Scalable Triangulation-based Logo Recognition

Yannis Kalantidis
National Technical University of Athens
Zografou, Greece
ykalant@image.ntua.gr

Lluis Garcia Pueyo
Yahoo! Research
Santa Clara, CA, USA
lluis@yahoo-inc.com

Michele Trevisiol
Yahoo! Research
Barcelona, Spain
trevi@yahoo-inc.com

Roelof van Zwol
Yahoo! Research
Santa Clara, CA, USA
roelof@yahoo-inc.com

Yannis Avrithis
National Technical University of Athens
Zografou, Greece
iavr@image.ntua.gr

ABSTRACT

We propose a scalable logo recognition approach that extends the common bag-of-words model and incorporates local geometry in the indexing process. Given a query image and a large logo database, the goal is to recognize the logo contained in the query, if any. We locally group features in triples using multi-scale Delaunay triangulation and represent triangles by signatures capturing both visual appearance and local geometry. Each class is represented by the union of such signatures over all instances in the class. We see large scale recognition as a sub-linear search problem where signatures of the query image are looked up in an inverted index structure of the class models. We evaluate our approach on a large-scale logo recognition dataset with more than four thousand classes.

1. INTRODUCTION

Logo or trademark recognition has been a well-studied subject for decades since it arises in many practical scenarios of modern marketing, advertising and trademark registration. Most successful approaches deal with recognition from sketches, images or video taken in an uncluttered background. This includes recognition and matching of clear logos on white background and television station logo recognition from videos. In the later case, prior information can be utilized, *e.g.* information regarding the logo position and size [8], [15], or the temporal correlation between frames [28].

When logos appear in *natural scenes* though, they are much harder to detect and can vary in size from *e.g.* 20×20 pixels on a footballer's shirt to the entire image. Although generic *object recognition* and *near-duplicate detection* are two related problems that have been extensively studied over the last decades, logo recognition in natural scenes doesn't necessarily fall under either category.

On one hand, logos can provide some useful prior information to assist detection—they usually contain text and simple geometric shapes and mostly appear on planar surfaces. On the other hand, they are a much broader category than near-duplicates and can take many different forms, or *variants*. A small part of the universe of the Adidas and Coca-Cola logos can be seen in Figure 1.

Global color or shape descriptors are commonly used for logo recognition in clean environments [32], [17], [7], [10]. However, such descriptors have not been successful when it comes to natural images, mainly due to the fact that they are extremely sensitive to background clutter.

The problem we study is the following: given an large annotated database of logos associated to different *classes* (or *brands*) and one query image (or video frame), the task is to detect if one or more of the brands appear in the query. The database consists of a relatively small number of logo *instances* per class, however it may contain a large number of classes. Having more than one instances per class makes detection robust against the multiple forms or appearance variations that a logo may take—again, see Figure 1.

Typically, the database can *scale* up to thousands of classes. Such scales are not very common in generic object detection or recognition. Detectors that operate sequentially for each class, (*e.g.* Viola and Jones [27]) are impractical here, even when sharing features [26]. To maintain fast response on such a large corpus, recognition should be *sub-linear* in the number of classes, as in [20].

We propose a novel representation whereby *local features* are grouped into *triplets* by means *multi-scale triangulation*. The latter applies *Delaunay* triangulation on local feature positions guided by the scales that are available from the feature detector. A simple *signature* is extracted from each triangle and incorporates both visual appearance and local geometry. Each class is represented by the union of signatures over all instances of the class. Such a *generative* model—where all training samples are present individually during query time—has also been successfully applied recently to face recognition [29].

Signatures are highly discriminative, hence sub-linear recognition is accomplished by means of a simple *inverted index* structure. During recognition, signatures are extracted from the query image exactly as in the database instances and class models are ranked according to the inverted index re-

