Hyperlink-Aware Object Retrieval

Wei Zhang, Chong-Wah Ngo, and Xiaochun Cao

***Abstract*— In this paper, we address the problem of object retrieval by hyperlinking the reference data set at subimage level. One of the main challenges in object retrieval involves small objects on cluttered backgrounds, where the similarity between the querying object and a relevant image can be heavily affected by the background. To address this problem, we propose an efficient object retrieval technique by hyperlinking the visual entities among the reference data set. In particular, a two-step framework is proposed: subimage-level hyperlinking and hyperlink-aware reranking. For hyperlinking, we propose a scalable object mining technique using Thread-of-Features, which is designed for mining subimage-level objects. For reranking, the initial search results are reranked with a hyperlink-aware transition matrix encoding subimage-level connectivity. Through this framework, small objects can be retrieved effectively. Moreover, our method introduces only a tiny computation overhead to online processing, due to the sparse transition matrix. The proposed technique is featured by the novel perspective (object hyperlinking) for visual search, as well as the object hyperlinking technique. We demonstrate the effectiveness and efficiency of our hyperlinking and retrieval methods by experimenting upon several object-retrieval data sets.**

***Index Terms*— Object retrieval, hyperlinking, re-ranking, object mining.**

I. Introduction

IN THE context of Web search, a hyperlink refers to a "pointer" between two web pages, where the source page has an HTTP link that references the destination page and is associated with some anchor text. While hyperlink analysis is known to play a critical role in Web search, the creation and exploitation of hyperlinks for visual search remains a topic seldom studied. The ability to cross-reference different visual entities (i.e., objects/locations/persons) in an image or video collection, for example, can greatly facilitate applications such as advertising, browsing and retrieval. Despite these advantages, manually creating hyperlinks in the visual domain is far less convenient and motivated, than creating hyperlinks among web pages. This paper studies the problem of object retrieval via automatic hyperlinking of visual entities. Unlike existing works (e.g., [1], [2]) that operate on frame-level links, we explore more general hyperlinks connecting visual entities with arbitrary sizes.

In this paper, the objects under consideration are assumed rigid, each with a well-defined boundary and a center [3]. Examples include buildings and logos, as opposed to amorphous background. Retrieving objects efficiently from a large-scale dataset is a feature highly demanded with the convergence of mobile computing and e-commerce. The popularity of e-commerce gives rise to the needs for retrieving small objects (e.g., products, logos, locations). At the same time, smartphones provide the right tool for precisely defining the object under query. Users can easily formulate a query using the portable camera and touch screen. Undoubtedly, object retrieval will play an indispensable role in this retrieval paradigm.

Despite the recent advancement of Deep Convolutional Neural Network features (DCNN) [4]–[6], state-of-the-art object retrieval techniques are mostly based on the Bag-of-Visual-Words (BoW) model [7], [8] and local features [9]. Combined with inverted file, BoW achieves a good trade-off between efficiency and effectiveness. Usually, reference images are ranked according to their cosine similarity to the query: $\text{rel}(\text{Q}, \text{R}_i) = \frac{\text{Q}\cdot\text{R}_i}{\|\text{Q}\|\|\text{R}_i\|}$, where Q and R_i represent the BoW vectors of the querying object and the *i-th* reference image, respectively. Intuitively, the numerator counts the *common parts* between the query and reference images, while the denominator normalizes the score. In the context of object retrieval, the query is usually a small Region of Interest (ROI), and the targets on reference images are also likely to be small and have different backgrounds. In this paper, we regard an object as *small* if it covers less than 10% of the area on the image. Taking TRECVID dataset (Section V-A) for example, 77% of the query objects are small. Apparently, $\|\text{R}_i\|$ "*over-norms*" the score of small objects on the reference image by including features on the background. Due to this "over-norm" effect, retrieving small objects is challenging in practice. However, by using object-level hyperlinks, we show that the effect can be bypassed during search re-ranking.

Another research problem for object retrieval lies in the speed efficiency. A quick online response is extremely important to users in real-world applications. We refer to *offline* (*online*) processing as the computational process involved before (after) observing the query. In the past decade, most techniques have focused on the online part, e.g., query expansion [10], spatial verification [8], and multiple assignment of visual words [11]. However, these methods inevitably slow down the online response by introducing extra computation. Studies on offline processing are limited to indexing techniques [7], [12] and feature augmentation [2], [13]. In general,

Manuscript received August 5, 2015; revised February 10, 2016 and May 19, 2016; accepted July 6, 2016. Date of publication July 11, 2016; date of current version July 21, 2016. This work was supported by the Research Grants Council, Hong Kong under Grant CityU 118812. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Wen Gao.

W. Zhang is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China, and also with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: wzhang.cu@gmail.com).

C.-W. Ngo is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: cscwngo@cityu.edu.hk).

X. Cao is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China (e-mail: caoxiaochun@iie.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2016.2590321

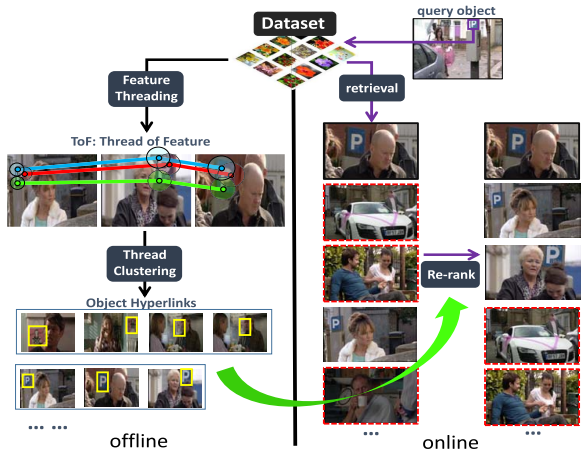


Fig. 1. The proposed framework for hyperlink-aware object retrieval. First, subimage-level hyperlinks are constructed on the reference dataset via offline mining (left). Then the initial retrieval results are re-ranked with the hyperlinks (right).

optimizing the online process is suitable for improving the retrieval precision because the query is already known. In contrast, the query remains unknown for offline processing, and arbitrary queries must be considered during offline processing. However, optimizing the offline process is better for reducing the online response time. Parsing the dataset offline to make online calculations more efficient is a general trend in big data computation. In this paper, we specifically address the efficiency problem by performing offline mining and by indexing frequent visual entities.

To address these problems, we propose an object retrieval technique via hyperlinking, as shown in Fig. 1. During offline processing, object-level hyperlinks are constructed by mining sub-image level objects from the reference set. Then, during online retrieval, the initial ranklist is re-ranked by the hyperlink-aware re-ranking. It is worth noting that the term "hyperlinking", as used here, is slightly different from Web hyperlinking. Web hyperlinking usually links an anchor text in a webpage to another webpage, whereas our hyperlinking connects several instances of the same visual entity across multiple images (Fig. 4). In this work, the object mining technique, which is to identify frequently appearing objects in a dataset, is adopted for hyperlinking objects among reference images. We first propose an object mining method and then study proper ways of re-ranking using the resulting hyperlinks.

For the "over-norm" problem, our strategy leverages the hyperlinks constructed via object mining, which establishes subimage-level rather than image-level links. Although the initial retrieval results suffer from the over-norm problem, the subimage-level hyperlinks address this problem by excluding cluttered backgrounds from hyperlinking, as well as by re-ranking using the subimage-level hyperlinks. Therefore, linking objects is far more important than linking images because in most cases, similar images do not suffer from the over-norm problem. Toward this end, we propose a scalable object mining method by exploiting the Thread of Features (ToF). Specifically, ToF is a compact representation that links consistent features across images and is extracted to reduce noise, discover objects, and speed up processing. More importantly, small objects can be easily discovered by exploiting correlated ToFs.

To address the efficiency problem, we exploit offline processing and introduce only minor computation overhead into the online retrieval process. Because the major part of the computation is completed offline, our solution is more efficient by its very nature. At the time of online retrieval, we re-rank the initial results through Random Walk [14], where a hyperlink-aware transition matrix is constructed for subimage-level connectivity. Due to the sparsity of the transition matrix, our re-ranking method is more efficient than traditional visual-based Random Walk such as [15], which operates on a dense transition matrix encoded with pairwise similarities of the top-K retrieved images. Moreover, existing visual re-ranking considers only the top-ranked shortlist. Images ranked beyond the shortlist are omitted during re-ranking. As a more general approach, hyperlink-aware re-ranking does not suffer from this limitation. To be precise, images that are ranked lower due to the over-norm effect can be upgraded to higher ranks with the aid of offline-mined hyperlinks.

