Handling Significant Scale Difference for Object Retrieval in a Supermarket

Yuhang Zhang*, Lei Wang*, Richard Hartley*† and Hongdong Li*†
*The Australian National University
†NICTA
Email: {yuhang.zhang, lei.wang, richard.hartley, hongdong.li}@anu.edu.au

***Abstract*—We propose an object retrieval application which can retrieve user specified objects from a big supermarket. Significant and unpredictable scale difference between the query and the database image is the major obstacle encountered. The widely used local invariant features show their deficiency in such an occasion. To improve the situation, we first design a new weighting scheme which can assess the repeatability of local features against scale variance. Also, another method which deals with scale difference through retrieving a query under multiple scales is also developed. Our methods have been tested on a real image database collected from a local supermarket and outperform the existing local invariant feature based image retrieval approaches. A new spatial check method is also briefly discussed.**

***Keywords*-image retrieval; local feature; scale invariance.**

I. Introduction

Recent years have seen quite some successful content based image retrieval systems which can find objects from video or image set [1], [2], [3]. One salient general character of these applications is to describe images with local invariant features. Local invariant features hold many appealing advantages. One of them is the scale invariance which has demonstrated excellent performance in the existing image retrieval systems. In this work we conduct an object retrieval in a supermarket. Differently, we show the deficiency of local invariant features in handling significant scale difference in image retrieval, and introduce our multi-scale methods to improve the situation. As shown in Figure 1, our query image contains the query object generally under a very large scale. In contrast, the corresponding database image contains the queried object under a much smaller scale together with many clutters. Moreover, since the query is to be submitted by a user, the retrieval system cannot know the exact scale of queries in advance and thus cannot down sample each query image with a predefined ratio.

We choose supermarket as our object retrieval context since it provides a comprehensive test for the capability of an object recognition system. Nowadays a typical supermarket carries around $50,000$ different products [4]. This diversity provide a critical challenge to the discriminative capability of an object recognition system. Moreover, in a supermarket each individual object presents itself among strong clutters. The illumination on different objects varies as they are located on different level of shelf. A big variety also lies in the viewing directions towards different objects. Occlusion and scale difference is also possible.

We thank the Coles store located in Woden Plaza, ACT. Their understanding and support make this work possible and are highly appreciated.

We analyze the deficiency of local invariant features and review the general framework of visual word based image retrieval in the section of literature review. A brief introduction to our previous work will also be conducted. We then introduce our new proposed methods and compare its performance with existing methods in experiments. The proposed methods tackle significant scale difference with two tactics. The first is through adaptively assigning the weight of local features (visual words) according to their repeatability over different scales. The second is through describing a query under different scales and retrieving each version of the query independently. The final retrieval result is then constructed by combining the retrieval results obtained under multiple scales. A new spatial check mechanism will also be briefly introduced before the conclusion.

II. Literature Review

To handle scale variance, Harris-Laplace detector [5], [6] works in two steps: firstly, a scale-space representation of the Harris function is created and the initial interest points are detected by selecting the local maxima in each 3×3 co-scale neighborhood under each scale; secondly, the extrema of the LoG (Laplacian of Gaussian) are searched for each initial interest point in the scale space within a variance range between $[0.7, 1.4]$, namely the identification of the characteristic scale. The location of the extrema of the LoG, or the characteristic scale of the initial interest point is determined by the image pattern but not the scale in which the image pattern is presented. Thus identical features can be extracted from an image even though the image is presented in different scales. This scale invariance mechanism is inherited by Harris-Affine detector and Hessian-Affine detector [5]. However, as mentioned in their work the repeatability of Harris-Laplace feature is no more than 70% even after a scale change of merely 1.4. That is because many points as well as information in the image disappears after a reduction in resolution. For SIFT (Scale Invariant Feature Transform) feature [7], the image is firstly

Figure 1. Query and database image example: the top is the query image containing the query object in a very large scale; the bottom is one of the database image containing the queried object in a much smaller scale together with much clutters.