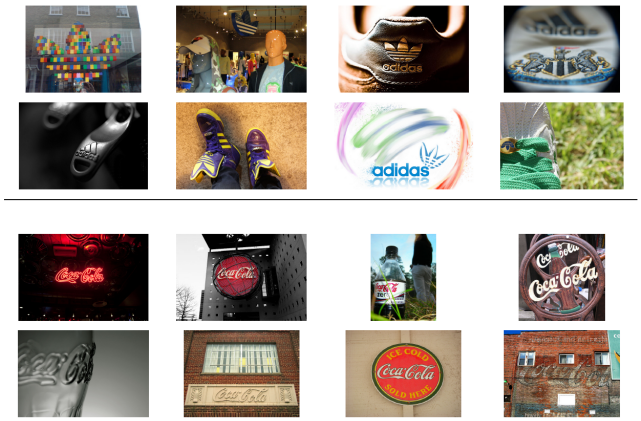

Figure 1: Sample instances of the Adidas (above) and Coca-Cola (below) logos, illustrating the different forms and appearance variations that logos may take in each class.

sponse. The index behaves like a *visual memory* with sub-linear response. Querying a database of thousands of logo classes typically takes milliseconds.

We compare our method against the *bag-of-words* model with *tf-idf* weighting as baseline [25]. We conduct experiments on our own logo data set collected from Flickr[1], since logo images in existing public data sets are either small or insufficient—for instance, the challenging *BelgaLogos* data set [13] focuses on retrieval and does not provide annotated images for training. Our dataset consists of 27 annotated classes and more than four thousand classes with only a few clear instances each. The latter are not annotated and play the role of *distractors*. We can thus simulate large-scale recognition without the need for large scale annotation.

The paper is structured as follows. Section 2 discusses existing work on several related problems. Section 3 presents multi-scale triangulation and signature generation, while Section 4 focuses on class model building and sub-linear recognition. Section 5 presents experimental results and Section 6 draws conclusions and discusses possible future directions.

2. RELATED WORK

Logo recognition. In one of the earliest works on logo recognition, Doermann *et al*. [7] use global affine shape invariants to prune and refine logo matching. In [10], Folkers and Samet propose a content based logo retrieval scheme, where logos are represented by Fourier descriptors and queries comprise geometric shapes. In [17] and [32] the authors focus on *documents* and use OCR techniques to keep the logo while discarding all text. On the contrary, in [5], regions containing text are first detected and regarded as tentative logo positions; logos are then recognized using color and shape features. All the above methods require logos to be on a clean white background.

Interest point grouping. There has been a lot of work recently in grouping *interest points* [4], [9], [11], [16], [24]. However, all these methods work directly on the *descriptor space* after combining interest points. Therefore they are impractical in a large scale retrieval scenario, where quantization *e.g*. with a *visual codebook* seems unavoidable.

Visual word grouping. When it comes to large scale—usually near-duplicate—retrieval, the *bag-of-words* model is the most successful one [25]. Although this model produces sufficient results for datasets up to *e.g*. 10^5 images, retrieval performance drops quickly as the dataset grows larger. More recent methods extend the basic model by grouping visual words. For instance, Yuan *et al*. [31] define *visual phrases* as frequent co-occurring visual word groups, extracted via *frequent itemset mining*. However, the authors do not use visual phrases for indexing; they rather only use them for top-down refinement of the visual codebook.

Indexing geometry. The need for embedding geometry in the index has appeared recently, with datasets scaling up to millions. The bag-of-words model fails to return a good image shortlist and suffers from false positives in this case. Perdoch *et al*. [21] discretize local feature shape (ellipses) to save memory. On the other hand, Avrithis *et al*. [1] incorporate *global geometry* in the index by means of *feature map hashing*. We only consider *local geometry* here; this makes sense since logos typically cover only a small part of the image.

Poullot *et al*. [23] group spatially neighbouring local features and index triangles. Instead of a trained visual codebook they use bucketing and extract a compact binary signature to represent local feature appearance. This representation is highly discriminative, but poses invariance limitations and is very sensitive to outliers. This is not an issue when dealing with near-duplicate detection, but in our problem the limitations would be prohibitive.

Logo retrieval. This is a related problem where, given a query depicting a clean logo, the task is to retrieve as many as possible images from a general image database containing the query logo. In [13] Joly *et al*. use an LSH-based approach and query expansion. The query is a clean cropped logo rather than an entire natural image as in our case. In [30], Wu *et al*. *bundle* local features corresponding to MSER regions and impose simple geometric constraints on the feature bundles. In both cases instances are returned individually and no class model is learnt.

