PAPER

Retrieval and Localization of Multiple Specific Objects with Hough Voting Based Ranking and A Contrario Decision

Pradit MITTRAPIYANURUK† and Pakorn KAEWTRAKULPONG††, Members

SUMMARY We present an algorithm for simultaneously recognizing and localizing planar textured objects in an image. The algorithm can scale efficiently with respect to a large number of objects added into the database. In contrast to the current state-of-the-art on large scale image search, our algorithm can accurately work with query images consisting of several specific objects and/or multiple instances of the same object. Our proposed algorithm consists of two major steps. The first step is to generate a set of hypotheses that provides information about the identities and the locations of objects in the image. To serve this purpose, we extend Bag-Of-Visual-Word (BOVW) image retrieval by incorporating a re-ranking scheme based on the Hough voting technique. Subsequently, in the second step, we propose a geometric verification algorithm based on A Contrario decision framework to draw out the final detection results from the generated hypotheses. We demonstrate the performance of the algorithm on the scenario of recognizing CD covers with a database consisting of more than ten thousand images. Our algorithm yield to the detection results of more than 90% precision and recall within a few seconds of processing time per image.

key words: recognition, localization, detection, image retrieval, object retrieval, scalable, large scale, bag of visual word, Hough voting, a contrario

1. Introduction

The current research on object recognitions can be divided into two major groups: (i) specific object (instance) recognition, and (ii) object-class recognition. Our proposed algorithm presented in this paper is along the lines of recognizing specific objects. Particularly, our concern lies in developing a scalable recognition algorithm that can simultaneously identify the identities and localize the locations of multiple planar textured-rich objects in an image. The algorithm can scale efficiently with a growing number of objects added into the database.

Recently, some works [1], [2] based on descriptor matching of local features (e.g. SIFT) were proposed to simultaneously recognize and localize multiple objects in images. However, these works can only perform efficiently with a small number of the objects in the database.

On the other hands, some works based on Bag-Of-Visual-Word (BOVW) image retrieval with Term Frequency-Inverse Document Frequency (TF-IDF) querying [3]–[6] are developed for large scale recognition tasks. However, one major limitation of these works lies in the assumption that there should be only a single object in the query image. Typically the accuracies of these works drop significantly when applied to images consisting multiple objects and cluttered backgrounds. This limitation was recently addressed in several works [7]–[10].

An example of the aforementioned limitation can be illustrated in Fig. 1. In this figure, there are totally 6 objects that simultaneously appear in the query image shown in Fig. 1(a). Specifically there are three different objects, i.e., CD-covers entitled LEBENS, ANTZ, and Dogz 3, where two instances for each specific object are located in different parts of the image. By applying a TF-IDF image retrieval [4], the top 5 most relevant database objects are listed in Fig. 1(b). We found that only one relevant object (LEBENS CD) is listed in the second rank. Meanwhile the other two relevant objects are ranked in the low orders, i.e. the 45th and the 407th orders (out of 11,444 database images), respectively.

To tackle the aforementioned limitation, we propose an algorithm that is extended from the state-of-the-art BOVW image retrieval by incorporating a novel ranking scheme based on Hough voting technique. Regarding to the same query image shown in Fig. 1(a), our proposed ranking scheme can efficiently retrieve all relevant objects in the top 5 ranking orders (2nd, 4th and 5th) as shown in Fig. 1(c). Furthermore, after applying the verification step of our algorithm, we also obtain the locations of the objects as shown with the bounding boxes in red color of Fig. 1(a).

With regard to the relevant literature [7]–[12], the novelty of our work lies in the aspect of combining and extend-

Fig. 1 Issue in large scale recognition of multiple objects: (a) A query image with 3 different specific objects (CD covers), each of 2 instances. (b) Top 5 retrieval result from TF-IDF image retrieval [4]. (c) Top 5 retrieval result from our object retrieval with Hough voting ranking. The locations of objects are also detected and shown with the red-colored bounding boxes in Fig. (a).
ing three major ideas to accomplish the task of simultaneous retrieval and localization of multiple specific objects with large scale database. These three main ideas include (i) the idea of large scale image search proposed in [3], [4], (ii) the idea of voting based recognition scheme proposed in [13], and (iii) the idea of multiple object recognition proposed in [14]. To our best knowledge, there is no previous work that proposes the similar scheme of combination and extension for accomplishing the aforementioned task [7]–[12]. Specifically, the key novelties of the proposed combination can be summarized as follows.

- As the hypothesis generation step, we propose to use a method combining a Hough voting based ranking scheme with large scale object image retrieval. Strictly speaking, our proposed scheme is an extension of the Takaki & Fujiiyoshi’s rotation-invariant voting scheme [13] into the large scale BOVW object retrieval framework [3]–[6]. Remark that, the key idea of our proposed ranking scheme is in the same spirit as the one presented by Shen et al. in [11]. However, unlike the Shen et al. approach, the scales and orientations of local features are used in our scoring process. This frees our method from additional computations for exhaustive pose-parameter search during the scoring process as required in [11].

- As the hypothesis verification step, we propose a method based on a-contrario decision framework [15]. Specifically, we propose to use Number of False Alarms (NFA) as the hypothesis quality score. We also propose a greedy based hypothesis selection that takes the main role in the task of multiple specific object recognition. Extended from the a-contrario based method proposed in [14], our greedy based algorithm is guided by the hypotheses initially generated by our proposed Hough voting based ranking.