Figure 2. Local feature extraction under different scales: From top to bottom are the original image, 2:1 down sampled image and 4:1 down sampled image. Hessian-Affine features marked as yellow ellipses are extracted from each of the tree images individually. As the scale of the image goes down, the number of extracted features declines sharply.

down-sampled into sequential octaves. Then within each octave the image is convolved with a Gaussian kernel to build the scale space representation, from which the local extrema of the DoG (Difference of Gaussian) is detected. By resampling images at each octave, features are extracted under all possible resolutions and a large range of scales can be covered. However, SIFT still bears the problem of dropping off features as the scale of the original image declines. Figure 2 gives an intuitive illustration of above discussion. The original image (top) is down-sampled at the rate of $2:1$ (middle) and $4:1$ (bottom) respectively. Harris-Affine features are then detected under each scale. Let us now assume that the top image is the query and the images at the middle and the bottom are to be retrieved from the database. Obviously, the number of features extracted from the query is much larger than that extracted from the two relevant images in the database. Does it hurt? The extra features from the query will not only find no true match but also generate a considerable number of false matches with irrelevant images (In the framework of visual word based image retrieval, through clustering a feature is always matched to some others). Hence an irrelevant image may accumulate a higher score than a relevant image does. In our experiments this problem forms the bottleneck of retrieval performance.

The general framework of visual word based image retrieval [2] is as follows. First, local invariant features are extracted from each image. The extracted features are then clustered to generate visual words [8]. Features in the same cluster are treated as matched and assigned the same label, namely the same visual word ID. Each image is then described by the visual words that it contains, in the form of a high-dimensional vector. Each dimension of the vector corresponds to one type of visual word. To achieve high retrieval accuracy, each dimension of the vector can be assigned a *tf-idf* (term frequency–inverse document frequency) weight [2] as shown in Equation 1. w_{ij} is the weight of visual word i in image j. n_j is the number of visual words in image j. n_{ij} is the number of occurrences of visual word i in image j. N_i is the number of images in the database that contain visual word i. N is the number of images in the database. The more a visual word appears in the database, the less important it is in describing an image; the more it appears in an individual image, the more important it is in describing this image. During retrieval, the similarity between two images is computed as the L_p-norm distance between their corresponding vectors. A larger distance indicates a smaller similarity. As shown in Equation 2, d_{mn} is the L_p-norm distance between image m and n. $\mathbf{v}_m$ and $\mathbf{v}_n$ are the two vectors corresponding to the two images. To achieve high retrieval speed, invert file is used to index the images in the database. That is, for each visual word the invert file records all the images that contain this visual word. Given a query, only images in the database sharing the same visual words with the query are assessed. Our work generally follows above framework but develops new mechanisms to handle large scale difference which will be discussed in the next section.

$$w_{ij} = \frac{n_{ij}}{n_j} \log \frac{N}{N_i} \tag{1}$$

$$d_{mn} = \left\| \frac{\mathbf{v}_m}{\|\mathbf{v}_m\|_p} - \frac{\mathbf{v}_n}{\|\mathbf{v}_n\|_p} \right\|_p \quad (2)$$

The work in this paper is an extension of our previous work [9]. In [9] we showed that the L_p-norm distance does not perform well when retrieving objects from strong background clutters. Instead, an inner-product in Equation 3 was proposed in [9] to measure the similarity between the query and each database image. Note that the similarity between two images is proportional to the inner-product between their corresponding vectors. That is to say, instead of measuring the difference between the two images, we assess the similarity between them directly. Moreover, Equation 1 has been modified to Equation 4 in [9]. According to Equation 1 the weight of each visual words is affected by not only the number of itself but also the number of other visual words in the same image. In such a manner the existence of irrelevant clutters will affect the matching score between identical visual words. Besides, in a supermarket multiple copies of one product are often placed together on a shelf. According to Equation 1, the existence of multiple copies will also change the weight of visual words. In this way, a relevant image containing a single true match in strong clutters may be beaten by an irrelevant image containing multiple copies of false match in weak clutters. Therefore, Equation 1 was replaced by Equation 4, in which the weight of a visual word is only determined by its occurrence frequency in the database.

$$d'_{mn} = \langle \mathbf{v}_m, \mathbf{v}_n \rangle \quad (3)$$

$$w'_{ij} = \begin{cases} \log \frac{N}{N_i}, & \text{if } n_{ij} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

In [9], to avoid the problem of large scale difference each query was manually down sampled to a similar scale to the true matches in the database. This drawback has to be removed in order to handle practical retrieval tasks, so in this work we avoid the manually down sampling and propose new methods to handle the large scale difference.