Logo detection in natural scenes. Kleban *et al*. [14] use the *Apriori* algorithm to identify frequent spatial configurations of local features extracted on a spatial pyramid. They construct an inverted index of such configurations, which they look up at query time. Mining requires a large amount of annotated training data per class and is computationally expensive. Bagdanov *et al*. [2] retrieve logos from sport videos by direct matching of SIFT descriptors between the query and a pool of instances in the database. Neither method can scale to more than just a few logos.

3. TRIANGULATION-BASED REPRESENTATION

In the following, we shall assume that local features are detected in all training images and local descriptors are extracted, like SIFT [18] or SURF [3]. Further, we assume there is a generic visual codebook, and each feature is as-

[1] `http://www.flickr.com`

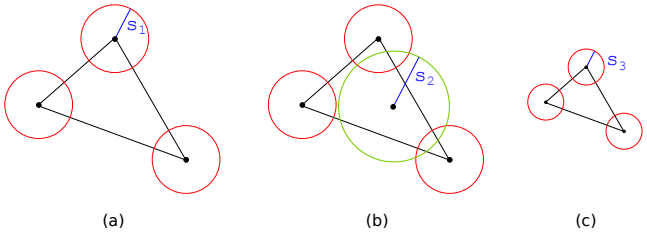

Figure 2: (a) A triangle with all vertices being features of a scale s_1. (b) An outlier feature of scale s_2 is added. This feature will affect triangulation only if $|s_1 - s_2| < w$, where w is the scale window size. (c) A triangle with all vertices being features of scale s_3. This triangle will match the one in (a) if all three visual words of the two triplets are equal.

signed the closest visual word in descriptor space, as in the *bag-of-words* (BoW) model [25]. We require a ground-truth for the training images, consisting of a bounding box and a class label for each logo instance in each image. We use bounding boxes for training, and entire images for recognition. In either case, we represent a logo instance by the set F of the local features it contains. For each feature $f \in F$, we denote by $x(f), s(f)$ its position and scale respectively as given by the detector.

3.1 Multi-scale Delaunay triangulation

We capture feature geometry using *Delaunay triangulation* [6] on local feature positions. Seen as a *global* structure representation, a Delaunay graph is quite sensitive to outliers and for this reason it is typically used *e.g.* for tracking in video sequences, as in [12]. *Local* configurations are more robust to outliers. Feature *triplets* are a common choice as in [23], which in our case correspond to faces of the Delaunay graph. To further enhance robustness, we exploit feature scale as given by the detector and constrain triangulation to features of similar scale.

To see how multi-scale triangulation increases the matching probability of two triangles, consider the triangle of Figure 2(a) where vertices are features of scale s_1. In Figure 2(b), an outlier of scale s_2 has been added. In our case, the outlier will only affect triangulation if scales are similar, *i.e.* $|s_1 - s_2|$ is small. On the contrary, using spatial nearest neighbors as in [23], the two triangles would never match.

Delaunay triangulation. Consider a set P of points on a plane and their *Voronoi diagram*, denoted by $\mathrm{Vor}(P)$. For each point $p \in P$, let $c(p)$ be the Voronoi cell of p. The *graph* $\mathcal{V}$ of $\mathrm{Vor}(P)$ has a vertex for every Voronoi cell and it has an edge joining two vertices if the corresponding cells are adjacent. Consider the straight-line embedding of $\mathcal{V}$, where each vertex $c(p)$ is mapped to p and each edge joining vertices $c(p)$ and $c(q)$ is mapped to line segment $\overline{pq}$. This embedding is the *Delaunay graph* of P, which we denote by $\mathcal{D}$. It is the *dual graph* of $\mathcal{V}$; it is also a *planar* graph, *i.e.* no two edges intersect.

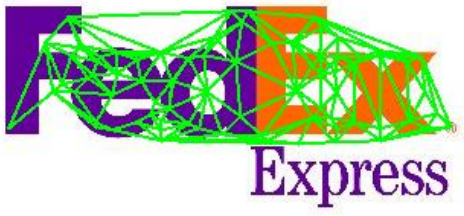

Figure 3: Delaunay triangulation of all local features extracted from a FedEx logo instance.