The remainder of this paper are organized as follows. We present some related works in Sect. 2. The overview of our proposed algorithm is explained in Sect. 3. The details of the algorithm are presented in Sects. 4 and 5. The experimental result is reported in Sect. 6. Finally, the conclusion is drawn in Sect. 7.

2. Related Work

In [1], the author proposes an object recognition algorithm based on matching of local features, i.e., SIFT. The algorithm can recognize multiple objects simultaneously appeared in images. Recently some similar work [2] is also proposed. Generally, the main drawback of the approach based on descriptor matching of local features is that the algorithm could run very slowly with respect to a growing number of model objects added into the database.

To tackle the issue in large scale object recognition, the authors in [3] propose to use both the notion of vector quantization on local feature descriptors and the idea of TF-IDF indexing on quantized features (which is borrowed from text retrieval area) to efficiently search for near-duplicate frames in videos. This approach of image retrieval is usually referred to as Bag-Of-Visual-Words (BOVW) approach. This idea is extended with a faster vector quantization scheme referred to as Vocabulary Tree in [4]. Relatively, in [5] the authors propose to use Approximate K-Means clustering for feature quantization.

Generally, these BOVW retrieval techniques work inaccurately in the case of query images containing multiple objects and clutter backgrounds. Several modifications are proposed to tackle this problem. The most relevant works to ours are the ones proposed in [7]–[11] and [12].

In [7], the authors resort to a large scale object image retrieval using Vocabulary Tree [4]. However, they propose to modify the ranking score by incorporating a weight obtained during feature quantization step. To our best knowledge, this scheme does not directly address the issues of multiple objects and clutter background in images. Furthermore, they exploit a sliding-window based approach to localize the bounding boxes of the objects in the image. Generally, one disadvantage of sliding window approach is that the algorithm could run very slow even though it is applied as a post-processing step.

In [8], the authors resort to a variation of weak geometric consistency [6] in ranking relevant object images. They use a 2D vote space on keypoint orientation and scale in the ranking process to retrieve a set of topmost relevant images. Our proposed ranking scheme in this paper is very similar to this relevant work. Alternatively, we propose to adopt a voting scheme suggested by [13] in conjunction with Inverse-Document Frequency (IDF) [16] and [4] in the ranking process. In the case of query images with multiple objects, our proposed ranking scheme yields to a better recall than the authors’ scheme [8] that only uses the number of votes corresponding to the peaks of voted spaces.

Recently, Shen et al. [11] propose an idea that stays along the same line of ours in which the authors propose a method for retrieving and localizing a single object in images. As a more flexible system, our method proposed in this paper aims to tackle the issue of retrieval and localization of multiple specific objects. Furthermore, the Shen et al. method requires an exhaustive search for object pose parameters during the the ranking step. In Sect. 4.1, we propose the hypothesis generation step that resorts to a Hough voting based scoring scheme that is somewhat similar to the Shen et al. approach. However, in our approach, the information on scales and orientations of local features (keypoints) are used in such a way that our method does not need any exhaustive pose-parameter search to accomplish the scoring process as the one proposed in [11]. With regard to the processing time issue, our ranking scheme should be more efficient.

In [9] and [10], the goals of these works are exactly the same as ours, i.e., simultaneous recognition and localization of multiple objects with large scale database. The authors’ propose to use some variations of unsupervised clustering techniques to the local features extracted in query images. In particular, [9] uses a technique referred to as Adaptive Win-
Overview to the algorithm can be explained as follows: The block diagram of our algorithm is depicted in Fig. 2. An algorithm in the verification stage (Sect. 5.2) is inspired by this voting framework to detect multiple instances of a single step. In [19], Barinova et al. propose a probabilistic Hough voting framework to detect multiple instances of a single specific object. Our greedy based hypothesis selection algorithm in the verification stage (Sect. 5.2) is inspired by this work.

Regardless of scalability issue, Rabin et al. [14] propose an algorithm for recognizing multiple objects by extending the notion of a contrario RANSAC [17], [18]. Our work shares the same idea in which we exploit the quantity NFA [15] as the hypothesis quality score in our verification step. In [19], Barinova et al. propose a probabilistic Hough voting framework to detect multiple instances of a single specific object. Our greedy based hypothesis selection algorithm in the verification stage (Sect. 5.2) is inspired by this work.

3. Overview of the Algorithm

The block diagram of our algorithm is depicted in Fig. 2. An overview to our algorithm can be explained as follows:

1. **Training phase**: the training images of all target objects are enrolled into the system. Associated with each image, we also assume that a bounding box of the object is annotated. Then, the SIFT features are extracted from the images. Each feature is quantized to represent a visual word by using Vocabulary Tree [4]. Then an object is modeled with a set of keypoints derived from SIFT. Each keypoint is specified with 5 entries: keypoint location (x,y), scale (σ), orientation (θ) and visual word ID. Finally, an inverted file based indexing structure [3] is constructed from the keypoints of all training object images. The inverted file index is abstractly a table of size equal to the number of visual words in the vocabulary used in the quantization step. The data filled in each row entry of table is a collection of information about training image keypoints (i.e. object ID, x, y, σ, θ) that belong to the corresponding visual word.