This paper addresses the problem of object retrieval using an offline hyperlinking process. The hyperlinking step mines objects effectively and efficiently. Then, our retrieval step utilizes the offline mined knowledge to retrieve small objects that typically suffer from the "over-norm" problem. The main contributions of this paper are summarized as follows:

- We address the retrieval problem via offline hyperlinking, which saves online computation and alleviates the over-norm problem. Using the knowledge mined offline, our solution is capable of dealing with the over-norm problem, and of re-ranking results beyond the shortlist.
- We propose an object hyperlinking method that automatically mines frequently appearing objects in a dataset. Compared to existing methods, our method mines small objects more effectively and efficiently.

This manuscript is built upon our previous conference paper [16] on object mining. In this paper, we investigate a different problem of object retrieval by studying proper re-ranking strategies using hyperlinks constructed offline. In addition, this paper includes more experiments and analysis to evaluate the performance of our object retrieval method. The rest of the paper is organized as follows: Section II reviews related works; Section III presents our algorithm for hyperlinking, and Section IV discusses the hyperlink-aware re-ranking; Section V evaluates the hyperlinking and re-ranking methods on different datasets; and Section VI concludes this paper.

II. Related Works

Because our study explores the potential value of hyperlinking for object retrieval, it is highly related to both object mining and retrieval. It is worth noting that for the purpose of object retrieval, we are particularly interested in hyperlinking frequent subimage-level objects, other than frequent images such as in [17] and [18].

A. Object Mining

Most previous studies only operate on small-scale datasets. Early studies on Common Pattern Discovery [19]–[21] model

this task as an optimization problem. These methods are computationally expensive, and are therefore limited to only hundreds of images. Furthermore, most common objects in these works are the saliency [22], [23] of each single image, which makes the problem easier. Frequent Itemset Mining (FIM) was initially used to find sets of products bought together (e.g., beer and diaper). Quack [24] introduced FIM in visual mining. However, because of the computationally expensive support-counting, in practice, this method can address only thousands of images. Sivic and Zisserman [25] extracts key objects and characters from a movie by clustering locally grouped features. However, this method is still slow due to its pairwise similarity evaluation. Moreover, it can only discover objects with predefined sizes.

Only a few methods are capable of scaling up for large datasets. Letessier *et al.* [26] used Random Maximum Margin Hashing to generate a prior distribution and then adaptively sampled and verified frequent objects. Pineda *et al.* [27] extracted objects by clustering visual words, where each visual word was represented as a set of images containing the visual word. This method is effective in small datasets, but its performance decays quickly as the number of images increases. Geometric min-Hash (GmH) [28] extended min-Hash [29] by considering the dependency among visual words. For each sketch, it computes the first hash key as standard min-Hash does, but then chooses secondary hash values within a local proximity of the first key. This method improves the collision probability for small objects. However, this performance boosting occurs only when the first key repeats, which is still difficult to achieve for small objects.

B. Object Retrieval

Object retrieval has been a topic of research for almost a decade. Previous studies have focused primarily on improving feature representation [9], [30]–[32], feature matching [2], [33], [34], and spatial verification [8], [35], [36].

Until quite recently, most works are designed for general image retrieval. Only a few works formally study the object retrieval problem. Recent works in TRECVID Instance Search [37] address the small object problem on both query and reference images. Ran [38] partitions each reference image into segments using Selective Search [39], where each segment is indexed and retrieved independently. Thus, the segments are matched directly to the query rather than to the full image. Obviously this method relies heavily on the quality of the partitioning mechanism. On the query side, Zhang and Ngo [36] proposed to weight the background context for query modeling, which increases the information quantity for small query objects.

There are also a few object retrieval methods that tweak the reference dataset. Query expansion [10], [40] augments the query using the highest-ranked retrieved images. However, doing this introduces considerable additional online computation because all computation can be conducted only after analyzing the query. Database-side feature augmentation [13] augments each image in the reference set with features from similar images. Spatial database-side feature augmentation (SPAUG) [2] further improves [13] by restricting feature augmentation to spatially verified image areas. All these methods require a matching graph [17], which can be regarded as image-level hyperlinking. However, this matching graph is computationally expensive to construct, and only favors similar images rather than sub-image objects.

Recent benchmark evaluations such as MediaEval [41] and TRECVID [37] have included the hyperlinking task. These evaluations consider a variety of entities such as events, concepts, people and name entities for hyperlinking large video archives. Textual cues, especially speech transcripts, play a crucial role in linking these entities. In contrast to our work, the focus is to recommend hyperlinked video snippets given a query clip rather than on object retrieval. This paper considers specifically visual-level entity linking and the exploitation of hyperlinks for online object retrieval.

TABLE I
KEY NOTATIONS USED IN SECTION III

Notation	*Definition*
l	number of neighboring features coupled into a patch
t	number of common visual words threshold for threading patches
I_i	image i
$sim(\mathrm{I}_i, \mathrm{I}_j)$	Jaccard similarity between I_i and I_j
K	length of Hamming codes
ht	threshold for Hamming distance
N	number of images in the dataset
n	number of local features in an image
w	number of visual words in the vocabulary
k	number of hash functions/tables
m	expected # common visual words between two random images
P_c	collision probability in the hash table
s	sketch size: number of hash keys in a sketch
ϵ	average Jaccard similarity between two random images

III. VISUAL HYPERLINKING

The fundamental problem underlying hyperlinking lies in mining frequent objects, such that links can be established by connecting common objects across images. Different from mining frequent images, the number of object hypotheses is several magnitudes large than images, which dramatically increases the computational cost. This section presents a bottom-up approach, starting from threading local features across images (Section III-A), followed by clustering the threads into objects for hyperlinking (Section III-B), and ending with a discussion of practical concerns such as scalability and robustness for online retrieval (Section III-C). Table I summarizes the key notations used in this section.

A. ToF Extraction

Thread of features (ToF) is defined as a set of consistent patches threaded across multiple images. In our implementation, each ToF is represented as a set composed of threaded images each of which is assigned a unique number. The consistency comes from visually similar appearance and spatially coherent neighborhood configuration among local features observed in different images. Basically, ToF serves as an elementary component that links visual objects in image or video collections. In principle, ToFs should be (1) compact to only link potential features from objects; (2) complete to cover as

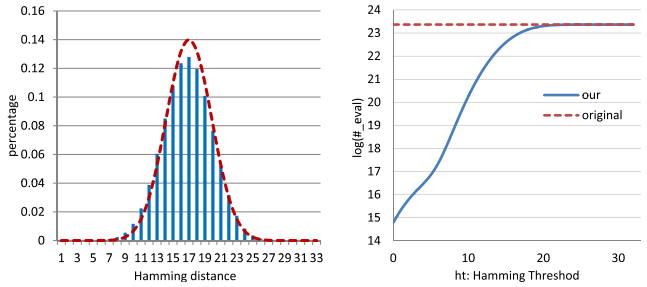


Fig. 2. Left: The distribution of Hamming distances among features extracted from 10k Flickr images. The curve $Bin(32, 0.5)$ is overlaid as red dashed line. Right: the number of evaluations (# *eval*) with respect to different Hamming signature threshold (ht). The red dashed line indicates the case without using Hamming code.

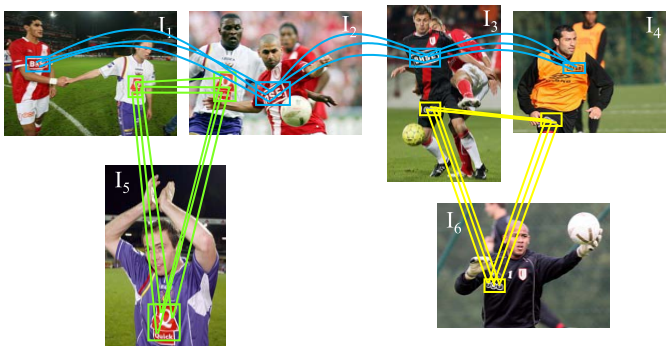


Fig. 3. Example ToFs on six images with three objects. Clustered ToFs are highlighted using the same color as follows: Blue: (I_1, I_2, I_3, I_4) - Base; Green: (I_1, I_2, I_5) - Quick; Yellow: (I_3, I_4, I_6) - TNT.

many objects as possible; (3) efficient to extract and thus be scalable on large datasets.

ToF extraction starts by quantizing SIFT features into visual words. Generally, when comparing two images, features quantized to the same visual words can be considered to be matched. Nevertheless, such matching is not necessarily robust for feature threading due to quantization error. Therefore, we also consider the neighboring points around each SIFT feature F to improve matching robustness. Let F be a central feature with l nearest neighboring features that can be considered to augment F as a small local patch. These neighboring features are selected within a small region centered at F, and with the similar scale of F. Therefore, a local patch is composed of a central feature F and a set of l neighboring features. Threading is performed by evaluating the similarities of patches across images. Basically, two patches are considered to be matched when they share some fraction of common visual words. Let w be the size of the visual vocabulary. Furthermore, consider a dataset with N images, each of which has an average of n local features. The number of similarity evaluations required by [25] for threading amounts to $(\frac{Nn}{w})^2 \times w = (Nn)^2/w$, which becomes computationally expensive for a large dataset.