If P is in general position[2], then all vertices of $\mathcal{V}$ have degree three, implying that all bounded faces of $\mathcal{D}$ are triangles. Under the same assumption, the triangulation is unique. It has the property of being *angle-optimal*, *i.e.* it maximizes the minimum angle over all triangulations of P. Its computation can be very efficient; see [6] for a detailed treatment. A Delaunay triangulation of all local features extracted from a FedEx logo instance is shown in Figure 3.

Multi-scale triangulation. Despite the uniqueness of Delaunay graph $\mathcal{D}$ given a set of points P on the plane, addition of one more point as an *outlier* will affect at least one triangle. Given that in practice outliers are the majority of features, it is quite common that the triangulations of two instances of the same logo may not share a single triangle. For this reason we *constrain* triangulation to *subsets* of features sharing *nearby* scales.

The intuition is that an outlier will only affect triangulation in specific cases, as shown in the example of Figure 2 mentioned above. On the other hand, a triangle of features detected at the same scale s_1 in one logo instance as in Figure 2(a) is still expected to be found having features at some other scale s_3 in a scaled instance of the same logo, as in Figure 2(c).

To build a *multi-scale* triangle representation for each logo instance, we take subsets of local features within a small log-scale *window* and repeat triangulation by *sliding* the window on the log-scale space. *Log-scale* makes sense because a window of size w of one image will be associated to some window of the same size w of a scaled version of the image. It is then expected that the corresponding triangulations will be built on the same feature subsets. By *scale* we shall refer to *log-scale* in the sequel.

Given a specific scale σ, define the scale *window* or *level* $L_\sigma = [\sigma, \sigma + w)$, where w is the window size. Now, given a logo instance represented by feature set F, define the *level-σ feature subset*

$$F_\sigma = \{f \in F \,:\, s(f) \in L_\sigma\} \tag{1}$$

as the set of all features having scale of level L_σ. Accordingly, define the *level-σ point subset*

$$P_\sigma = \{p \in \mathbb{R}^2 \,:\, p = x(f) \text{ for } f \in F_\sigma\} \tag{2}$$

[2]We say that a set of points is in *general position* if it contains no four points on a circle. For randomly distributed points, the chance that four points lie on a circle is very small [6].

Figure 4: Triangulations of local features at multiple scale space levels.

as the set of all corresponding feature *positions* on the plane. Otherwise stated, $P_\sigma = x(F_\sigma)$ is the image of feature subset F_σ under position map x. Similarly, define by $S = s(F)$ the set of all scales, as the image of the entire feature set F under scale map s, and let $s_{\min} = \min(S), s_{\max} = \max(S)$.

Algorithm 1 MULTI-SCALE TRIANGULATION

1: **Input:** Feature set F, window size w, step t
2: **Output:** Triangle collection $\mathcal{T}$
3: $\mathcal{T} \leftarrow \emptyset$
4: $\sigma \leftarrow s_{\min}$
5: **while** $\sigma < s_{\max}$ **do**
6: $\quad F_\sigma \leftarrow \{f \in F : \sigma \le s(f) < \sigma + w\}$
7: $\quad \mathcal{D}_\sigma \leftarrow \text{DELAUNAY}(x(F_\sigma))$
8: $\quad \mathcal{T} \leftarrow \mathcal{T} \cup T_\alpha(\mathcal{D}_\sigma)$
9: $\quad \sigma \leftarrow \sigma + t$
10: **end while**

All we do then is triangulate P_σ and construct a corresponding *level-σ Delaulay graph* $\mathcal{D}_\sigma$ for a number of different scales σ. We iterate by *sliding* the scale window L_σ, *i.e.* linearly incrementing σ in the log-scale space by *step* t. For each scale σ, we keep all detected triangles $T_\alpha(\mathcal{D}_\sigma)$ in level-σ Delaunay graph $\mathcal{D}_\sigma$ having area above α, typically set to 10 square pixels. Finally, we construct the *triangle collection*

$$\mathcal{T} = \bigcup_{\sigma \in \Sigma} T_\alpha(\mathcal{D}_\sigma) \tag{3}$$

as the *union* of all such triangles over all levels Σ used in our sliding window scheme. Algorithm 3.1 outlines our multi-scale triangulation process. In our experiments we typically use parameter values $w = 2.5$ and $t = 2$. Figure 4 shows the triangulations at multiple scale space levels for the local features of the image shown in Figure 3.