III. Multi-scale Image Retrieval with Local Invariant Features

As stated above, scale difference leads to the imbalance in feature numbers between the query and its true matches in the database. The extra features from the larger scale generate mismatches and cause retrieval to fail. We handle this deficiency in the following two ways.

A. Adaptively Assigning Weight before Retrieval

The first way is through adaptively assigning weights to different visual words. Weighting visual words is not a new mechanism in the literature, however, previous works do not pay sufficient attention to the repeatability of visual words over scale variance [2], [3]. Based on our previous discussion, a proper weighting scheme should depress the weight of extra words produced by larger scale and increase the weight of consistent words that can be found under a sequence of scales. To check the repeatability of a visual word with regard to a query object, we resize the query image into multiple scales and then describe each scale of the query with the same vocabulary. As we can imagine, some of the visual words can survive multiple scales whereas some cannot. We then assign weight to each visual word through Equation 5. $\widetilde{w}_{ij}$ is the weight of visual word i in query image j. q_{ijk} is the number of occurrences of visual word i in image j under scale k. q_{jk} is the total number of visual words in query image j under scale k. Through Equation 5, a visual word surviving more scales becomes more important for a query. Note that the usage of q_{jk} indicates that a visual word appearing under a scale where few visual words can be found has higher weight. This is reasonable because these visual words are more consistent against scale change.

$$\widetilde{w}_{ij} = \sum_k \frac{q_{ijk}}{q_{jk}} \log \frac{N}{N_i} \quad (5)$$

Note that we only resize the query image but not the database images, because resizing all images in the database and extract features under all possible scales not only demand enormous computation time but also require huge storage space. For the database images, we still weight each visual word according to Equation 4, which can effectively handle background clutters and multiple copies of the same object.

Besides assessing the repeatability of each visual word, Equation 5 also gives each query a more comprehensive description because it brings features extracted from multiple scales into one bag. Building an expanded description of the query image is not a new idea in the literature. Whereas the work in [10] expands the query description based on the images in the database, our expansion focuses on the scale aspect and does not require additional images. In latter sections we will refer this method as **A**.

B. Combine Matching Scores after Retrieval

In Section III-A we assign weights to visual words according to its repeatability over different scales. Parallel to that method, we can simply treat the query under different scales as independent queries. After performing retrieval with each of them individually, we combine the results from different scales to form a final result for the original query. Both methods in current section and the next section follow this idea.

Denote the matching score obtained by the query m under scale k with database image n as $\widetilde{d}_{mkn}$. Since the combination is implemented over multiple scales, we cannot compute the matching score $\widetilde{d}_{mkn}$ simply using Equation 3. That is

because a query under a larger scale usually has more visual words. However, through Equation 3, having more visual words tends to generate higher matching score. If so, the query under the largest scale will dominate the combination. We modify Equation 3 to Equation 6. $\mathbf{v}_{mk}$ represents the query image m under scale k. $\mathbf{v}_n$ represents the database image n. The denominator $\langle \mathbf{v}_{mk}, \mathbf{v}_{mk} \rangle$ computes the inner-product between the query image m under the scale k and itself. Through Equation 6, possessing more visual words no longer results in a higher matching score. Hence a query image under a larger scale does not hold a dominating position any more.

$$\widetilde{d}_{mkn} = \frac{\langle \mathbf{v}_{mk}, \mathbf{v}_n \rangle}{\langle \mathbf{v}_{mk}, \mathbf{v}_{mk} \rangle} \tag{6}$$

Define that $\widetilde{d}_{mkn}$ reaches its maximum at the scale k'. Obviously, $\widetilde{d}_{mk'n}$ suggests the highest matching score that the database image n can obtain with the query image m under all scales. If they are true match, k' indicates the scale under which the database image n can match most of the visual words possessed by the query image m. In other words, the feature imbalance between the two relevant images is approximately removed. If they are not true match, k' only indicates a scale under which the query looks most similar to an irrelevant database image n. Two relevant images under the same scale are more similar than two irrelevant images under their most similar scales are. Thus we construct the final retrieval result by sorting all database images according to its $\widetilde{d}_{mk'}$, namely the highest matching score between itself and the query under all scales. We label this method as **B1**.