To speed up this process, we embed each central feature in a patch with a short binary Hamming signature [42] for early pruning. The signature is computed by randomly projecting the SIFT features to a lower dimensional space and coding them into binary signatures. Threading is conducted by only evaluating only those patches that share the same visual word and where the Hamming distance between their signatures is smaller than a specified threshold ht. Consequently, the total number of evaluations is further reduced to:

$$(\frac{Nn}{w} \times \mathrm{CDF}(ht))^2 \times w = \frac{(Nn)^2}{w} \times \mathrm{CDF}^2(ht), \qquad (1)$$

where $\mathrm{CDF}(ht)$ is the Cumulative Distribution Function of the Hamming distance, which could be approximated as a Binomial distribution $Bin(K, 0.5)$ for a K-bits Hamming signature. We use $K = 32$-bits Hamming codes throughout this paper. Fig. 2 (left) plots the distribution of Hamming distances in 10k random Flickr images. As shown, the probability mass function of $Bin(32, 0.5)$ roughly fits the actual distribution. According to Eq. 1, threading involves only a fraction of $\mathrm{CDF}^2(ht)$ evaluations. For a commonly used threshold, e.g., $ht = 10{\sim}12$, this fraction is around $(\sum_{i=0}^{10\sim12} \mathrm{PMF}(i))^2 = 0.06\%{\sim}1.16\%$, where PMF stands for the probability mass function. Fig. 2 (right) plots the actual number of evaluations when mining on the 10k dataset.

ToF extraction can be efficiently implemented with an inverted-file index. Specifically, each quantized SIFT feature in an image is indexed in an inverted index together with its Hamming signature. In addition, the visual word IDs of the l neighboring features are also indexed along with each feature. ToFs are extracted by traversing the posting list of every visual word in the index. With the help of the Hamming signature, we can efficiently extract several ToFs from each posting list by traversing the inverted file structure. Note that with the inclusion of the l neighboring features, the inverted index built for hyperlinking cannot be kept in memory as in visual search. However, in the context of visual hyperlinking, due to the independence of the posting lists, we sequentially load and process each posting list separately at the time of ToF extraction.

B. Object Mining

Because a ToF links images that share consistent patches, we can also represent a ToF as a set of linked images. With this representation, grouping similar ToFs is equivalent to finding the co-occurred patches observed in different images. Therefore, the goal of mining here is to cluster the ToFs such that each cluster corresponds to a candidate frequent object shared among different images. We employ min-Hash (mH) [29], which is a randomized algorithm for efficient estimation of set similarity, for ToF clustering. The algorithm hashes the ToF, represented as a set of images, to a minimal index according to a random permutation. The collision probability between two sets approximates their Jaccard coefficient, presuming that mH uses a large number of hash functions. Furthermore, mH groups s number of hashing keys as a s-tuple called *sketch*. The probability that two ToFs T_1 and T_2 having at least one sketch collision is given by:

$$P_C(\mathrm{T}_1, \mathrm{T}_2) = 1 - (1 - sim(\mathrm{T}_1, \mathrm{T}_2)^s)^k. \qquad (2)$$

By mH, ideally the set of ToFs that collide in the hash tables ideally refers to the set of images sharing similar group of patches. Fig. 3 shows a toy example of six images with three common objects among them. Because each ToF is represented

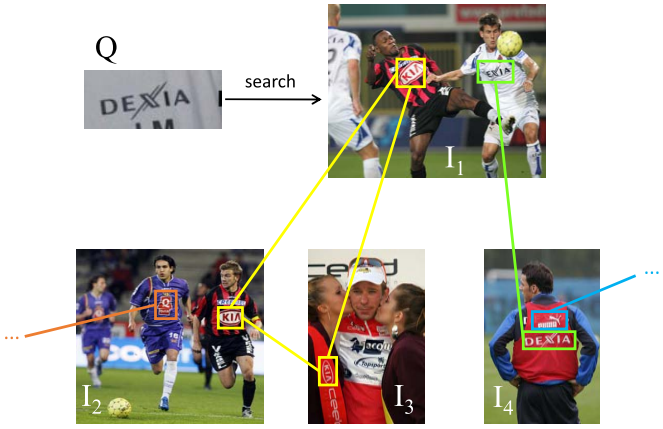


Fig. 4. Direct involving all hyperlinks associated with query retrieved image is risky. Although query Q retrieves I_1 as true response, I_2 and I_3, involved by the hyperlink in yellow, should not be considered in re-ranking.

as a set of image IDs,[1] it is easy to see that these ToFs can be readily grouped into clusters, each of which corresponds to an object by Jaccard similarity. In other words, the union set of images in a cluster tells the potential object holders that can be hyperlinked as sharing a common object. In practice, to avoid false links due to feature noises during the ToF extraction, the following heuristics to restrict the lengths of ToFs and the number of hyperlinks are adopted. First, a small value of the Hamming distance threshold in the range of $ht = 8 \sim 10$ (see Eq. 1) and a large number of common visual words, $t = 3 \sim 5$, are expected when threading patches into a ToF. Second, for a cluster of ToFs, only those images that are linked by most ($\geq$80% in our implementation) of the ToFs are considered for hyperlinking, which can be easily implemented with cluster voting. As false links between images can propagate noises resulting in adverse effects for applications such as search re-ranking, these two practical concerns are crucial to keep the noise level in a minimum level.

In our context, a hyperlink is defined as a link connecting instances across the images as voted by the clustered ToFs. It is worth noting that the bounding boxes of common objects could be proposed for images along the hyperlink, given that the ToF cluster actually captures the location of co-occurred patches. For small objects such as logos and signboards, the bounding boxes may only enclose only a small fraction of the images (Fig. 9). At the other extreme, for hyperlinks between two near-duplicate images, the bounding boxes may simply enclose the entire images. In practice, most bounding boxes only partially align with the actual object (e.g., the Presidential logo, and the Parking sign board in Fig. 9).

C. Discussion

One critical issue governing the success of hyperlinking for object retrieval is object scale. Generally speaking, the ability of hyperlinking smaller-scale objects is more valuable than hyperlinking large ones, because the latter (e.g., near-duplicate images, landmark buildings) are not always a problem for visual search. Min-Hash (mH) normally operates on the whole

[1]For example, the three threads in blue can all be represented as $\{I_1, I_2, I_3, I_4\}$

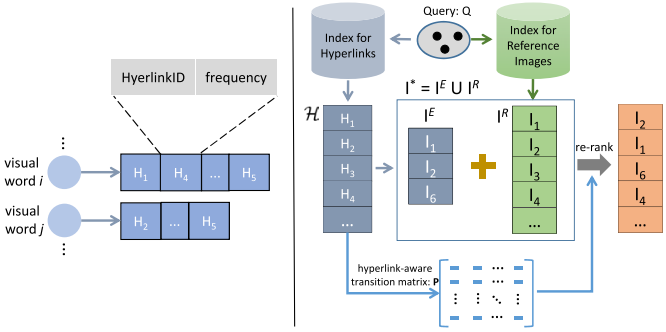


Fig. 5. Illustration for query-expanded re-ranking. Left: index structure of the hyperlinks. Right: re-ranking with the offline constructed hyperlinks.

image level [18]. To guarantee the success rate for mining small scale objects, mH needs a fairly large number of hash tables. In this subsection, we discuss the reason why the ToF (instead of the image itself) is adopted as the unit of hashing, and how small scale objects can still be mined even using fewer numbers of hashing tables.