3.2 Triangle representation

Each triangle consists of three local features corresponding to three visual words. To represent, index and match a triangle, a *signature* is generated as an ordered triple of the three visual word labels of the triangle features, in lexicographically ascending order. Two triangles match if their signatures are identical, *i.e.* if all three visual words are identical.

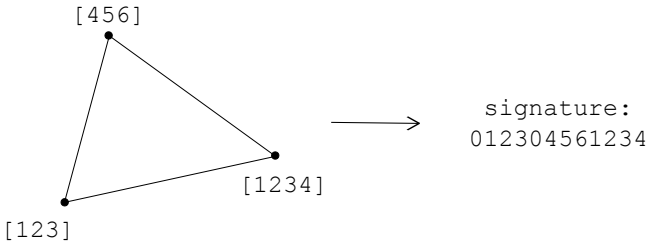

Figure 5: Left: A triangle with the visual word labels of its features in brackets. Right: extracted signature.

In practice, the three label ids are encoded into a long integer, each taking up a small number of bits. For instance, consider the triangle shown in Figure 5, assuming vocabulary size 10^4. It is assigned to visual words `1234`, `123` and `456`, hence its signature is ordered triple (`0123`, `0456`, `1234`), or equivalently integer `012304561234`.

The proposed representation has a number of important properties. It is highly *discriminative*, since *all three* visual words must be identical in order to match a single triangle. It is *robust against appearance variation* because we use a relatively small visual codebook and let the triangle representation compensate for the loss in discriminative power.

It *encodes local geometry* in terms of nearby feature triples arising from Delaunay graphs. It is *robust against variation in spatial configuration* because the relative position of features in a triangle is discarded and because multi-scale triangulation is largely insensitive to outliers.

4. CLASS MODELS AND RECOGNITION

We follow an offline, supervised process to produce a generative model for each logo class. Given all training images of a class, triangles are extracted from each logo instance and *all* signatures are accumulated into the class model. Such an accumulation makes sense, since it allows recognition of any variant of the logo, provided that it appears in at least one of the training images. In certain cases, recognition is possible even if *parts* of a variant appear in different training images. Keeping only *frequent* signatures, as in [31] or [14], would not allow this.

4.1 Class model building

Each class model associated to a logo brand consists of the signatures of all triangles extracted from all instances of the class, along with their frequency of appearance. *Frequency* is defined as the number of logo instances in which each triangle appears, normalized by the total number of instances in the class. This assumes that signatures in an instance are unique, which is typically the case because signatures are highly discriminative. We use frequency to weigh database matches during inverted index queries as in the bag-of-words model.

Our signatures contain local geometry information apart from appearance. Relative positions are discarded but spa-

tial proximity is taken into account because the spatial configuration of three features appearing in a triangle satisfies the Delaunay triangulation properties. We are thus *implicit indexing local geometry* in our logo class models.

Indexing signatures in an inverted file is straightforward, since they are represented by integers. *Tf-idf* weighting is applicable to signatures exactly as visual words. In fact, since the training set is annotated, even *relative frequency* weighting could be used [19]. In this case *index atoms*—the equivalent of images in the baseline bag-of-words model—would correspond to our classes.

The number of triangles per class model varies a lot, and depends on image resolution, logo complexity and scale in the training instances. However, our signature index is so sparse that both memory and recognition speed remain largely unaffected in all our experiments, even in cases of highly populated models.

4.2 Recognition as sub-linear search

Our intuition for logo recognition is that if a logo is present in a query image, at any size or position, its local feature configuration will match the configuration of a number of training instances. Hence Delaunay triangulation over the logo region will produce a number of similar triangles. Our multi-scale triangulation makes this matching robust to outliers.

Although the scale (or orientation) of a logo in a query image may differ from all training instances, it is expected that triangles of features of equal scale will be repeatable in images of different scales.

Recognition is seen as a *sub-linear search* process. The idea is similar to the *visual memory* of [20] but we use the inverted index instead of a decision tree. This approach is very scalable, exactly as the bag-of-words model. We have scaled to thousands of class models in our experiments, with recognition times in the order of a few seconds.

