Logo Localization and Recognition in Natural Images Using Homographic Class Graphs

Raluca Boia RBOIA@IMAG.PUB.RO
Image Processing and Analysis Laboratory
University "Politehnica" of Bucharest, Romania, Address Splaiul Independenţei 313

Corneliu Florea CORNELIU.FLOREA@UPB.RO
Image Processing and Analysis Laboratory
University "Politehnica" of Bucharest, Romania, Address Splaiul Independenţei 313

Laura Florea LAURA.FLOREA@UPB.RO
Image Processing and Analysis Laboratory
University "Politehnica" of Bucharest, Romania, Address Splaiul Independenţei 313

Radu Dogaru RADU_D@IEEE.ORG
Natural Computing Laboratory
University "Politehnica" of Bucharest, Romania, Address Splaiul Independenţei 313

Abstract

We propose a method for localization and classification of brand logos in natural images. The system has to overcome multiple challenges such as perspective deformations, warping, variations of the shape and colors, occlusions, background variations. To deal with perspective variation, we rely on homography matching between the SIFT keypoints of logo instances of the same class. To address the changes in color, we construct a weighted graph of logo interconnections that is further analyzed to extract potentially multiple instances of the class. The main instance is build by grouping the keypoints of the graph connected logos onto the central image. The secondary instance is needed for color inverted logos and is obtained by inverting the orientation of the main instance. The constructed logo recognition system is tested on two databases (FlickrLogos-32 and BelgaLogos), outperforming state of the art with more than 10% accuracy.

to be published in Machine Vision and Application. For final version please check on Springer

1. Introduction

A logo is a graphic entity containing colors, shapes, textures, and perhaps text as well, organized in some spatial layout format and identifies goods, services, or organizations. Logo localization and recognition is a subproblem of object detection and recognition and a challenging pattern recognition task.

The marketing industry is continuously looking for methods to automatically evaluate and, subsequently to increase the impact of marketing campaigns (Chan et al., 2010), (Lewis et al., 2014). It is easy to identify the following use cases: (1) In automotive industry, the automatic logo recognition is used in marketing studies, allowing the producers to better understand their customers by analyzing their transportation patterns; also it complements the plate registration towards car identification especially in forensics. (2) Identification and recognition of logos in official documents can improve classification and processing efficiency. (3) In sports transmissions, the duration and position of the logos display impacts financial factors such as sponsors requiring a certain level of visibility of their trademarks to be ensured; an example in this direction is in the work of Bagdanov et al (Bagdanov et al., 2007). (4) General advertising aims to assess the impact of marketing campaigns; companies are interested in logo detection while gathering evidence of similar already existing logos, discovering either improper or non–authorized use of their logo or exposing visually infringing logos, etc.

Table 1. Logo recognition methods in natural images.

Method	Descriptor	Learning Scheme	Particularities	Retrieval/Recogn.
Romberg et al. (2011)	SIFT	BoW	Bundling	Retrieval + Recogn.
Revaud et al. (2012)	SIFT	BoW	Weigting SIFT	Retrieval
Romberg et al. (2013)	SIFT	BoW	Bundling +Weighting	Retrieval
Lu et al. (2014)	HoG	SVM	2-Level	Recogn.
Li et al. (2014)	HoG+ASIFT	SVM /NN	HoG prefiltering	Retrieval + Recogn.
Krapac et al. (2014)	SIFT	BoW	Class prototype	Retrieval
Ries et al. (2014)	HoG	SVM	CCCP	Retrieval
Boia & Florea (2015)	SIFT	RANSAC	Class Homography Graph	Recogn.

1.1. Related Work

Logo recognition is a particular use case of generic object recognition class of solutions. The algorithms from the generic class of object recognition can be divided in two categories: generative (Lowe, 1999), (Schneiderman and Kanade, 2003), (Fergus et al., 2003), (Dorko and Schmid, 2005) and discriminative (Torralba et al., 2004), (Opelt et al., 2006), (Viola and Jones, 2004), (Kumar and Hebert, 2004). Discriminative techniques learn directly from the given labelled data a way to separate the data space into class segments (Ng and Jordan, 2001). They usually don't easily handle missing data and have a low ability of incorporating prior knowledge. Generative algorithms solve a general problem as an intermediate step and usually create object class models using jointly the data and the labels, being more suitable to high intra-class variation as is the case of logo recognition. The proposed method falls in the generative category.

Into the specific problem of logo recognition, we note two main directions: specific domain recognition (such as vehicle logo or document logo recognition) and general logo recognition in natural images.