Let's first contrast hashing using images and ToFs as units. Assume a dataset with N random images, each of which has n local features on average, and a visual vocabulary $\mathcal{V} = \{v_1, v_2, ..., v_w\}$ of size w for quantization. Furthermore, assume that local features in this dataset are quantized to each visual word with equal chance. For a pair of random images I_A and I_B, let $X_1, X_2, ..., X_w$ be a list of indicator random variables with

$$X_i = \begin{cases} 1, & \text{if } v_i \in I_A \ \& \ v_i \in I_B, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

That is, $X_i = 1$ only if both images have the i-th visual word. Since each X_i is identical and independent to each other, the expected number of common visual words m is given by:

$$m = \mathbf{E}[\sum_{i=1}^{w} X_i] = w \times \mathbf{E}[X_1] = w(1 - (\frac{w-1}{w})^n)^2. \tag{4}$$

Let $x = 1/w$. Then, the term $(\frac{w-1}{w})^n = (1-x)^n$ can be expanded with Taylor expansion near $x = 0$ as $1 - nx + O(x^2)$. This linear approximation for $m = n^2/w$ is already accurate enough, since x approaches 0 for a large vocabulary. Then the Jaccard similarity between a random pair of images can be written as:

$$\epsilon = \frac{|I_A \cap I_B|}{|I_A \cup I_B|} = \frac{m}{2n - m} \approx \frac{n}{2w - n}. \tag{5}$$

This ϵ is important since it estimates the average similarity for random image pairs, which can be used to threshold false positives. For each $\binom{N}{2}$ pairs of images, the number of image pairs found in k hash tables follows the Binomial distribution $Bin(k, \epsilon^s)$. As a result, the expected number of random image pairs mined from the whole dataset is:

$$RC = \binom{N}{2} \times k \times \epsilon^s. \tag{6}$$

Take Oxford105k used in [18] for example, where $N = 104,844$, $n = 2,805$, $w = 2^{17}$, $k = 512$, and $s = 3$. According to Eq. 6, $RC = 3.34 \times 10^6$, which roughly matches

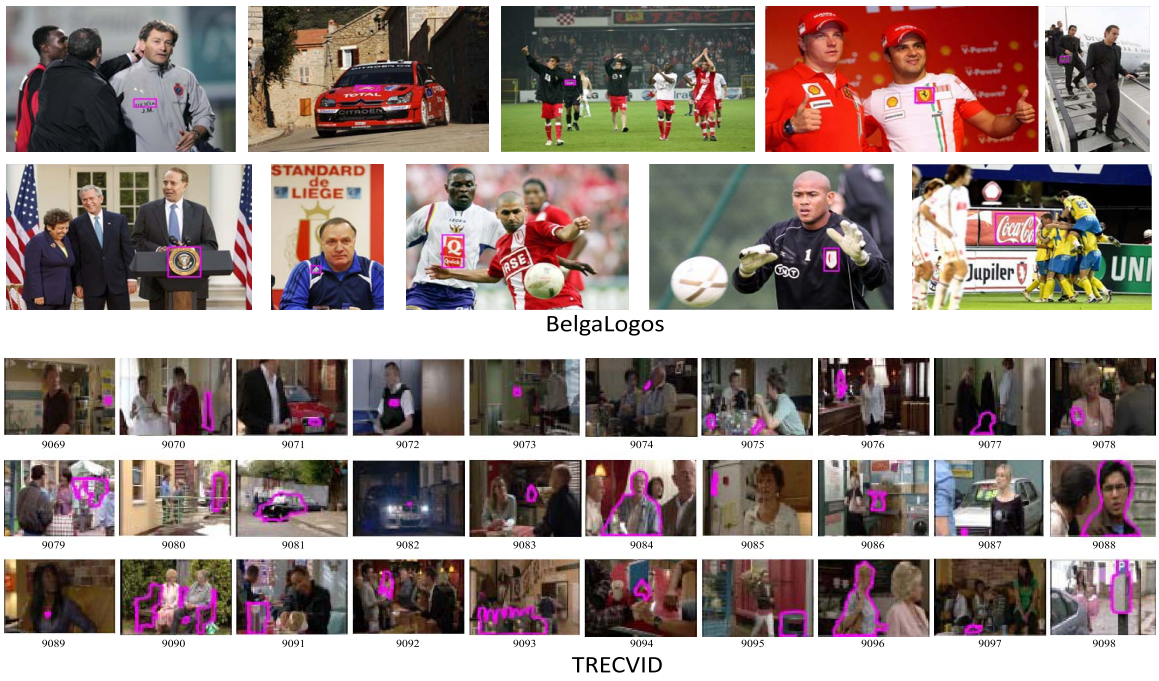


Fig. 6. Example query images from BelgaLogos (top) and TRECVID (bottom) dataset. The querying objects are outlined with magenta. Since most queries are in small size, this figure is best viewed in color.

the number reported in [18]: $38.4 \times N = 4.02 \times 10^6$, where both *random* and *true* collisions are counted. That is to say, in addition to true image pairs, more than 3 million random pairs are expected to be extracted from a 100k dataset. Moreover, because RC grows quadratically in N, the scalability of mining in large datasets is also an issue.

Consider a set of images that share a small object, where $m \ll n$ and ϵ goes to zero by Eq. 5. Thus, hashing images can hardly find the object. In contrast, hashing ToFs does not suffer from this problem, because the probability of ToF collision is independent from object scale. Furthermore, the ToF is relatively "clean" since any surrounding features not relevant to ToF should have been excluded during ToF extraction. Indeed, the only factor that matters for collision, indeed, is the similarity among the ToFs composing of an object. It is obvious that, due to this compact and clean representation, the similarities among ToFs composing an object will be much higher than the similarities among the images that share the object. According to Eq. 2, to ensure a high probability of collision $P_C > \alpha$, we need at least:

$$k > \log_{1-sim^s}(1-\alpha) \tag{7}$$

hash tables. In other words, far fewer k hash tables are needed with a slightly higher sim, since $\log_{1-sim^s}(1-\alpha)$ is quite sensitive around the common value $sim \approx 0.02$. This property is critically useful for object mining, since a large amount of computation can be avoided by using fewer hash tables.

IV. Hyperlink-Aware Re-Ranking

With visual hyperlinks, an image collection can be visualized as a graph with the images as nodes and the hyperlinks as edges. Algorithms such as Random Walk [15] can be applied to assign each image a static score indicating its representativeness. For example, an image containing objects that are also found in many other images in a collection is likely to receive a higher score. For online retrieval, these hyperlinks can also be leveraged for re-ranking an initial set of retrieved images using Random Walk. Specifically, the set of initial retrieved images, denoted as $\mathrm{I}^R = [\mathrm{I}_1, \mathrm{I}_2, ..., \mathrm{I}_K]$, can be expanded even for images (denoted as I^E) that may not have been initially retrieved but that are hyperlinked by the images in I^R. In this way, a graph composed of $\mathrm{I}^* = \mathrm{I}^R \cup \mathrm{I}^E$ can be established for re-ranking. Let $\mathbf{x}_0$ be the vector encoding the initial retrieval scores of images. Then Random Walk is conducted by

$$\mathbf{x}_{n+1}^T = \alpha \mathbf{x}_n^T \mathbf{P} + (1-\alpha)\mathbf{x}_0^T, \tag{8}$$

where $\mathbf{x}_n$ is the score vector after the n-th iteration, and $\mathbf{P}$ is a transition matrix where each element $p_{i,j}$ encodes the probability of traversing from image i to j through the hyperlinks between them. By the Power Method [43], Eq. 8 is guaranteed to converge after a finite number of iterations.

Nevertheless, this strategy is risky as the hyperlinks could include images or objects not relevant to the query, not to mention false hyperlinks that could introduce noises during re-ranking. There are cases where a retrieved image may have more than one hyperlink to other images, or a hyperlink associated with the retrieved image may not feature the query object. Fig. 4 shows an example where the query Q retrieves image I_1, while two other images I_2 and I_3 hyperlinked by I_1 are included for re-ranking even though that they are irrelevant to Q. In this chapter, we propose two approaches to address this problem from the spatial and visual perspectives, respectively. Here, the central problem here is to verify the relevancy between Q and the established hyperlinks. The first approach (Section IV-A) estimates the spatial extent of the object on reference images, and considers only hyperlinks

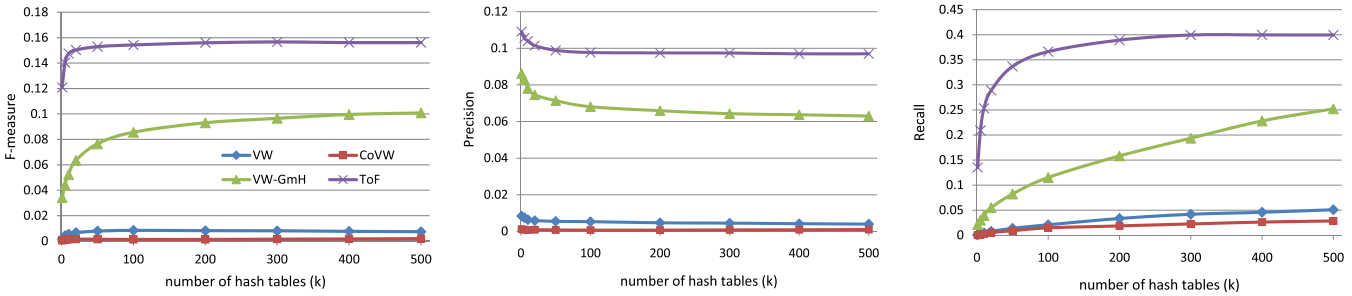


Fig. 7. Hyperlinking performance on Oxford105k dataset. Left: F-measure. Mid: Precision. Right: Recall.

that spatially align with the estimated area. The second approach (Section IV-B) directly considers hyperlinks where the underlying objects are visually similar to the query.