At query time, the triangulation-based representation described in Section 3.1 is repeated for the query image. This time all features are taken into account. We generate signatures for all query triangles and look them up in the index. The response is the number of identical signatures found per class model; we use this number as a score to rank models. A class model is verified if a minimum number of signatures is found.

Spatial matching and verification per training instance could naturally follow as in the bag-of-words model. However, our discriminative signature representation and the sparsity of the index alone are enough to yield high precision. Spatial matching would increase query time without a significant benefit in terms of precision. Matching triangles between two instances of the same class are shown in Figure 6.

5. EXPERIMENTS

We compare our method against the *bag-of-words* model with *tf-idf* weighting as baseline [25]. Since the proposed approach shares the basic feature representation with the baseline, *i.e.* visual codebook of local features, the difference in performance comes mainly from the indexing stage where in our case local geometry is included. In order to evaluate properly on a large-scale scenario, we created our own logo dataset collected from Flickr[3] that contains more than $4K$ classes and is described below.

Figure 6: Matching triangles between two instances of the same class.

5.1 Dataset

A total of 27 classes were chosen for the dataset[4], each one corresponding to a brand, according to the following criteria: (i) To be able to find enough test images in natural environments, (ii) to have a variety of topics (not just car brands, for instance). We then manually selected 40 images per class from Flickr, such that every selected image effectively contained at least one instance of the brand's logo. Once the initial collection was built, all 1080 images were annotated with bounding boxes of the logo instances in the image. We allowed multiple logo instances per class image.

This annotated collection of logos was then split in a test set and a train set. From the 40 images per brand, 30 were randomly selected to be part of the training set, while the rest were the test set.

The query set was then formed with a subset of the test set (5 images times 27 classes = 135 query images). The other half of the test set was used for parameter estimation. To complete this set with negative examples, *i.e.* images that do not contain any logo, we manually gathered 135 more images taken in natural environments, ensuring that they did not contain any class instance.

Finally, the *distractor* set was built: 4397 images from Flickr were selected, all coming from the group "Identity + Logo Design". After visual inspection, we saw that almost all images in this group contain clear logo images, so there was no need for a bounding box annotation.

Sample images from the dataset are presented in Figure 7. The first two rows consist of images containing logos, the third row consists of images that contain no logo, while the last row depicts some of the distractor logo set.

5.2 Evaluation Protocol

We have used SURF [3] local features and descriptors in all experiments. In order to be robust against appearance variations, a small visual codebook of $3K$ visual words is used for our method, denoted as *msDT* in the result figures. The performance of the bag-of-words model typically increases with vocabulary size [22] when the problem at hand is near-duplicate detection. However, this is not always true for logo retrieval. Many logo instances found in natural images are far from near-duplicates as shown *e.g.* in Figure 1. We therefore found a vocabulary of $5K$ visual words to be

[3] `http://www.flickr.com`
[4] `http://image.ntua.gr/iva/datasets/flickr_logos/`

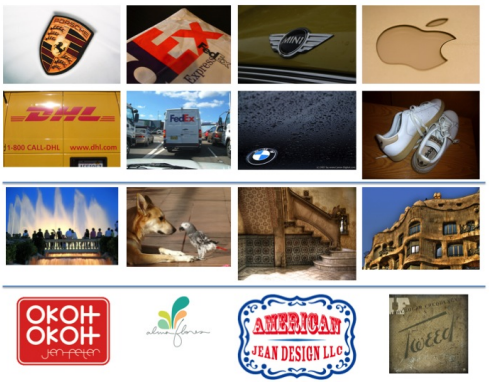

Figure 7: Sample images from the dataset. Rows 1 and 2: images containing logos. Row 3: images containing no logo. Row 4: images from the distractors set.

a good choice for the baseline. The visual vocabularies were all created using 1 Million descriptors from Flickr images.

For the baseline bag-of-words, we normalized the codebook vectors using the l_1 norm and matched them using histogram intersection. To choose the parameters of our approach, *i.e.* the window size w and the step t of the multi-scale triangulation, we experimented on half of the test collection, used for the parameter estimation. The best performance was achieved using $w = 2.5$ and $t = 2$. It is worth noting that accuracy was over the baseline for *all* parameter combinations we checked, when it comes to the large-scale experiment, *i.e.* using the $4K$ distractor logos.