In our problem, a relevant database image cannot always obtain a higher matching score than an irrelevant database image does under all scales. Hence, we pick the scale under which the database image can obtain the highest score in Method **B1** for comparison. Actually, it is even less likely for an irrelevant database image to always beat a relevant image under all scales. That is so say, if we accumulate the total matching score between a database image and the query image under all scales, relevant images should more likely obtain higher total scores than irrelevant images do. Hence, in Method **B2** we construct the final matching score for each database image by summing up the matching scores under all possible scales and then sort.

C. Combine Sorting Ranks after Retrieval

Besides matching scores, each database image also gets multiple sorting ranks when being retrieved against the query under multiple scales. Like combining matching scores, we can also combine these ranks to construct a final result. We cannot sort the database images according to their best rank, because there are more than one "first" which we cannot distinguish. We cannot simply sum up all ranks of each database image either, otherwise the final rank of a database image will be dominated by its lower ranks (note that lower ranks are represented as large numbers). Instead we only sum up the best sequential n ranks of each database image as shown in Equation 7, where R denotes rank. The value of n is to be determined experimentally. We label this method as **C**.

$$R_{\mathbf{final}} = \min_i (R_i + R_{i+1} + ... + R_{i+n-1}) \tag{7}$$

Figure 3. Similar but not identical objects: the same bottle, the same brand and the same cup on the labels but different coffee for customers. We treat them as two different objects during retrieval, although many identical local features can surely be extracted from them.

IV. Experimental Result

Our images are collected from a local Coles supermarket. Totally, eighteen 30-meter-long shelves and all the products on them have been captured by $3,153$ images as the database[1]. 300 additional images of the objects on the shelves have been taken and used as query images. As illustrated by Figure 1, each query image contains a single object under quite large a scale and each database image contains three or more levels of a market shelf including all the objects on the shelf. All the $3,153$ database images and 300 query images are used in our experiments. Each image is either $2,272 \times 1,704$ or $2,592 \times 1,944$ in size which is much larger than most known image database [11], [12]. This high resolution is necessary to hold sufficient information for each object appearing in the image. The ground truth is built by manually checking the query image against each database image. *Only* completely identical objects are treated as true matches. That is to say, retrieving NESCAFÉ BLEND 43 out when NESCAFÉ DECAF (see Figure 3) is queried is wrong!

From all the $3,153$ database images, we detected $51,537,169$ Hessian-Affine features in total. Some of the Hessian-Affine features are shown in Figure 4. These Hessian-Affine features are then described with SIFT descriptors [13], each of which is a 128 dimensional unit vector. We then use hierarchical k-means to cluster the $51,537,169$ unit vectors into 1,000,000 clusters (over 3 levels, the branch factor $k = 100$ on each level). Thus we obtain a visual vocabulary containing 1,000,000 visual words. Each image is then described with these visual words.

[1]this image set together with the queries and the ground truth has been made available on web: http://yuhang.rsise.anu.edu.au/

Figure 4. Some Hessian-Affine features detected from one of the database images: the top is the database image and the bottom are some of the Hessian-Affine feature detected.

Invert file is used to index the database images, which reduces the retrieval time to no more than 0.025 seconds per query on a Linux server equipped with Dual Core 2.8GHz CPU and 4G RAM. We also implement the retrieval on other machines and find the retrieval speed is mainly constrained by the size of RAM.