2. **Testing phase**: given a query (test) image which may contain multiple relevant objects and clutter background, we want to determine the identities and the 2D locations of objects. This is achieved by the following steps.

   a. **Feature Extraction**: A set of SIFT keypoints are extracted from the test image. The corresponding visual words are obtained by quantizing the SIFT descriptors of keypoints.

   b. **Hypothesis generation**: A set of initial object hypotheses are obtained by using BOVW image retrieval with our proposed Hough voting ranking scheme. Each hypothesis is specified with an object ID and a coarse object center location in the image.

   c. **Hypothesis verification**: We propose to use an a contrario decision framework [15] in conjunction with a RANSAC based estimation of planar transformation (e.g. homography, affine) to draw out final detection result from the generated hypotheses. The detection result consists of a listed of object IDs and their corresponding bounding box locations in the image.

4. Hypothesis Generation Using Object Retrieval with Hough Voting Based Ranking

In this section, we will explain in details of our novel image ranking scheme based on Hough voting technique. This ranking scheme can overcome the shortcomings of most state-of-the-art algorithms on BOVW image retrieval [3]–[6] that are failed to recognize images consisting of multiple specific objects. Our proposed idea is in the same spirit of weak geometric consistency as proposed in [6], [8]. The algorithm for this part consists of the following steps.

4.1 Voting from Keypoint Matches

For each keypoint extracted in the test image, we establish a set of corresponding matches to the database image keypoints in which a matching is declared if the keypoints belong to the same visual word. These matches are efficiently retrieved with the help of the inverted file based indexing structure [3]. Each match will cast a vote for a candidate hypothesis in 3D Hough-vote space corresponding to object identity (Ov) and object center location (xv, yv) in the test image. This voted space is represented in the form of 3D accumulator bins (2 for image location and 1 for object ID).

For estimated object center locations, we resort to the (in-plane) rotation-invariant voting scheme as suggested in [13]. That is, see Fig. 3(a) for an illustration, a match between the test image keypoint specified with (xv, yv, σv, ŷv)}
to the object center in the test image. Meanwhile, the voted locations (plus markers) from wrong matches and the database keypoint (A computation, (b) Examples of voting: The voted locations (star markers) illustrate the voted location: (a) The quantities involved in the Fig. 3 − W markers) from wrong matches W − W′ to Z − Z′ are not clustered.

Fig. 3 Illustration of voted location: (a) The quantities involved in the computation, (b) Examples of voting: The voted locations (star markers) from correct matches A − A′ to E − E′ tend to cluster at a location closed to the object center in the test image. Meanwhile, the voted locations (plus markers) from wrong matches W − W′ to Z − Z′ are not clustered.

and the database keypoint (xᵣ, yᵣ, σᵣ, θᵣ) of the object identity Oᵣ will cast a vote for the entry (Oᵣ, xᵣ, yᵣ) of 3D voted accumulator. The voted object center location (xᵥ, yᵥ) can be calculated by the following equations.

\[
xᵥ = xᵣ + \sigmaᵣ \frac{\Delta \varphi}{\Delta \varphi} \cos(\theta + \varphiᵣ - \varphiᵢ) \tag{1}
\]

\[
yᵥ = yᵣ - \sigmaᵣ \frac{\Delta \varphi}{\Delta \varphi} \sin(\theta + \varphiᵣ - \varphiᵢ) \tag{2}
\]

where \(\theta = \tan^{-1}(\Delta \varphi / \Delta \varphi)\). And, \(\Delta x\) and \(\Delta y\) are the x- and y-distances between the database keypoint location to an arbitrary object reference location in the training image.

Specifically, in our implementation, we use the object center location derived from the centroid of bounding box vertices that are annotated with the training image. In Fig. 3(b), we show some graphical examples of the voted locations. Typically, the voted locations from correct matches tend to cluster at some position (see the grouping of star markers in the figure).

Moreover, we also take into account the effect of inter-image burstiness [16] by incorporating a vote weight obtained from IDF weight of the corresponding visual word [4]. That is, a match of visual word \(vᵣ\) will cast a vote for the location defined in (1) and (2) with the weight \(wᵢ = IDF(vᵣ) = \log(Nᵣ/ Nᵢ)\), where \(N\) is the total number of the objects in the database, and \(Nᵣ\) is the number of the database objects that consist of at least one visual word \(vᵣ\).

Undoubtedly, the effect of intra-image burstiness [16], i.e. repeated patterns, should also be taken into account. Especially, this effect could happen when there are several related or similar objects in the image. For example, the case of different coca-cola cans in which these bottles share many similar patterns. However, to tackle this effect still requires more in-depth investigation in which it may need some special steps to take care the problem. Therefore, we would like to leave this issue for future works.

To further reduce memory usage requirement, we quantize the location \((xᵥ, yᵥ)\) to vote for a coarser resolution accumulator bin. That is, we make a single voted location bin to cover 20x20 pixels. Therefore, for the image of size 640x480 pixels, this will require a 2D array of size 32x24 for the accumulator of one object identity. In the same spirit as [1], we diminish the boundary effects in bin quantization by additionally voting each match to neighborhood bins (±1) in voted x and y dimensions.

4.2 Find Peaks

Typically, if there exists an instance of database object in the test image, the weights accumulated in the voted space of the corresponding object identity will form some peak in the vicinity of estimated object center location. This peak will correspond to a hypothesis. An illustrated examples of voted spaces generated from the matches in the test image (previously shown in Fig. 1) is depicted in Fig. 4(b). In the picture, we show the voted spaces combined from five different voted spaces corresponding to the top 5 retrieved objects shown in Fig. 4(c).