To decide whether a logo is present in a query image, we set thresholds on the similarity of both multi-scale Delaunay triangulation and bag-of-words. To choose the optimal thresholds we experimented again on the parameter estimation test collection. The best values were 0.15 histogram intersection similarity for the baseline and 4 matching triangles for our method.

5.3 Results

We conducted two experiments on our Flickr logo collection. For the first one, we included in the index only the 27 annotated classes, *i.e.* the 1080 manually annotated images. For the second experiment, we also added the $4K$ distractor logos in the index, simulating a large-scale scenario. We measure accuracy as the percentage of correctly recognized logo plus non-logo images, over the total sum of queries.

In the first case, when no distractors are present, the baseline performs slightly better than multi-scale Delaunay Triangulation. Such a result confirms the state-of-the-art performance of bag-of-words models in absence of outliers. Performance results when varying the number of training images are presented in Figure 8. To vary the number of training images, we split the training set into 6 random subsets of 5 images per class.

However, when we add the 4K distractor classes the figures change dramatically. As expected, the bag-of-words model seems clearly affected by the outlier classes and shows a big drop in performance. Accuracy in this case is on average 17% less. On the other hand, the multi-scale Delaunay Triangulation seems to be much less affected in presence of

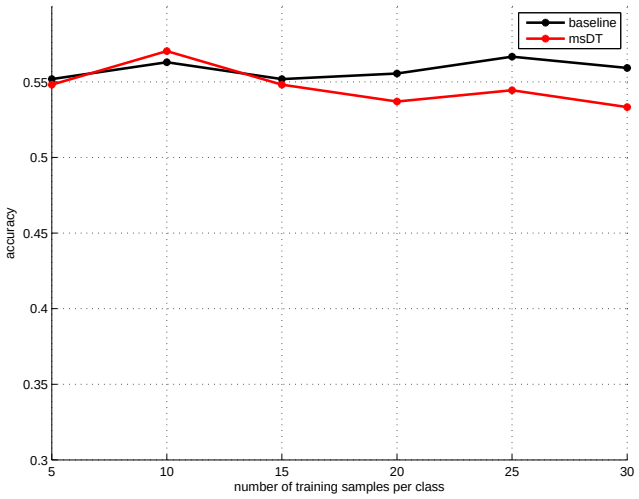

Figure 8: Performance of the proposed multi-scale Delaunay Triangulation approach (msDT) against the baseline Bag-of-Words, on our own Flickr logo dataset *without the distractor classes* (only 27 classes present in the index).

the distractor classes. Although its performance drops, it just decreases for 5.5% on average. The strict nature of the triangle representation allows far less false positives to appear and "confuse" recognition. Results for recognition using all $4K$ classes in the index are presented in Figure 9.

As far as query time is concerned, the proposed approach is no slower than the baseline. Although on query time we look up far more triangles in the index than there are features, posting lists per signature are a much shorter for our inverted index, due to the fact that image signature vectors are far more sparse in the signature space of size N^3 for a vocabulary of N visual words.

6. CONCLUSIONS

The contribution of this work is two-fold. First, we propose a novel discriminative triangle representation that includes local geometry information. The triangles come from multi-scale Delaunay triangulation, a robust process that can be reproduced in the query image. Each triangle forms a signature that can be indexed using an inverted file structure. This allows robust recognition in less than a second, for a logo database of thousands of classes. Second, we propose a simple learning process that accumulates triangles of each class, while preserving the appearance frequency as well. Experiments on a large dataset show that multi-scale triangulation outperforms bag-of-words. The latter performs really well for a few dozen logos, but is very prone to errors as the number of logos grows.

In the future, optimizations for RANSAC based on the multi-scale Delaunay triangulation signatures could be investigated in order to localize the logo in the query image. Each matching triangle defines a unique affine transformation that can guide RANSAC efficiently. Advantages of using a more sophisticated scale-space could also be investigated. Finally, we intent to exploit the speed and scalability of the proposed approach for logo detection in videos.