The document logo recognition, (Zhu and Doermann, 2007) was the first category to gain momentum. While being characterized by binarization, noise and resolution loss, it is not affected by perspective or warping distortions and other challenges associated with natural logos.

The vehicle logo detection is of high interest for marketing branches of automotive industry; here we note the work of Psyllos et al. who used a Bag of visual Words (BoW) derived solution (Psyllos et al., 2010), and tested on small databases with few classes. More recently, this niche has been developed by the introduction of the *Vehicles29* dataset (Krapac et al., 2014). Yet this category is not influenced by large warping as the background material is rigid.

While dealing with the *logos in natural images* two problems are approached: logo retrieval (i.e. given an image query containing logos, nominate other images with similar content) and logo integrated recognition (i.e. place a bounding box around the logo - localization followed by identification of its class). Most of the state of the art systems address the retrieval aspect.

The first approaches (Bagdanov et al., 2007), (Joly and Buisson, 2009) concerning generic logo retrieval were limited in handling large image collections. Later methods (Kleban et al., 2008) retrieved logos by performing frequent item-set mining to discover association rules in spatial pyramids of visual words. Revaud et al. (Revaud et al., 2012) used a bag-of-words (BoW) based approach coupled with learned weights to penalize inter-class appearances, while Romberg et al. (Romberg et al., 2011) enhanced the BoW system by embedding spatial knowledge into the cascaded index. Romberg and Lienhart, (Romberg and Lienhart, 2013), extended the BoW by bundling on the min-hashing of SIFT-based visual words.

Reporting the location of logos was met in the work of Lu et al. (Lu et al., 2014), who built a two level detection system where the first step contains an ensemble of linear detectors while the subsequent K-D trees discriminate among the resulting feature vectors. Also Li et al. (Li et al., 2014) used a Support Vector Machine (SVM) to select the HoG described potential windows of interest and further classify them with affine SIFT and nearest neighbor; yet they still report precision and recall instead of detection rate. A summary of existing works may also be followed in table 1.

1.2. Paper Structure and Contributions

Logo detection is extremely challenging mostly due to perspective deformations, varying background, occlusions, resolution variability, since the logos can high resolutions from 1000×1000 to 20×20 pixels. Another issue to handle is the warping of the patterns, since not all supports are planar. Although it can be accounted as a problem of near-duplicate retrieval, logo recognition differs by the fact that it must handle large variations in the color schemes or even in the shape of the logos of the same class. This high intra-class variability is caused by the very common process of

Figure 1. Sample images containing the logos of Adidas, Coca Cola and Paulaner (from the FlickrLogos-32 database) in the left, and Puma (from the BelgaLogos database) in the right. Note the variability in logo appearance due to size, shadowing, color balance, warping, etc.

rebranding of the companies and ro properly address this issue we the multiple instance learning paradigm.

The proposed system is tested and validated on two databases: FlickrLogos-32 and BelgaLogos. Figure 1 contains examples from both datasets, revealing some of the mentioned challenges.

In the current paper we contribute by a new method for class description that offers supplementary capabilities for generalization and, more importantly, with a system exhibiting great performance in logo recognition and localization. The current work is developed from the method previously reported by us (Boia and Florea, 2015) with following major technical differences: (1) while in our older work a single template is build per class (single instance learning), here the class graph is processed and, if necessary, a secondary model is automatically build to model the examples depicted in inverted colors (thus implemented as multiple instance learning - MIL); (2) while in our older work (Boia and Florea, 2015) an undirected binary graph was used, here we switch to a weighted graph that permits better homography composition and automatic identification of the nodes cluster thus of the secondary model. Furthermore, here extend the testing and the analysis of the results.

The paper structure consists in the following sections: in section 2 we present the method used for describing individual objects (i.e. logos) and the necessary steps to build the class model; meanwhile we will describe the method employed for matching the logos in the training database and respectively with the learned models while testing; the matching and the way of building the class model are central aspects of the proposed system. Implementation details are presented in section 3, followed by the achieved results and discussions in the same section.

2. Building the class model

The proposed method (see the scheme in figure 2) builds class models by gathering the relevant features (descriptors and locations) from all the training logo crops (i.e. extracted sub-images of the images that contain only the logo region) for each class. In order to identify the corresponding features from the images, a homography matching is employed for each pair of images in the training set of the same class. Using SIFT description and RANSAC algorithm, this process will reveal the perspective transform that connects each two images. By gathering the information of this stage, a class graph is built in which the images are the nodes and the homograhies between them are the edges. The most representative image of the class will be the node of the graph with the highest number of connections to the nodes, as it clearly holds the most relevant information. After being projected by the homographies found, the keypoints and descriptors from all the images in the class are gathered on this central image. The class graph not only has the role of selecting the central image, but it also indicates the homography chain connecting images that do not have a direct link to the central one. Next a quality map of the matching separates the good fits from the bad ones.