Note that, the combined voted space is fictitiously created for the purpose of visualization only. Particularly, we visualize the voted weights as pixel intensities (0–255) in image space where the areas with lower intensity (darker) indicate stronger weight values than brighter areas. We also label the numbers 1 to 5 for the distinctive peaks corresponding to the hypothetical instances of these candidate objects.

Furthermore, it is noteworthy to mention that the peaks for the candidate objects 1 and 3 in Fig. 4(b) do not correspond to any actual object in the image. Subsequently, these incorrect hypotheses corresponding to the peaks will be rejected by the verification step as explained in the next section.

To finish the hypothesis generation task, a set of potential peaks in the voted spaces will be identified. In this work, we apply a conventional Non-Maximum Suppression (NMS) to select a set of distinctive peaks. Particularly, we opt to draw out the hypotheses corresponding to the topmost \(N_{\text{cand}}\) peaks where the value of \(N_{\text{cand}}\) could be in the range of 50 to 500.

For each identified peak, we generate a candidate hypothesis that is defined by an object identity and an estimated object center location corresponding to the peak location. We also accompany the estimated object center location with the supporting keypoint matches that vote for the
peak. To save the memory space requirement, we do not keep all information of keypoint matches during the voting process. Therefore, the supporting matches associated to the selected hypotheses (topmost $N_{cand}$ objects) will be recomputed. Explained in the next section, these keypoint matches will be used by a geometrical alignment based verification algorithm to localize an exact bounding box location of object in the image.

5. Hypothesis Verification Based on A-Contrario Decision Framework

In this stage, we will make a decision whether the likelihood of each generated hypothesis is strong enough to declare the presence of an object instance. This step will also resolve the ambiguity from conflicting hypotheses.

To decide whether a hypothesis is the actual object instance, we apply a geometric consistency verification method [5] that makes a decision based on planar transformations (e.g. affine or homography) estimated from the putative matches between the test image keypoints and the keypoints of a candidate database object. In general, we could make the decision by resorting to a simple approach of thresholding the number of inlier matches obtained after applying a geometrical model estimation (e.g. RANSAC homography estimation). In this context, an inlier match means a match that the residual error is less than a threshold. And, the residual error means the distance between the position of a test image keypoint and the transformed position of matched database keypoint. However, setting of a fixed threshold leads to several issues that need to be carefully handled. First, as the confidence measure, it is very hard to choose a single threshold on number of inliers for deciding the existences of object instances in such a way the chosen value could work well for most of images. For example, thresholding the number of inliers with some fixed value (e.g. 20 matches) may work in some test images. However, this value may not be suitable in several other test images (e.g. the images with smaller object sizes that averagely have less number of detected features). Furthermore, since our work concerns in the detection of multiple instances of the same object and/or multiple specific objects, we need a systematic way to compare the qualities of estimated models that supported by different sizes of keypoint matches. Also, we have another inherent issue on inlier/outlier discrimination in which we need to set a threshold on the residual errors of keypoint matches. To handle the aforementioned issues, we resort to the $a$-contrario approach [15] in the same manner as the ones used in [14], [17], [18].

For the conflicting hypothesis issue, we consider that any two or more candidate hypotheses are conflicted to each other if these hypotheses share some test image keypoints in their putative matches. An example of conflicting hypotheses in a test image can be shown in Fig. 5(a). In this image, it consists of only one object instance. However, we found three conflicting hypotheses that partially share putative keypoint matches. The first hypothesis (Spiderman CD-cover) corresponds to the actual object instance. Meanwhile, the second and third hypotheses correspond to invalid object instances. These three hypotheses share the test image keypoints in the vicinity of “PlayStation” logo of CD-cover. The corresponding bounding boxes of these hypotheses are also overlapped to each other. These bounding boxes are visualized with three different colors i.e., red, green and yellow in which the red bounding box is the correct one. Specifically, to resolve conflicting hypotheses, we propose a greedy based algorithm for selecting plausible hypotheses.

5.1 $a$-Contrario Based Hypothesis Quality Score

In short, the $a$-contrario decision methodology [15] is based on the Helmholtz principle in which a meaningful geometrical-based event, in an image, will be perceived if the likelihood that the corresponding event occurs by chance is very low. Specifically, the methodology associates a computational quantity, referred to as Number of False Alarms (NFA), to the geometric event. The NFA of an event is defined as the expectation of number of occurrences of the event under a background (noise) model. This background model is referred to as $a$-contrario model. In general, an event with very low value of NFA is considered as meaningful event.

In the literature, there are several variations on works based on $a$-contrario model (see the recent monograph [15] for a comprehensive review) in which basically they are different on how to mathematically define and compute NFA that are exploited in various application contexts e.g. feature grouping, motion/change detection, feature matching, etc. Particularly, in our proposed algorithm, we resort to the idea of using $a$-contrario approach for measuring the likelihood of a hypothesis (i.e., hypothesis quality score) in term of Number of False Alarms (NFA).

The value of NFA in our proposed method is computed in the same manner as the methods of using $a$-contrario ap-
proach for geometrical model estimation given a set of local feature matches between two images [14],[17],[18]. That is, the formula for computing the NFA can be expressed by:

\[
NFA = \min_{c+1 \leq k \leq N} n_{fa}(k)
\]

where

\[
n_{fa}(k) = (N - c) \left( \begin{array}{c} N \\ k \end{array} \right) \left( \frac{\epsilon_2}{\epsilon_k} \right)^{k-c}
\]

and

- \(N\) is the number of putative keypoint matches associated with a hypothesis.
- \(c\) is a constant in which \(c = 3\) for the case of affine transformation and \(c = 4\) for the case of homography.
- \(\epsilon_k\) is the \(k\)th least residual error among \(N\) putative matches given an estimated model of planar transformation (affine or homography).
- \(w'\) and \(h'\) are the width and height of the test image.