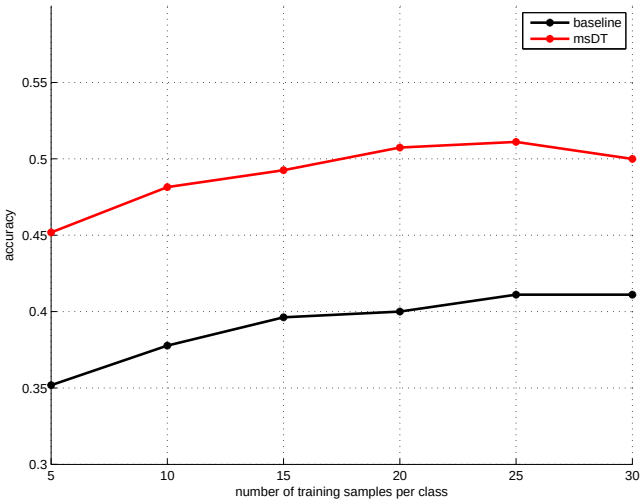

Figure 9: Performance of the proposed multi-scale Delaunay Triangulation approach (msDT) against the baseline Bag-of-Words on our own Flickr logo dataset *with the distractor classes* (more than 4K classes in total in the index).

7. ACKNOWLEDGMENTS

This work was supported by the European Commission under contract FP7-215453 WeKnowIt.

8. REFERENCES

[1] Y. Avrithis, G. Tolias, and Y. Kalantidis. Feature map hashing: Sub-linear indexing of appearance and global geometry. In *ACM Multimedia*, 2010.

[2] A. Bagdanov, L. Ballan, M. Bertini, and A. Del Bimbo. Trademark matching and retrieval in sports video databases. In *MIR*, 2007.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

[4] M. Brown and D. Lowe. Invariant features from interest point groups. In *BMVC*, 2002.

[5] T. Chattopadhyay and A. Sinha. Recognition of trademarks from sports videos for channel hyperlinking in consumer end. In *ISCE*, 2009.

[6] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars. *Computational geometry: algorithms and applications*. Springer, 2008.

[7] D. Doermann, E. Rivlin, and I. Weiss. Logo recognition using geometric invariants. In *ICDAR*, 1993.

[8] A. dos Santos and H. Kim. Real-Time Opaque and Semi-Transparent TV Logos Detection. In *WACV*, 2007.

[9] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. *ECCV*, 2004.

[10] A. Folkers and H. Samet. Content-based image retrieval using Fourier descriptors on a logo database. In *ICPR*, 2002.

[11] K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features. 2006.

[12] H. Jiang and S. X. Yu. Linear solution to scale and rotation invariant object matching. In *CVPR*, 2009.

[13] A. Joly and O. Buisson. Logo retrieval with a contrario visual query expansion. In *ACM Multimedia*, 2009.

[14] J. Kleban, X. Xie, and W. Ma. Spatial pyramid mining for logo detection in natural scenes. In *ICME*, 2008.

[15] B. Kovar and A. Hanjalic. Logo detection and classification in a sport video: video indexing for sponsorship revenue control. *Storage and Retrieval for Media Databases*, 4676(1):183–193, 2001.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *BMVC*, 2004.

[17] Z. Li, M. Schulte-Austum, and M. Neschen. Fast Logo Detection and Recognition in Document Images. 2010.

[18] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[19] C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 1 edition, July 2008.

[20] S. Obdrzalek and J. Matas. Sub-linear indexing for large scale object recognition. In *BMVC*, 2005.

[21] M. Perd'och, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, 2009.

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

[23] S. Poullot, M. Crucianu, and S. Satoh. Indexing local configurations of features for scalable content-based video copy detection. In *ACM workshop on Large-scale Multimedia Retrieval and Mining*, 2009.

[24] Y. Shen and H. Foroosh. View-invariant action recognition using fundamental ratios. In *CVPR*, 2008.

[25] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[26] A. Torralba, K. Murphy, and W. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *CVPR*, volume 2, 2004.

[27] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[28] J. Wang, L. Duan, Z. Li, J. Liu, H. Lu, and J. Jin. A robust method for tv logo tracking in video streams. 2006.

[29] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *PAMI*, pages 210–227, 2008.

[30] Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling features for large scale partial-duplicate web image search. In *CVPR*, 2009.

[31] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: from visual words to visual phrases. In *CVPR*, 2007.

[32] G. Zhu and D. Doermann. Logo matching for document image retrieval. In *ICDAR*, pages 606–610, 2009.