To show how the scale of query images impacts the retrieval performance, we resize each query image into 10 different scales with a scale step factor of 1.25 and implement retrieval under each scale. Figure 6 shows the retrieval performance with local invariant features given the queries under 10 different scales. The horizontal axis is the number of returned images. The vertical axis is the percentage of queries that have found at least one true match (ALOT) among the retrieved images. According to the descending order of retrieval performance, the ten scales are ranked as 6 5 7 4 3 8 2 1 9 0. Scale-6 gives the best performance because most true matches happen around this scale. However we cannot know this scale in advance, and scale-6 may not perform best if we use another set of queries or retrieve in another database. Giving the worst performance, scale-0 corresponds to the original scale of the query image. It performs worst because it suffers most from the problem of feature number imbalance. This will be the retrieval result if we solely rely on local features to handle scale difference as in [1], [2], [3]. Scale-7, scale-8 and scale-9 do not perform well either because the queries are down sized to too small scales, which cannot provide sufficient information about the query.

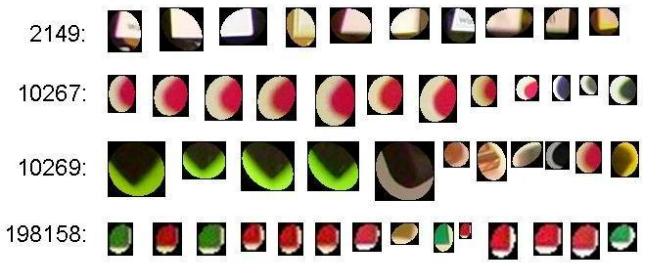

Figure 5. Visual word example: in this graph we show 4 different visual words. The number in front of each row is the word ID. The thumbnail images following the word ID are some of the Hessian-Affine features clustered to the same visual word. As expected, the Hessian-Affine features that belongs to the same visual word are quite similar. Note that their color can be different because the features are extracted from grey-value images. Besides, the visual word 10267 and 10269 are slightly similar to each other. That is also reasonable. Their word ID should be read as $1-02-67$ and $1-02-69$, which suggests that they were ever in the same cluster during hierarchical k-means. Only at the bottom level of the hierarchy were they separated.

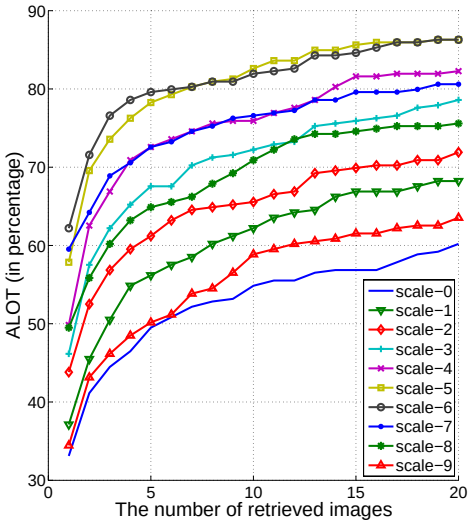

Figure 6. Retrieval with local invariant features under a single scale: From scale-0 to scale-9, the query images are down sampled to 1.25^0 to 1.25^{-9} times of their original scales. Although local invariant features are used, the retrieval performance varies dramatically as the scale of the query changes. ALOT stands for At Least One True match.

This paper proposes four methods to tackle the scale difference. Method **A** is to assign weight to visual words according to its repeatability over scale variance. Method **B1** is to sort all database images according to its highest matching score with the query under different scales. Method **B2** is to sort all database images according to its

total matching score. Method **C** is to sort all database images according to the sum of its best sequential n ranks. We empirically set $n = 4$ in method **C**. Figure 7 shows the performance of all the proposed methods when implemented with 10 different scales. The four proposed methods achieve comparable results and all of them outperform scale-0 in Figure 6 which solely relies on local invariant features to handle scale difference.