The function \(n_{fa}(k)\) in (4) can be considered as a measure to evaluate the quality of the candidate set of \(k\) inliers in which we consider that the threshold on residual errors for inlier/outlier discrimination is \(\epsilon_k\). This measure combines the quantity of residual errors and the set size (number of inliers) into a single quantity. To find the best set of inliers, we search for the value of \(k\) in the range of \(c + 1\) to \(N\) that minimizes \(n_{fa}(k)\) as expressed in (3). For the value of \(k\) that yields to the minimum \(n_{fa}(k)\) (i.e., \(NFA\)), the set of \(k\) inliers corresponding to the matches that have the lowest \(k\) residual errors are the best set of inliers.

Given a threshold \(\epsilon\), if \(NFA \leq \epsilon\); consequently we consider the set of matches (the hypothesis) to be \(\epsilon\)-meaningful. This implies that the best set of inlier matches corresponding to the value of \(NFA\) in (3) is likely to be the matches corresponding to an actual object instance in the test image. Moreover, the smaller that the value of \(NFA\) is will indicate that the corresponding hypothesis is more meaningful. This is useful in the comparison of hypotheses with different sizes of keypoint matches. Usually, the \(NFA\) values of correct hypotheses are very small (e.g., \(10^{-30}\)). In general, the value of this threshold \(\epsilon\) can be fixed to 1 (or \(10^{-3}\) in some works). Therefore, if the \(NFA\) value of a hypothesis is larger than the threshold, we reject the hypothesis.

In our algorithm, we intend to make a larger value of hypothesis quality score means that the hypothesis is more likely to be correct. Therefore, we define the quality score of a hypothesis by the negative of logarithm of \(NFA\). That is, for the \(NFA\) of a given hypothesis \(h_i\), the quality score is given by: \(Q(h_i) = -\log(NFA)\).

With regard to the aspect of implementation, given a set of putative matches of a hypothesis, the value of \(NFA\) can be computed according to (3) and (4) by using a random sampling based procedure (e.g. RANSAC). That is, in each iteration of the estimation, first we randomly choose \(c\) samples out of \(N\) putative matches and use them to estimate the parameters of planar transformation. Then, the residual errors of all \(N\) matches are calculated according to the estimated parameters. And, all matches are sorted in the increasing order by their residual errors. If we denote \(\epsilon_1, \epsilon_2, ..., \epsilon_k, ..., \epsilon_N\) be the residual errors of sorted matches, then we compute the values of \(n_{fa}(k)\) for \(k = c + 1, c + 2, ..., N\) by (4). Remark that, the values of \(\epsilon_1, \epsilon_2, ..., \epsilon_c\) are very close to zero, since they correspond to the samples used in the estimation. Next, as in (3), we compute the value of \(NFA\) in which it is equal to the minimum value of \(n_{fa}(k)\). For the \(k\) that yields to the value of \(NFA\), the set of inlier matches is the matches corresponding to the lowest \(k\) residual errors. We repeat these steps until either a meaningful inlier set (\(NFA \leq \epsilon\)) is found or the number iterations performed are greater than a maximum number of iterations allowed.

By using the \(a\)-contrario approach, we gain several benefits with regard to the task of multiple object localization. First, the \(NFA\) can be used as a confidence measure to decide the existence of object instance in which generally the threshold on \(NFA\) can be fixed. Furthermore, the score based on \(NFA\) allows us to compare the hypotheses that have different sizes of matches. Finally, as presented in [14],[17],[18], we do not have to specify the threshold for discriminating the inlier/outlier matches as required in the conventional RANSAC approach. The value of inlier/outlier threshold is adaptive during the \(NFA\) computation process.

5.2 Verification Algorithm

To fulfill the verification stage, we propose a greedy based algorithm that iteratively selects the potentially valid hypotheses among the conflicting hypotheses. Before going into details, we want to emphasize that our algorithm is based on the key assumptions as follows. First we assume that a keypoint in the test image can contribute to only a single object instance. Second, we also assume that there is no partial occlusion in the test image. Although we found that our algorithm can resist to some certain degree of partial occlusions, we leave this problem as a future work.

Expressed with a pseudo-code in Algorithm 1, our verification algorithm performs greedily in the hypothesis selection in which a single best hypothesis at each iteration is pulled out from the list of candidate hypotheses at Line 12. To compare among hypotheses, we adopt the \(NFA\) based quality score where the corresponding pseudo-codes are on the lines 5 to 9. Particularly, the \(A_{\text{Contrario RANSAC Planar Estimation}}\) procedure at Line 6 will compute the value of \(NFA\) given the set of putative matches of a hypothesis by using the method based on RANSAC as explained in Sect. 5.1.

