2006)
29. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In: ECCV Workshop on Statistical Learning in Computer Vision (2004)
30. Yeh, T., Lee, J.J., Darrell, T.: Fast concurrent object localization and recognition. In: CVPR (2009)
31. Lampert, C.H.: Detecting objecs in large image collections and videos by efficient subimage retrieval. In: ICCV (2009)
32. Perd'och, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. In: CVPR (2009)