A. Localized Re-Ranking

Since this approach explicitly considers the spatial extent of the hyperlinks, we name this approach as localized re-ranking. For a reference image I_i in I^R, a hyperlink H associated with I_i can be spatially cast onto I_i by fitting a normal distribution $N(\mu_h, \delta_h^2)$, where μ_h and δ_h^2 indicate the mean and variance of the spatial locations of the ToFs compositing H, respectively. Similarly, the query Q can also be casted onto I_i by tracking the matching locations between Q and I_i. Another normal distribution $N(\mu_q, \delta_q^2)$ can be fitted to the matched points on I_i. Take Fig. 4 as an example, the hyperlink in yellow is spatially casted to I_1 as shown by the yellow box of I_1, and Q is mapped onto I_1 as shown by the green box of I_1. Using this strategy, we spatially verify the relevancy between H and Q with a *Z-test* between the two distributions $N(\mu_h, \delta_h^2)$ and $N(\mu_q, \delta_q^2)$:

$$z = \frac{\mu_q - \mu_h}{\sqrt{\delta_q^2 + \delta_h^2}}. \tag{9}$$

In our implementation, z values less than 2.33 are accepted as relevant, which corresponds to a 1% significance level.

The main challenge of this strategy is to precisely estimate $N(\mu_h, \delta_h^2)$ and $N(\mu_q, \delta_q^2)$. However, in practice, precisely locating an object by fitting a 2D normal distribution using only a few matching points is difficult. Typically, an object has fewer than 20 such points and they are sparsely distributed for an object. Moreover, erroneous hyperlinks introduced by false positive images in I^R will result in adverse effect while re-ranking.

B. Query-Expanded Hyperlinking

While localized re-ranking ideally excludes irrelevant hyperlinks via spatial overlap checking, false matches between the query Q and I^R can still introduce irrelevant hyperlinks. An alternative and indeed more feasible way is by retrieving from the mined hyperlinks whose underlying common objects are visually similar to Q, and including the involved images for re-ranking. This strategy is more reliable, since it directly considers the visual similarity between the query and hyperlinks.

Fig. 5 shows the detailed framework for re-ranking based on query-expanded hyperlinking. Two inverted indexes are constructed for fast retrieval of images and hyperlinks. The first index, for reference images, refers to the data structure where each visual word points to a posting list of images that contain the visual word. This structure facilitates fast retrieval of images that have at least one common visual word in common with Q. The second index keeps offline-mined hyperlinks, where each hyperlink is represented as a BoW vector depicting the underlying common object. The detailed structured for this index is shown at the left side of Fig. 5, where each visual word points to a posting list of hyperlinks that include the visual word. Each entry in the index includes the hyperlink ID and frequency of the visual word. During online retrieval, the sets of images I^R and hyperlinks $\mathcal{H}$ that are relevant to the query can be rapidly retrieved by traversing only a few posting lists in both inverted indexes. Denoting I^E as the set of images connected by hyperlinks in $\mathcal{H}$, Random Walk (Eq. 8) can be conducted on the image set $\mathrm{I}^* = \mathrm{I}^R \cup \mathrm{I}^E$. It is important to note that the transition probability matrix $\mathbf{P}$ can be constructed based on $\mathcal{H}$, i.e., $\mathbf{P}_{i,j}$ equals the number of hyperlinks connecting the images I_i and I_j. It is also worth noting that $\mathbf{P}$ could be very sparse if $|\mathrm{I}^E| \ll |\mathrm{I}^R|$, which is usually true since hyperlinks only connect only images that share a set of ToFs. As a result, re-ranking can be extremely efficient in practice.

V. Experiments

In this section, we conduct experiments for evaluating the performances of visual hyperlinking (Section V-B) and search re-ranking (Section V-C). As there is no benchmark available for visual hyperlinking, we first fully annotate a dataset (Oxford5k, Section V-A) to evaluate visual hyperlinking.

A. Datasets and Implementation Details

Three datasets (Oxford, BelgaLogos, and TRECVID) are used for the experiments. The Oxford5k dataset, which contains our hand-annotated hyperlinks for more than 350 objects, is used for the hyperlinking evaluation. The other two datasets, which include objects queries in the image and video domains, respectively, are used for re-ranking evaluation.

1) Oxford: The Oxford5k dataset has 5,062 Flickr images with 11 landmarks manually labeled as the ground-truth. Since Oxford5k contains many more common objects than the 11 officially labeled objects, we further annotated the dataset, resulting in 364 clusters of objects involving 2,369 images.

TABLE II
QUERY TOPICS FOR TRECVID'13 & 14

TV13		TV14	
ID	*Topic Name*	*ID*	*Topic Name*
9069	circular no-smoking logo	9099	a checkerboard band on pol-cap
9070	small red obelisk	9101	a Primus washing machine
9071	an Audi logo	9102	this large vase
9072	a Police logo	9103	a ketchup container
9073	this cat face	9104	this woman
9074	a cigarette	9105	this dog
9075	a SKOE can	9106	a London Underground logo
9076	this bust of Queen	9107	this Walford Station entrance
9077	this dog	9108	these 2 ceramic heads
9078	a JENKINS logo	9109	a Mercedes star logo
9079	this CD stand in the market	9110	these etched glass doors
9080	this public phone booth	9111	this dartboard
9081	a black taxi	9112	this HOLMES lager logo
9082	a BMW logo	9114	a red public mailbox
9083	a chrome and glass cafeteria	9115	this man
9084	this man	9116	this man
9085	this David magnet	9118	a Ford Mustang grill logo
9086	these scales	9119	this man
9087	a VW logo	9120	a wooden park bench
9088	Tamwar	9121	a Royal Mail red vest
9089	this pendant	9122	this round watch
9090	this wooden bench	9123	a white plastic kettle
9091	a Kathy's menu	9124	this woman
9092	this man	9125	this wheelchair with armrests
9093	these turnstiles	9126	a Peugeot logo
9094	a ketchup dispenser	9127	this bust of Queen Victoria
9095	a green public trash can	9128	this F pendant
9096	Aunt Sal		
9097	these checkerboard spheres		
9098	a P (parking automat) sign		

The full set of annotations has been made publicly available.[2] For scalability test, the Oxford5k dataset is then mixed together with the Oxford100k dataset with 99,782 Flickr images. We name this combined dataset as Oxford105k. Note that the additional Oxford100k images are not annotated; instead, they are treated as distracting images during the evaluation.

BelgaLogos is composed of 10,000 images with 26 query logos [44]. The dataset is regarded as a standard benchmark for logo retrieval in natural images. Fig. 6 (top) shows several example queries.

TRECVID INS is an instance search (INS) [37] dataset aimed at searching for objects/persons/locations from large video collection. The dataset is composed of 470k video clips (amount to 640k keyframes) extracted from the BBC TV series "EastEnder" - Programme material © BBC. We conduct experiments using 57 query topics released by TRECVID in years of 2013 (denoted as TV13) and 2014 (TV14), as listed in Table II. Each query topic is accompanied by several query images, where each is associated with a mask specifying the object instance being queried. Fig. 6 (bottom) shows example queries from TV13. In general, this dataset is challenging due to the large visual variations among different instances of the same objects.

2) Implementation Details: A hierarchical vocabulary [12] with one million leaf nodes ($\omega = 10^6$) is trained on 100k randomly crawled Flickr images. Two patches of different images are hyperlinked if at least three ($t = 3$) out of ten ($l = 10$) neighbor features share the same visual word. The number of Hamming bits (K) is set to 32, and the threshold for Hamming distance (ht) is 10. For min-Hash, the number of hash tables (k) is 512, with sketch size (s) being set to 3. For re-ranking, the parameter values are $\alpha = 0.3$ in Eq. 8 and $z > 2.33$ in Eq. 9.

[2][Online] http://vireo.cs.cityu.edu.hk/gt_clusters.oxford5k

Fig. 8. Example mining results from Oxford105k dataset, using our method ToF. Top 3 rows: example objects in ground-truth set. Images with green-solid border are in the "junk" set of Oxford dataset, while red-dashed border indicates false positives. Bottom 4 rows: example objects outside ground-truth set. The last row is a false cluster.