To achieve fast retrieval speed, it is preferable to retrieve on few scales only. Thus we test our methods on a smaller number of scales. Scale step is increased to 1.5 and the retrieval is implemented on 5 different scales. Note that the scale range is kept as it was ($1.25^{10} \approx 1.5^5$). Under such a configuration, $n = 3$ optimizes the performance of method **C**. The performance of all methods is presented in Figure 8. Compared with the result in Figure 7, the performance of Method **A**, **B1** and **B2** slightly declines. In contrast, the performance of Method **C** not only holds up but also rises 3 percents at the first returned image. Moreover, when retrieval is implemented under 5 different scales, Method **C** performs almost as good as scale-6, the best one in Figure 6. Over 60% of the queries find true matches in the first returned image. Over 75% of the queries find true matches in the top 5 returned images. The four proposed methods constantly achieve better results than scale-0 in Figure 6. Figure 9 and Figure 10 show some of the queries and the top 2 database images retrieved by method **C**. Figure 11 particularly demonstrates our system's capability in distinguishing similar objects.

V. Spatial Check on Visual Word

Spatial check has been a popular post-verification mechanism in the literature of local invariant feature based image retrieval [14], [2], [15]. By checking the spatial consistency between the p-nearest neighbors of the two matched local features, false matches are removed. However, the spatial check method used in the literature usually checks the position of affine features corresponding to each visual word after retrieval, which not only requires recording the on-image-position of each visual word but also consumes much computation during runtime. Here we shift the spatial check from feature level to visual word level. When building the invert file, following the entry of each visual word there are not only all the images containing this word but also the 15 spatial nearest visual words (precomputed) in each image. In this manner, we could implement the spatial check simultaneously as retrieving. That is, only those visual words which not only have identical word ID but also have at least two identical neighbors out of the 15 nearest ones are treated as matched. As shown in the top row of Figure 12, two images (left and right) containing common regions are captured. The common area between them is cropped out with yellow rectangle. The second row shows all the matched features based on visual word ID only. False

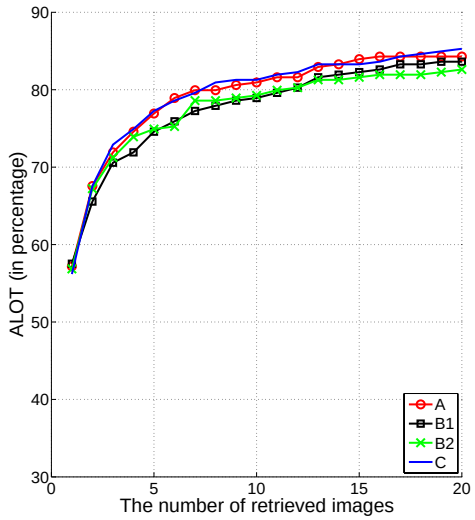

Figure 7. Retrieval under 10 different scales with the four proposed methods: **A**-assign the weight of visual words according to their repeatability over different scales; **B1**-retrieve the query in multiple scales independently and sort the database images according to their highest matching scores; **B1**-retrieve the query in multiple scales independently and sort the database images according to their total matching scores; **C**-retrieve the query in multiple scales independently and sort the database images according to their best sequential 4 ranks. The performance of the four proposed methods are comparable. ALOT stands for At Least One True.

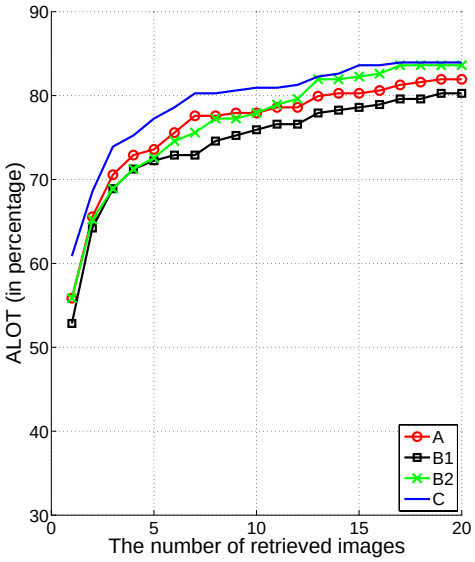

Figure 8. Retrieval under 5 different scales with the four proposed methods: whereas the other three methods suffer a decline in performance, Method **C** slightly goes up and becomes the obvious best. ALOT stands for At Least One True.