If the quality score of a selected hypothesis is still greater than the fixed threshold (\(-\log(\epsilon)\)), we add the hypothesis into the detection result at Line 21, where the bounding box of the detected object instance in the test image is determined by the back-projection of database object bounding box as in Line 20. Before starting the next iteration, we remove the matches, that are associated with the test image keypoints of the selected hypothesis from other
remaining hypotheses (Line 23–25). We iterate these steps until no hypothesis whose score is greater than the threshold is found where the stop flag is set to true in Line 27 to indicate the termination of iterations. The final result in L is the list of object instances detected in the test image. An example of the iterations of verification algorithm on the same test image in Fig. 1 can be illustrated in Figs. 5 (b) to (d). Note that the incorrectly generated hypotheses corresponding to the peaks labeled with the numbers 1 and 3 in Fig. 4(b) are rejected by the verification algorithm.

6. Experiments

6.1 Implementation Details

We implement the proposed algorithm by using several toolkits. For feature extraction, we use the SIFT implementation from VLFeat (http://www.vlfeat.org). We exploit VOCSEARCH [20] for the task of visual word quantization with Vocabulary Tree [4] and the task of inverted file indexing [3]. We also used the pre-trained vocabulary tree consisting of 1M visual words that is provided by VOCSEARCH. We implement BOVW object retrieval with Hough voting (Sect. 4) in C++ by modifying the source code provided by the VOCSEARCH. Finally, our hypothesis verification (Sect. 5) is implemented with MATLAB in which we follow the guideline provided by Moisan et al. in [18]. In addition, we also incorporate a rigidity constraint as suggested in [21] into our RANSAC homography estimation. The main purpose of enforcing the constraint is to prevent non-physically meaningful homographies.

6.2 Datasets

We evaluate our proposed algorithm with the Caltech Game CD/DVD covers dataset [22]. This dataset consists of 11,400 images for CD/DVD covers of video games. We use these images as the training images. One training image is assigned to an unique object identity. The sizes of these images are about 400x400 pixels. After SIFT extraction, we found that roughly there are about 1,000 SIFT keypoints extracted in each training image.

For collecting the test images, we select 50 CD covers, print out these images with a color laser printer in the actual size and put them into jewel CD cases. Then we take the pictures of these CD covers within cluttered background environments with a digital camera. The original resolution of images we capture is 1600x1200 pixels. All test images are captured in color, but are converted to gray scale before applying to our algorithm. Nevertheless, for the purpose of processing time, we resize the images to 1024x768 pixels before applying the SIFT extraction module. The number of SIFT keypoints in each test image is in the range of 3,000 to 5,000 keypoints. Strictly speaking, we can consider the selected 50 CD covers as the probe set whereas the remaining ones (about 11,350 CDs) are the distracter.

We created 8 different sets for test images. The images in these sets consisted of a varying number of CD covers in images as explained in the first two columns of Table 1. In the second column, Ni and Nc are the number of different specific CD types and the number of instances of each types in an image, respectively. Meanwhile, Nj is the number of images in the test set. For example, the test set 8 consists of 35 images; each image consists of 3 different CD covers with 2 instances of each specific CD cover. These different sets demonstrate several configurations of objects in images (e.g. multiple instances of the same specific object, multiple specific objects and multiple instances/multiple specific objects). Totally, there are 495 images with 1,446 instances of CD-covers in all test sets. For the ground truth, we manually annotate an object identity and bounding box vertices for each CD cover instance in the test images.

6.3 Results

To evaluate the detection results, we justify each detection result as either true positive or false positive. A detection result is considered as a true positive if (i) it has the same object identity as the one of ground truth, and (ii) ratio of intersection over union between the detected bounding box and the ground truth bounding box is larger than 0.5. Then,
Table 1  The description of different test sets (first two columns), the detection performance of our algorithm and the detection performance when the verification based on the number of inliers.

<table>
<thead>
<tr>
<th>Set</th>
<th>nP (NtxNt)</th>
<th>Ours</th>
<th>NumInliers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100 (1x1x100)</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>220 (1x2x110)</td>
<td>0.91</td>
<td>0.98</td>
</tr>
<tr>
<td>3</td>
<td>400 (1x4x100)</td>
<td>0.98</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>100 (2x1x50)</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>120 (4x1x30)</td>
<td>0.96</td>
<td>1.00</td>
</tr>
<tr>
<td>6</td>
<td>208 (2x2x52)</td>
<td>0.93</td>
<td>0.99</td>
</tr>
<tr>
<td>7</td>
<td>108 (6x1x18)</td>
<td>0.96</td>
<td>1.00</td>
</tr>
<tr>
<td>8</td>
<td>210 (3x2x35)</td>
<td>0.85</td>
<td>1.00</td>
</tr>
<tr>
<td>Total</td>
<td>1,466</td>
<td>0.941</td>
<td>0.995</td>
</tr>
</tbody>
</table>

Fig. 6  Some qualitative results on different test sets. The back-projected bounding boxes of detected objects are shown in red.

we measure the accuracy of our proposed method on each test set with the precision (prec = TP/(TP + FP)) and the recall (rec = TP/nP), where TP is number of true positives, FP is number of false positives, nP is total number of positives in a test set.

The detection results of our algorithms on eight test sets are listed in Table 1 (the columns Rec. and Prec. under the heading Ours). Some of the qualitative results for each test set are shown in Fig. 6. Of all test sets, our algorithm can detect 1,379 instances of CD covers (TP) from totally 1,466 CD-covers (nP) in all test images. Also, the algorithm returned very small number of false positives (7 out of 1,386 detections). Roughly, our proposed algorithm yields to more than 90% of recalls and precisions in all test sets. Besides by the main role of a-contrario based geometric verification, we observe that enforcing the rigidity constraint in RANSAC process as suggested in [21] to reject inconsistent homographies is also helpful for reducing number of false positives.