TABLE III
THE PERFORMANCE (F-MEASURE) ON THE SUBSET OF *Small* OBJECTS IN OXFORD DATASET. A MID-LEVEL VALUE OF $k = 300$ (1∼500 EVALUATED IN FIG. 7) IS ADOPTED FOR ALL METHODS

	VW	CoVW	VW-GmH	ToF
Oxford5k	0.0006	0.1370	0.1670	0.1836
Oxford105k	0.0006	0.0005	0.0347	0.0994

B. Hyperlinking

We compare three types of features for hyperlinking: ToF (our method), VW (visual words) and CoVW (co-occurring VW [27]). For VW, each image is represented as a bag-of-words vector, while for CoVW each visual word is represented as bag-of-images. All these features (ToF, VW, and CoVW) use min-Hash (mH) [29] for feature hashing. Additionally, we also test the Geometric min-Hash (GmH) on VW (denoted as VW-GmH) because of its improved performance as reported in [28]. In the experiment, the sketch size is set to $s = 3$ for all methods. The performance is evaluated based on the *F-measure*, which is defined as the harmonic mean of precision and recall. Concretely, precision is defined as the fraction of correct pairwise hyperlinks among all the established hyperlinks, and recall is defined as the fraction of ground-truth hyperlinks correctly established.

Fig 7 shows the performances of the various approaches on the Oxford105k dataset. Note that ToF performs consistently better in terms of precision and recall across all the settings. Even with only 50 hash tables, ToF outperforms VW with 500 hash tables. When mining small-sized objects, larger number of hash tables is expected for VW. However, as VW

Fig. 9. Example hyperlinks mined from the BelgaLogos (left) and TRECVID (bottom) dataset. Three images in a row correspond to a hyperlink outlined with the estimated bounding boxes. The threads of visual words across rectangles are colored differently.

TABLE IV

THE RUNNING TIME COMPARISON FOR HYPERLINKING ON OXFORD DATASET. FOR TOF, THE TOTAL TIME (C) IS DECOMPOSED TO FEATURE THREADING (A) AND HASHING (B) AS: A + B = C. m FOR MINUTE

	VW	CoVW	VW-GmH	ToF
Oxford105k	9.5 m	8.0 m	9.6 m	6.3 + 1.3 = 7.6 m

TABLE V

PERFORMANCE (mAP) COMPARISON FOR DIFFERENT RETRIEVAL METHODS

	baseline	Hsu	SPAUG	HL-Loc	HL-QryEx
BelgaLogos	0.2772	0.2694	0.2605	0.2697	0.3052
TV13	0.1976	0.1801	0.1646	0.1855	0.2042
TV14	0.1966	0.1871	0.1750	0.1848	0.2060

is vulnerable to random collision especially on large dataset, increasing the number of hash tables only results in marginal improvement. The GmH version of VW reduces the effect of random collision by careful selection of secondary hash keys. While large improvement is attained, its performance is still lower than that of ToF. CoVW basically shows the worst performance due to noisy bag-of-images representation. The approach is particularly sensitive to the large numbers of distracting images in the dataset, which introduces false images into the bag-of-images representation and subsequently results in erroneous similarity measures of words during hashing.

1) Performance on Small Objects: Considering the importance of hyperlinking small objects for object retrieval, we further investigate the performance of small objects hyperlinking on the Oxford dataset. To separate small objects from large ones, we first calculate the average Jaccard similarity for pairwise images in each ground-truth cluster, and then cut the object clusters into halves based on those similarities. Object clusters with similarities below 50% are regarded as small. Table III presents the detailed performance on the small subset. On the Oxford5k dataset, GmH version of VW shows better performance than mH. Although CoVW attains reasonably good result on the Oxford5k, its performance does not scale up as the number of images increases, and its F1 score drops significantly on the large Oxford105k dataset. Overall, ToF copes better with the size of the dataset and exhibits the best performances when linking small objects. Figure 8 shows examples of the objects hyperlinked by ToF on the Oxford105k dataset. In addition to the objects labeled in the ground-truth set (the top three rows), ToF is able to mine unlabeled objects from the Oxford100k dataset (see the bottom four rows). As observed, some of the patterns labeled as "junk" (less than 25% of the region is visible) in [8] are extracted by ToF, demonstrating its ability to link small objects. A falsely linked object due to a repeated pattern is also shown in the last row of Fig. 8.

2) Speed: Table IV summarizes the running time. The experiments are all conducted on an 8-core machine with 2.67GHz CPU and 16GB memory. In addition to feature hashing like the other methods, ToF requires an additional step of feature threading step (Section III-A). Despite this, ToF is still more efficient than others because the average length of ToF is much shorter than that of the bag-of-words or bag-of-images approaches used by VW and CoVW, respectively. Essentially, the hashing time is mainly affected by two factors: the total time is linear in the number of sets; and computing the hash key for a set is linear in the size of the set. After feature threading, both these factors are reduced significantly due to the compactness of ToF. Note that in this experiment, for fair comparison, all methods adopt the same number ($k = 100$) of hash tables. It is also worth noting that ToF usually requires much fewer hash tables in practice, which could further reduce the running time for real-world applications.

C. Re-Ranking With Hyperlinks

We evaluate our hyperlink-aware re-ranking approach on the BelgaLogos and TRECVID (TV13 and TV14) datasets. As shown in Fig. 6, both datasets include abundant small objects, and thus suffer from the over-norm problem. BelgaLogos uses small logos as queries, which appear in diverse backgrounds (e.g., car, wall, bag, football field). TRECVID

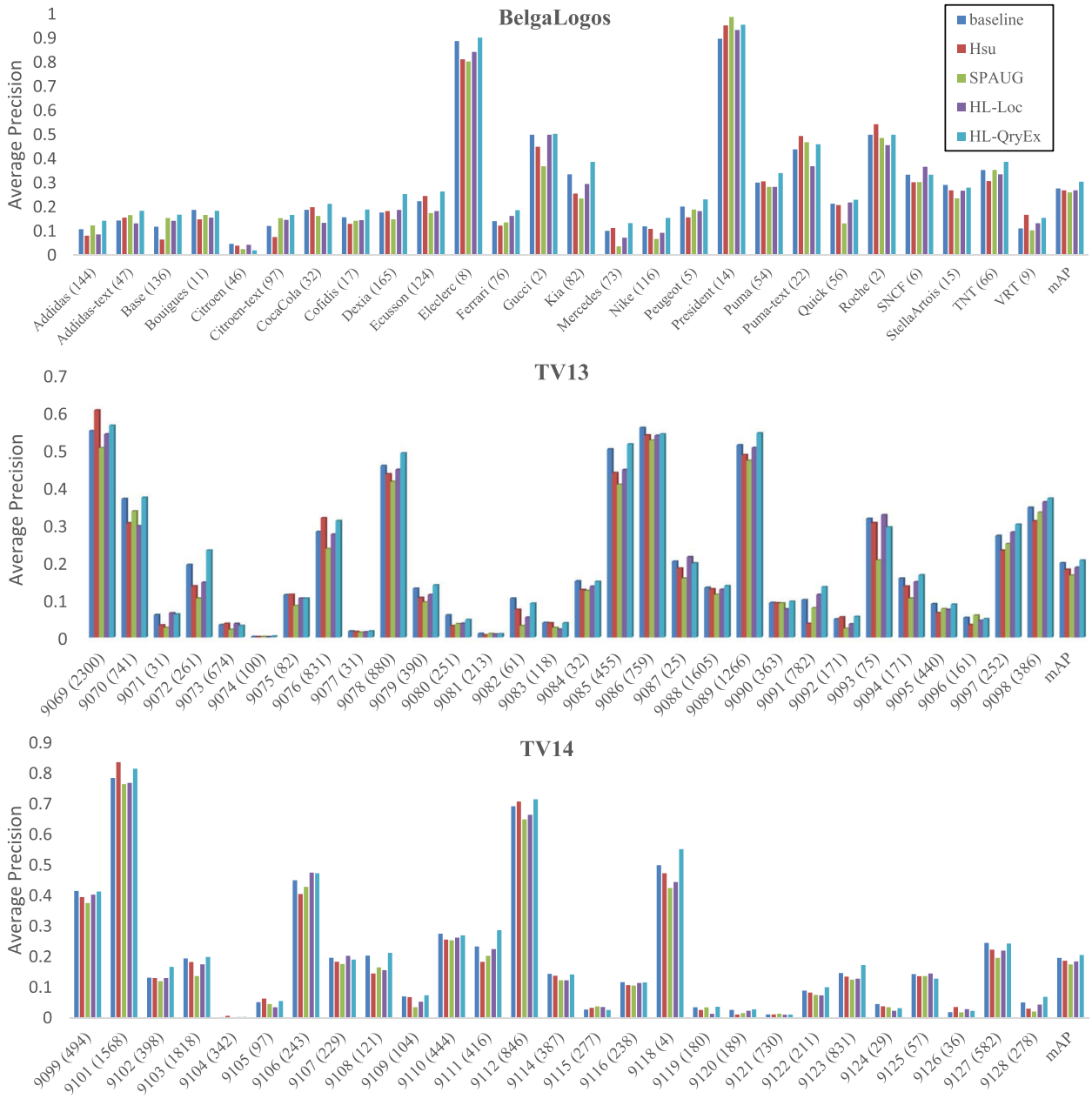


Fig. 10. Performance comparison for different spatial verification techniques. Top: BelgaLogos; Mid: TV13; Bottom: TV14. Numbers in parentheses indicate the number of ground-truth results for each query.

is more challenging in terms of visual variations as a result of non-rigid transformation, visual quality and view-point changes.