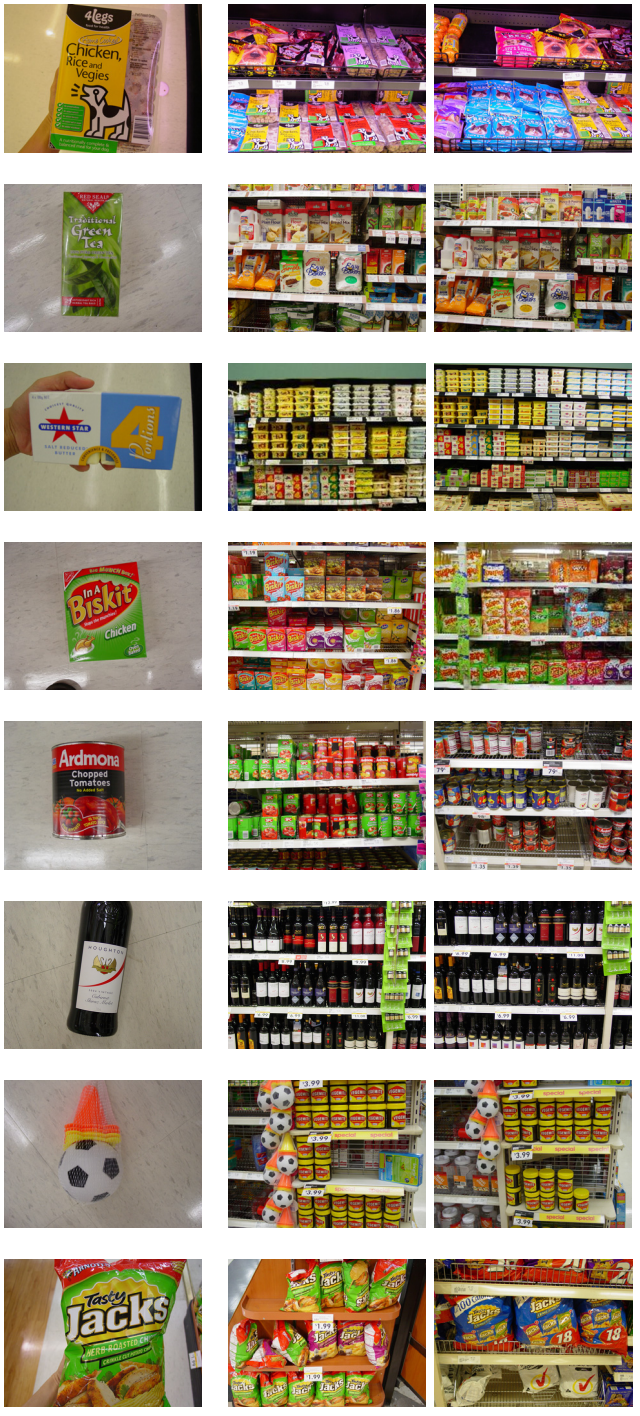

Figure 9. Retrieval Examples: the left column are the queries, the middle and the right columns are the first and the second returned images by Method **C**