6.3.1 Effectiveness of the a-contrario Based Verification

In this section, we show the effectiveness of the hypothesis verification using the a-contrario approach as proposed in Sect. 5 by comparing with a baseline verification that makes a decision based on the number of inliers. Specifically, the pseudo-code of the baseline verification is similar to the one shown in Algorithm 1, except that in Line 6 and Line 8. In the line number 6, the routine named $A_{Contrario\_\text{RANSAC\_Planar\_Estimation}}()$ is replaced with the procedure that performs a conventional RANSAC based planar transformation estimation. Meanwhile, in the line number 8, now the hypothesis score is assigned with the number of inliers found by the RANSAC based procedure. To establish a fair comparison, for both the a-contrario based verification and the baseline verification, we use exactly the same intermediate results that are obtained from the hypothesis generation step. The comparative results on the dataset in Sect. 6.2 are shown in Table 1 in which the precision (Prec.) and the recall (Rec.) of the baseline are shown in the columns under the heading NumInliers. For the threshold setup on the RANSAC parameters of the baseline, we set the value of the threshold for inlier/outlier discrimination to 10 pixels and we decide the existence of an object instance if at least 10 inliers is detected (i.e., minimum number of inliers is 10).

As it can be seen from the result, in every test set, our proposed method using the a-contrario based verification gives better recall rates than the ones of the baseline verification that uses the simple decision based on the number of inliers. With regard to all test sets, the recall rate of the baseline is 0.924 in which it can detect only 1,355 instances of CD covers from totally 1,466 instances. Meanwhile, the recall rate of our method using the a-contrario based verification is 0.941 in which it can detect 1,379 instances of CD covers (i.e., 24 more instances). Although, the precision rate of our method is slightly less than the baseline. Both methods give the very high precision rates (0.995 for ours and 1.00 for the baseline). In Sect. 6.6, we also perform another similarly comparative evaluation by using a different dataset. From the results shown in that section, the a-contrario based verification still outperforms the baseline.

6.4 Processing Time

The averaged processing time per image of our algorithm is presented in Table 2. We measure the processing times of algorithm on a PC with Intel Core 2 Duo 3 GHz and 2 GB of RAM. We want to emphasize that the verification step presented in Sect. 5 is still implemented with MATLAB.

6.5 Comparative Results on Large Scale Image Search

In this section, we report the comparative result between our proposed ranking scheme based on Hough voting technique and a baseline method. In particular, we use the Nister and Stewenius’s large scale image search approach proposed in [4] as the baseline method. We perform this evaluation on the Caltech Game CD/DVD covers dataset as explained in
Table 3 Comparison of recall rates (in percentage) between our Hough voting based ranking and the Nister & Stewenius (N-S) approach [4] at $N_{top}=20, 60,$ and 100.

<table>
<thead>
<tr>
<th>Test set</th>
<th>Ours</th>
<th>N-S approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N_{top}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>60</td>
</tr>
<tr>
<td>1</td>
<td>99.0</td>
<td>99.0</td>
</tr>
<tr>
<td>2</td>
<td>92.7</td>
<td>96.4</td>
</tr>
<tr>
<td>3</td>
<td>98.0</td>
<td>99.0</td>
</tr>
<tr>
<td>4</td>
<td>97.0</td>
<td>100.0</td>
</tr>
<tr>
<td>5</td>
<td>93.3</td>
<td>95.8</td>
</tr>
<tr>
<td>6</td>
<td>91.3</td>
<td>98.1</td>
</tr>
<tr>
<td>7</td>
<td>91.7</td>
<td>97.2</td>
</tr>
<tr>
<td>8</td>
<td>84.8</td>
<td>94.3</td>
</tr>
</tbody>
</table>

 Sect. 6.2. The purpose of this comparison is to validate our proposed image ranking scheme with respect to the conventional large scale image search algorithms.

We want to emphasize that this comparison is performed only at the level of object retrieval to draw out a list of candidate objects in images without applying any geometric verification as post-processing. For the comparative evaluation of overall effectiveness, i.e., both retrieval and localization results, we will report the result in Sect. 6.6.

To compare these two approaches, we will evaluate how well the algorithms can retrieve a set of relevant database objects in the top ranked orders by measuring the recall (in percentage) of the results within the top database candidates. If the query image consists of multiple instances of a single specific object, we will consider them as one sample instance in the recall computation. Furthermore, we vary the number of top database candidates, i.e., $N_{top}$ to be 20, 60, and 100. Thus, at any value of $N_{top}$, if the retrieval results in the top $N_{top}$ ranked list hit the labels in the ground truth, we will count these results in the recall calculation.

The results for this comparative evaluation can be shown in Table 3. From the result, our proposed ranking scheme outperforms the baseline method. Our algorithm yields to above 90% of recall at all values of number of top database candidates ($N_{top}$). Meanwhile, in the test sets 5, 7 and 8, the baseline method is completely failed to retrieve the relevant objects in which the recalls are less than 50% at all values of number of top candidates.

One interesting point observed from the result is that the Nister & Stewenius approach [4] only provides the good results (% of recalls $>70$%) on the test sets 1 to 3. As a matter of fact, the query images in these test sets consist of only one specific object where the number of instances of a specific object in each query image are 1, 2 and 4, respectively. As the number of instances of the same specific object in images is increasing, the Nister & Stewenius approach seems to perform more accurately.