1) Compared Approaches: For fair comparisons, all methods compared in this section are based on the same baseline [45], which integrates the BoW retrieval model [7], Hamming embedding [42], multiple assignments [11], and topological spatial verification [36]. We include the following methods for comparison: *baseline* [45], *Hsu* [15]: Random Walk re-ranking I^R with pairwise image-level similarities; *SPAUG* [2]: extending [13] by augmenting each reference image with spatially consistent features on similar images in the dataset; and our two versions of re-ranking with hyperlinking (*HL-Loc*: localized re-ranking in Section IV-A, *HL-QryEx*: query-expanded re-ranking in Section IV-B). In our implementation, we set $\alpha = 0.3$, learned from a separate validation set. Note that most of these methods suffer from the over-norm problem, since the retrieval/re-ranking is conducted at the image level. In our experiment, mean Average Precision (mAP) is adopted for evaluating the retrieval performances.

2) Performance Comparison: Table V summarizes the overall performance of various re-ranking methods on the BelgaLogos and TRECVID datasets, and Fig. 10 details the performances for each query. In general, the baseline performs reasonably well on all datasets. Neither Hsu nor SPAUG

Fig. 11. Example ranks by different methods in the context of object retrieval. Each row corresponds to a query, a true reference image and the rank positions of the reference image.

TABLE VI

PERFORMANCE (mAP) ON TRECVID BY SPLITTING THE 57 QUERIES TO "NON-OVERLAP" AND "OVERLAP" WITH THE ESTABLISHED HYPERLINK. NUMBERS IN PARENTHESES INDICATE THE NUMBER OF QUERIES IN EACH CATEGORY

	baseline	HL-QryEx
non-overlap (35)	0.1070	0.1100
overlap (22)	0.3405	0.3563

improves the baseline, since both of them cannot cope with small objects. For Hsu, the transition probability matrix encodes only image-level similarity, which cannot properly handle images that share a small logo/object. Similarly, for SPAUG, augmenting the entire reference image is far less helpful in retrieving small objects. SPAUG is intended to augment spatially verified features from other images, and thus the image representation becomes more robust. However, SPAUG can only offline augment images, and thus the augmentation process is independent of the query. Most of the time, the augmented features are not necessarily derived from objects similar to the queries, making the results of SPAUG not adaptive to a given query. Although SPAUG can exclude spatially inconsistent features from similar images into feature augmentation, it is still difficult for it to locate the right object to augment. HL-Loc suffers from a large number of falsely included hyperlinks, which degrades its performance in re-ranking. HL-QryEx outperforms other methods by properly using the hyperlinks for re-ranking. On the BelgaLogos dataset, 19 out of 26 queries are improved over the baseline. The reasons for this promising result are two folds. First, our offline-constructed hyperlinks connect sub-image level objects. Poorly ranked small objects can be promoted to the front of the ranking due to these subimage-level hyperlinks. Second, false linking that confuses the re-ranking process is effectively excluded from re-ranking. Using tight parameters (e.g., ht, t) for hyperlinking and query-expanded re-ranking, our re-ranking approach focuses on the queried object during re-ranking. Fig. 11 shows examples of the rankings for two relevant images retrieved by different methods. Due to the over-norm problem, the baseline results in a rather poor ranking even though the objects on the reference images are clear and rigid. On the contrary, HL-QryEx and HL-Loc bypass the over-norm problem by directly hyperlinking small objects via mining.

Although the overall improvement introduced by our method on TRECVID is not as large as that on BelgaLogos, the re-ranking approach still manages to produce systematically better results. Hyperlinking objects in the TRECVID dataset is much more difficult, due to the large visual variations among instances of the same object. Therefore, our "hyperlink-and-rerank" strategy results in a less significant improvement on the TRECVID dataset.

To show that the performance gain on TRECVID dataset is not by chance, we conduct a significance test (paired-sample t-test) with the null hypothesis $\mathcal{H}_0$: there is no performance difference between the baseline and HL-QryEx. Upon completion of the test, $\mathcal{H}_0$ is rejected at the 0.05 significance level.

For our methods, the performance is closely related to the quality of the hyperlinks. Since we adopt the SIFT feature, rigid and texture-rich objects (e.g., Dexia, President, 9111-this dartboard) are more likely to generate high-quality hyperlinks. On the other hand, non-planar and non-rigid objects (e.g., 9077-this dog, 9096-Aunt Sal) involve no links or false links. Therefore, queries involving poor links are either not affected or degraded in performance. Note that, our mining technique is also compatible with other features (e.g., MAC [46]) that work better on non-rigid objects when discovering semantically similar objects.

The performance can be explained by inspecting the overlap between the established hyperlinks and the querying objects. For challenging queries, such as 9074-a cigarette and 9110-these etched glass doors, no hyperlinks are established for any instances of the queried object. We split the queries in TV13 and TV14 into "non-overlap" and "overlap" based on the cardinality of the hyperlinks $\|\mathcal{H}\|$ retrieved by a query. A query is considered as "overlap" if $\|\mathcal{H}\| \geq 30$. Table VI summarizes the performance after this partition. As expected, the improvement is more significant on the "overlap" subset (a relative improvement of 4.6% on overlap, and 2.8% on non-overlap).

By inspecting each individual query, we find that the performance is somewhat related to the number of relevant results in the ground-truth set. For example, the Pearson's coefficient between the number of ground-truth and relative improvement (HL-QryEx over baseline) is 0.59 on BelgaLogos. On one hand, it is easier to hyperlink with more instances of the object in a dataset, because there are greater chances for frequent objects (e.g., Adidas, Nike, 9069-no smoking logo, 9101-Primus washing machine) to be hyperlinked. On the other hand, the re-ranking process can also benefit from a large number of relevant results in the ranklist.

While encouraging, the degree of improvement for objects in larger size is limited by our approach. This is not surprise as large objects suffer less from over-norm problem than small objects. On Oxford5k dataset with 55 image queries, which contain mostly large objects, the performances of HL-QryEx and baseline are 0.5426 and 0.5406, respectively. As baseline already manages to retrieve most relevant images, further leveraging hyperlinks only leads to little improvement. In short, the degree of improvement by our approach

TABLE VII

RUNNING TIME PER QUERY (IN MILLISECONDS) BY DIFFERENT RETRIEVAL METHODS. TOP-500 IMAGES ARE CONSIDERED FOR RE-RANKING. TIME IN PARENTHESES: (RETRIEVAL/RE-RANK TIME)

	baseline	Hsu	SPAUG	HL-Loc	HL-QryEx
BelgaLogos	11 (11/-)	825 (11/814)	48 (48/-)	186 (11/175)	183 (11/172)
TRECVID	626 (626/-)	1431 (626/805)	1865 (1865/-)	839 (626/213)	824 (626/198)

is generally more apparent for retrieving small than large objects.

3) Speed: Since online response time is critical for real applications, we further compare the efficiency for various retrieval methods. As shown in Table VII, both our hyperlinking based methods (HL-Loc and HL-QryEx) slightly increase the running time due to the sparsity of the hyperlink-aware transition matrix. In contrast, Hsu is much slower in re-ranking due to dense matrix computation, and SPAUG does not involve a re-ranking step, but is slow when retrieving images with augmented features, not to mention time for constructing the matching graph.

VI. CONCLUSION

This paper presented a framework for object retrieval via visual hyperlinking. In this framework, a network of subimage-level hyperlinks is first established via the proposed object mining technique, which uses ToFs for efficient hyperlinking. Moreover, we investigated the hyperlink-aware re-ranking algorithms to enable fast Random Walk through object-level connections. We quantitatively evaluated both hyperlinking and retrieval on datasets with abundant small objects, and demonstrated the effectiveness of proposed methods in handling small objects. Based on our study, the over-norm problem is bypassed via the proposed "hyperlink and re-rank" framework. More importantly, our framework introduces only a small computational overhead to online retrieval.

Our approach currently suffers from a number of limitations. First, spatial regularity through geometric verification is not considered during ToF generation, which may result in spatially inconsistent results. As other approaches, large object variations, such as due to non-rigid deformation and viewpoint change, will result in performance degradation as noted in the experiments. Future work includes the incorporation of MAC [46] and objectness properties [3] for more robust generation of ToF. Second, this paper assumes static dataset and hence dynamic generation of hyperlinks for newly added images is not considered. Extending current work for dynamic dataset requires incremental update of hash tables and hyperlink index for continuous mining of objects and establishment of new hyperlinks, which are not trivial issues and worth further investigation.