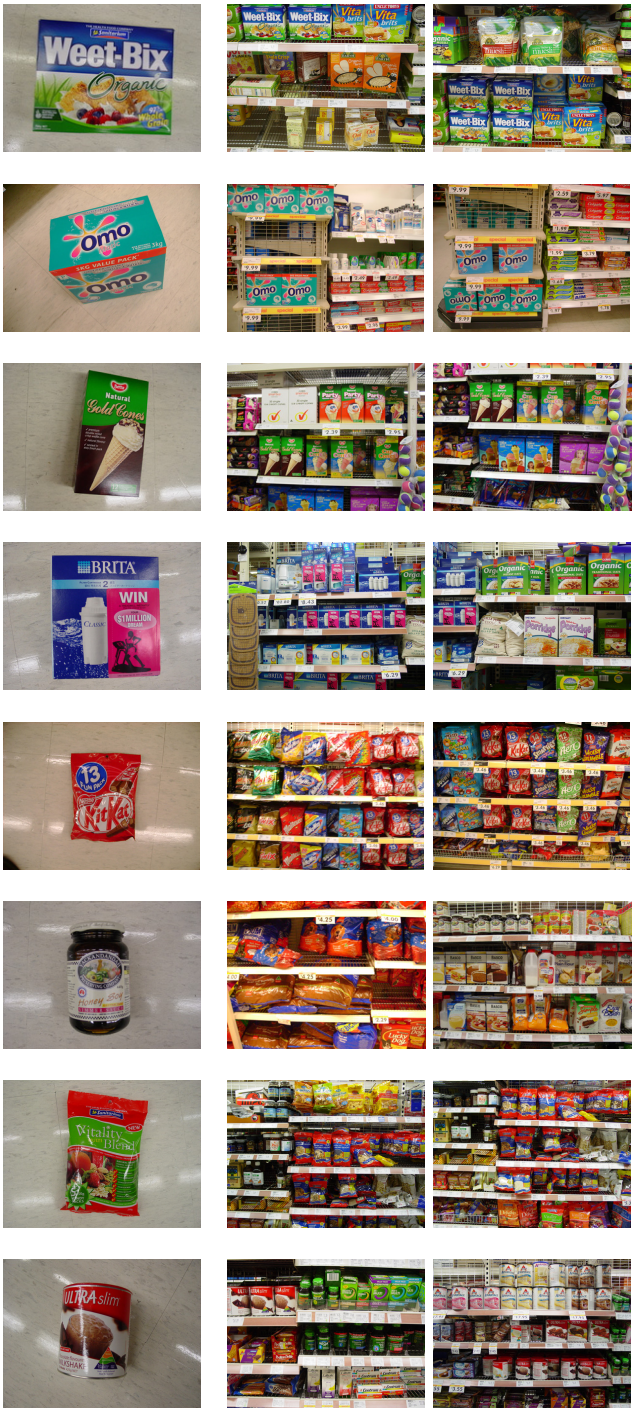

Figure 10. Retrieval Examples: the left column are the queries, the middle and the right columns are the first and the second returned images by Method **C**

Figure 11. Finding NESCAFÉ of the particular type: we enlarge the corresponding region in the database image to show that we have found exactly the same object.

Figure 12. Visual word based spatial check: first row - original images; second row - matches based on visual word ID; third row - matches approved by visual word based spatial check; fourth row - remaining false matches.

matches out of the common area are observed. The third row shows the matches approved by the visual word based spatial check. After removing those false matches, only 7 matches out of the common area are left. They are actually *true* matches. As the bottom row shows, these 7 matches are caused by different objects manufactured by the same corporation, thus sharing the same brand.

VI. Conclusion

We proposed an object retrieval application in the environment of a supermarket. Our methods successfully make up the deficiency of local invariant features in dealing with large scale difference and improve the retrieval performance. With the invert file, even the query is retrieved under 10 different scales, the retrieval time for each query does not exceed 0.25 seconds. The major increased computation time is caused by the feature extraction under multiple scales. Future work can be devoted to boosting the speed of feature extraction under multiple scales.

References

[1] J. Sivic, F. Schaffalitzky, and A. Zisserman, "Efficient object retrieval from videos," in *EUSIPCO '04*, 2004.

[2] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV '03*, 2003.

[3] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *CVPR '06*, 2006, pp. 2161–2168.

[4] M. Nestle, "The soft sell: how the food industry shapes our diets," *Nutrition Action Healthletter*, 2002.

[5] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *IJCV*, vol. 60, no. 1, pp. 63–86, 2004.

[6] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A comparison of affine region detectors," *IJCV*, vol. 65, no. 1-2, pp. 43–72, 2005.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[8] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in *ICCV*, 2005, pp. 604–610.

[9] Y. Zhang, L. Wang, R. I. Hartley, and H. Li, "Where's the weet-bix?" in *ACCV (1)*, vol. 4843, 2007, pp. 800–810.

[10] 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.

[11] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *CVPRW '04*, 2004, p. 178.

[12] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *ICCV '03*, 2003, p. 273.

[13] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *PAMI*, vol. 27, pp. 1615–1630, 2005.

[14] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *PAMI*, vol. 19, no. 5, pp. 530–535, 1997.

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