This phenomenon can be explained regarding to the key notion of the BOVW image retrieval as follows. Since the BOVW engine will favorably search for database objects whose collections of visual words (without considering spatial information) are similar to the ones extracted from the whole query image. Therefore, as the number of identical object instances is increasing, the same collections of visual words in the query image will be elevated with more relevant visual words extracted from incremental object instances in which it will bring out more correctly relevant database images into the final query result. In contrast to the state-of-the-art proposed in [4], our proposed algorithm can solve the issues of both multiple specific objects and multiple instances of the same object.

6.6 Comparative Results on Simultaneous Recognition and Localization of Multiple Objects

In this section, we report the comparative results on overall effectiveness between our proposed method and other previous works presented in [9], [10]. Particularly, we evaluate both the retrieval and localization capabilities of the methods.

To accomplish the evaluation, we use the other dataset kindly provided from the authors of [9] and [10]. This dataset consists of the local features extracted from about 251,000 book images. For each book image, a set of Hessian-Affine regions and SIFT descriptors are extracted. Each SIFT descriptor is quantized into a visual word by using the authors’ trained vocabulary tree of 1M leaf nodes (visual words). For the test set, it consists of 27 images containing 155 books.

To adapt this dataset to our proposed method, we directly derive the keypoint scales from the hessian-shapes of hessian-affine regions. Unfortunately, due to some circumstance of dataset availability, there is no information on keypoint orientation of each feature in the dataset. For the sake of simplicity, we assume that the database objects and the objects in query images almost appear upright. Therefore, we can assume that the orientations of both the database keypoints and the object keypoints in query images are all the same with zero value.

By using the same evaluation protocol as presented in [9], [10], the comparative results in terms of precision/recall rates can be shown in Table 4. To obtain the comparable precision/recall rates, we ignore any detected instance whose $NFA$ value is larger than $10^{-5}$. Furthermore, to visually appreciate the comparison, we also show some examples of quantitative results on multiple object localizations in Fig. 7.

As we can see from Table 4, our proposed method outperforms both the Lin et al. approach [9] and the Wu et al. approach [10] in terms of recall rate. Furthermore, the precision rate of our method (92.8%) is comparable to the best of these two approaches (93%).

Particularly, we want to emphasis again that the results obtained from our algorithm reported here is subject
Fig. 7 Examples of qualitative results of our method, Lin et al. [9] and Wu et al. [10].

to the condition that the keypoint orientations are unknown and assumingly approximated to zero. As far as we concern, we believe that if the information on keypoint orientations are available on the dataset, the result from our proposed method should be even better than the currently reported result.

With regard to the inherent ideas of [9] and [10], the localization results obtained from our method are not limited to a grid-based manner, i.e., rectangular-shaped bounding boxes (see the bounding boxes in Fig. 7 for the comparison). Hence, our method could give more flexible and more accurate localization results. This could be an advantage especially in context of detection and tracking of multiple objects as in many augmented reality applications. Specifically speaking, our method can be used as the detection module to localize the tight bounding boxes of objects in the first frame as the tracker initializations. Then it can hand over the task to the tracking module for processing the subsequence frames.

Continuing from Sect. 6.3.1, we perform an additional experiment on this book dataset to show the effectiveness of the a-contrario based hypothesis verification with regard to the baseline verification that makes a decision based on the number of inliers. The results of the method using the baseline verification is shown in Table 4 under the column heading named NumInliers. In terms of both precision and recall rates, our proposed method using the a-contrario based verification still outperforms the method with the baseline verification in the same way as reported in Sect. 6.3.1.

6.7 Failure Cases

There are some cases when our algorithm fails to detect the objects in images. First, our algorithm still produces some false positive results as shown in Fig. 8(a). This failure is due to the nature of a-contrario frameworks that are inferior if there is small number of inlier matches involved in the NFA computation as mentioned in [18]. Second there are some miss-detections as shown in Fig. 8(b) where the dashed circles are plotted to indicate the miss-detections. From our observation, these cases are mostly due to adverse illumination changes (e.g. specular noise or glares at CD cases). Finally, our algorithm completely fails to detect the objects in the presence of significant viewpoint changes as illustrated with some examples in Fig. 8(c). As mentioned in [23], the problem could be solved by adopting a visual vocabulary that takes into account of viewpoint changes.

7. Conclusions

We have presented a scalable recognition algorithm for simultaneously identifying the identities and detecting the locations of multiple objects in images. Our approach is extended from Bag-Of-Visual-Word (BOVW) image retrieval by incorporating a novel Hough voting based scoring. We also incorporate an a-contrario based decision framework into our greedy based hypothesis verification. The evaluation with a large scale object database on a set of test images of CD-covers have shown the promising results of our proposed algorithm.

Some of possible future works can be listed as follows. First, we are interested in applying the probabilistic Hough voting framework proposed in [19] into our hough voting scoring. This could make our hypothesis generation to be more robust than the current algorithm that is still based on the conventional non-maximum suppression (NMS). Second, with regard to the applications, it is useful if we could make the algorithm to be robust to viewpoint changes. This problem is also central to the works on BOVW large scale image search. Finally, as we mentioned in Sect. 4.1, the issue of objects with repeated patterns, i.e. intra-image burtiness, should be investigated in greater details. A real practical scenario that could be problematic to our proposed method is the case when it is applied to detect the products that have several packaging versions that are very similar.

Acknowledgments

This work is supported by the Higher Education Research Promotion and National Research University Project of
Thailand, OHEC.

References