REFERENCES

[1] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. van Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 777–784.

[2] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. CVPR*, 2012, pp. 2911–2918.

[3] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 73–80.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[5] Y. Jia *et al.* (2014). "Caffe: Convolutional architecture for fast feature embedding." [Online]. Available: http://arxiv.org/abs/1408.5093

[6] J. Tang, L. Jin, Z. Li, and S. Gao, "RGB-D object recognition via incorporating latent data structure and prior knowledge," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1899–1908, Nov. 2015.

[7] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. ICCV*, 2003, pp. 1470–1477.

[8] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. CVPR*, 2007, pp. 1–8.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[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 *Proc. ICCV*, Oct. 2007, pp. 1–8.

[11] J. Philbin, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. CVPR*, 2008, pp. 1–8.

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

[13] P. Turcot and D. G. Lowe, "Better matching with fewer features: The selection of useful features in large database recognition problems," in *Proc. ICCV*, 2009, pp. 2109–2116.

[14] K. Pearson, "The problem of the random walk," *Nature*, vol. 72, no. 1865, p. 294, 1905.

[15] W. H. Hsu, L. S. Kennedy, and S.-F. Chang, "Video search reranking through random walk over document-level context graph," in *Proc. ACM Multimedia*, 2007, pp. 971–980.

[16] W. Zhang, H. Li, C.-W. Ngo, and S. F. Chang, "Scalable visual instance mining with threads of features," in *Proc. 22nd ACM Multimedia*, 2014, pp. 297–306.

[17] J. Philbin and A. Zisserman, "Object mining using a matching graph on very large image collections," in *Proc. 6th ICVGIP*, 2008, pp. 738–745.

[18] O. Chum and J. Matas, "Large-scale discovery of spatially related images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 371–377, Feb. 2010.

[19] H. Liu and S. Yan, "Common visual pattern discovery via spatially coherent correspondences," in *Proc. CVPR*, 2010, pp. 1609–1616.

[20] H.-K. Tan and C.-W. Ngo, "Localized matching using Earth Mover's distance towards discovery of common patterns from small image samples," *Image Vis. Comput.*, vol. 27, no. 10, pp. 1470–1483, Sep. 2009.

[21] J. Yuan and Y. Wu, "Spatial random partition for common visual pattern discovery," in *Proc. ICCV*, 2007, pp. 1–8 .

[22] Y. Cheng, H. Fu, X. Wei, J. Xiao, and X. Cao, "Depth enhanced saliency detection method," in *Proc. Int. Conf. Internet Multimedia Comput. Service*, 2014, pp. 23:23–23:27.

[23] J. Chen, H. Zhao, Y. Han, and X. Cao, "Visual saliency detection based on photographic composition," in *Proc. 5th Int. Conf. Internet Multimedia Comput. Service*, 2013, pp. 13–16.

[24] T. Quack, V. Ferrari, and L. V. Gool, "Video mining with frequent itemset configurations," in *Proc. 5th CIVR*, 2006, pp. 360–369.

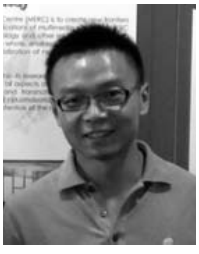
[25] J. Sivic and A. Zisserman, "Video data mining using configurations of viewpoint invariant regions," in *Proc. CVPR*, 2004, pp. 488–495.

[26] P. Letessier, O. Buisson, and A. Joly, "Scalable mining of small visual objects," in *Proc. 20th ACM Multimedia*, 2012, pp. 599–608.

[27] G. F. Pineda, H. Koga, and T. Watanabe, "Scalable object discovery: A hash-based approach to clustering co-occurring visual words," *IEICE Trans. Inf. Syst.*, vol. E94-D, no. 10, pp. 2024–2035, Oct. 2011.

[28] O. Chum, M. Perd'och, and J. Matas, "Geometric min-hashing: Finding a (thick) needle in a haystack," in *Proc. CVPR*, 2009, pp. 17–24.

[29] A. Broder, "On the resemblance and containment of documents," in *Proc. SEQUENCES*, 1997, p. 21.
[30] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jun. 2014, pp. 512–519.
[31] J. Wan *et al.*, "Deep learning for content-based image retrieval: A comprehensive study," in *Proc. 22nd ACM Multimedia*, 2014, pp. 157–166.
[32] J. Tang, Z. Li, M. Wang, and R. Zhao, "Neighborhood discriminant hashing for large-scale image retrieval," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2827–2840, Sep. 2015.
[33] H. Jégou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *Proc. CVPR*, 2009, pp. 1169–1176.
[34] C.-Z. Zhu and S. Satoh, "Large vocabulary quantization for searching instances from videos," in *Proc. 2nd ICMR*, 2012, Art. no. 52.
[35] W.-L. Zhao, X. Wu, and C.-W. Ngo, "On the annotation of Web videos by efficient near-duplicate search," *Trans. Multimedia*, vol. 12, no. 5, pp. 448–461, Aug. 2010.
[36] W. Zhang and C.-W. Ngo, "Searching visual instances with topology checking and context modeling," in *Proc. 3rd ACM Int. Conf. Multimedia Retr.*, 2013, pp. 57–64.
[37] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid," in *Proc. MIR*, 2006, pp. 321–330.
[38] R. Tao, E. Gavves, C. G. M. Snoek, and A. W. M. Smeulders, "Locality in generic instance search from one example," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2099–2106.
[39] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.
[40] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, "Total recall II: Query expansion revisited," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 889–896.
[41] M. Eskevich, R. Aly, D. Racca, R. Ordelman, S. Chen, and G. J. F. Jones, "The search and hyperlinking task at mediaeval 2014," in *Proc. MediaEval Workshop*, 2014, pp. 1–2.
[42] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *Int. J. Comput. Vis.*, vol. 87, no. 3, pp. 316–336, May 2010.
[43] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the Web," in *Proc. World Wide Web Conf.*, 1998, pp. 161–172.
[44] A. Joly and O. Buisson, "Logo retrieval with a contrario visual query expansion," in *Proc. 17th ACM Multimedia*, 2009, pp. 581–584.
[45] W. Zhang and C.-W. Ngo, "Topological spatial verification for instance search," *IEEE Trans. Multimedia*, vol. 17, no. 8, pp. 1236–1247, Aug. 2015.
[46] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–8.



Wei Zhang received the B.Eng. and M.Eng. degrees from Tianjin University, Tianjin, China, in 2008 and 2010, respectively. He received the Ph.D. degree from the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong, in 2015. Before joining Chinese Academy of Sciences, he was a Visiting Scholar in the DVMM Group, Columbia University, New York, NY, USA, in 2014. He is currently an Assistant Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His current research interests include large-scale visual instance search and mining, multimedia, and digital forensic analysis.

Chong-Wah Ngo received the M.Sc. and B.Sc. degrees in computer engineering from the Nanyang Technological University of Singapore, Singapore, and the Ph.D. degree in computer science from The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong.

He was previously a Post-Doctoral Scholar with the Beckman Institute, University of Illinois in Urbana-Champaign, Champaign, IL, USA. He was also a Visiting Researcher with Microsoft Research Asia, Beijing, China. He is currently a Professor with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. His current research interests include large-scale multimedia information retrieval, video computing, multimedia mining, and visualization.

Dr. Ngo was an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA (from 2011to 2014). He was a Conference Co-Chair of the ACM International Conference on Multimedia Retrieval 2015 and the Pacific Rim Conference on Multimedia 2014. He also served as a Program Co-Chair of ACM Multimedia Modeling 2012 and ICMR 2012. He was the Chairman of the Hong Kong Chapter of ACM from 2008 to 2009.

Xiaochun Cao received the B.E. and M.E. degrees in computer science from Beihang University, Beijing, China, and the Ph.D. degree in computer science from the University of Central Florida, Orlando, FL, USA.

After graduation, he spent about three years at Object Video Inc. as a Research Scientist. From 2008 to 2012, he was a Professor with Tianjin University, Tianjin, China. He has been a Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, since 2012. He has authored and co-authored over 120 journal and conference papers.

Prof. Cao is a fellow of the IET. He is on the Editorial Board of the IEEE TRANSACTIONS OF IMAGE PROCESSING. His dissertation was nominated for the University of Central Floridas university-level Outstanding Dissertation Award. In 2004 and 2010, he was a recipient of the Piero Zamperoni Best Student Paper Award at the International Conference on Pattern Recognition.