In order to reduce the errors and increase the amount of information accumulated in the model, the graph's edges are weighted with weights inversely proportional to the number of pairs of matching points revealed by the homography connecting the nodes. This is one of the improvements over the previous work (Boia and Florea, 2015), where the class graph was unweighted. This modification leads to a noticeable increase of the recognition accuracy. While many previous works describe classes by a single prototype (Romberg and Lienhart, 2013), (Boia and Florea, 2015), here we use potentially two the second modelling color inverted cases. The system automatically detects the classes which present inverted instances, using a class compactness criteria and generates a second model of the class.

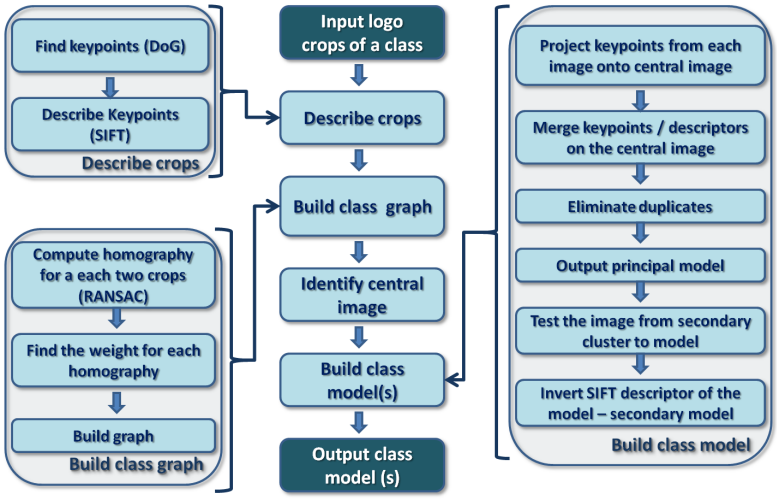


Figure 2. The schematic of the proposed system while learning the classes. The main technical contributions lay in the "Build class model" block.

2.1. Feature extraction

The image description is done using the Scale-invariant feature transform (SIFT) (Lowe, 2004) algorithm: the Difference-of-Gausssians (DoG) keypoint extractor and SIFT local features for the describing the keypoints' neighbourhoods. The choice of this algorithm is motivated by its robustness to changes in scale, orientation, shear, position, camera viewpoint and illumination and it was widely used in applications for logo recognition (Romberg et al., 2011), (Romberg and Lienhart, 2013), (Revaud et al., 2012), (Psyllos et al., 2010). While other choices for keypoints do exist, we do not take them into account in our paper but we refer the interested reader to the work of Mikolajczyk et al. (Mikolajczyk et al., 2005).

SIFT offers a robust representation of the logos; however adaptation to the particularities of the logo problem are needed. First of all, an increase of the default edge threshold that eliminates peaks of the DoG scale space will enforce the extraction of a higher number of keypoints in the logo area. The final value of the threshold is 100, which is 10 times higher than the default one, a value empirically found to be good enough to expand the description of the image with a high number of features, for a better description, while not introducing irrelevant keypoints. The general idea is: more keypoints increase the chances to match extremely distorted logos to the class prototype, but too many keypoints make the algorithm too slow and increase the false positives as the keypoints will turn into pixels and too many matches can be found.

Another adaptation consists in upscaling the small-sized training logos, a procedure that increased the number of keypoints extracted. Revaud et al. (Revaud et al., 2012), while experimenting with BelgaLogos (characterized by very small logos) noted that: the performance on small logos is lower than on larger ones and doubling the image size increases the accuracy. Instead, only images having one dimension below $T_{size} = 200$ pixels are upscaled. This value was established empirically by multiple tests of homography matching between small sized logos.

2.2. Image matching

Image matching has the purpose of revealing the spatial correspondences between the pairs of images. This process, also known as image registration is done by finding the proper projective transform between the input images. Its purpose is to estimate the homography matrix that links the plane of the second image to the one of the first. This solution functions for planar or almost planar scenes. Since the logos are usually on planar supports or at least parts of them are planar, the method proves to be highly efficient in logo recognition.

2.2.1. Homography

The homography transform is the connection between two images of the same scene. Denoted by H, it has the role of mapping a point (a, b) from the plane of the first image to the coordinates (x, y) on the plane of the second image:

$$\begin{bmatrix} a \\ b \\ 1 \end{bmatrix} H = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, where \; H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (1)$$

Homography estimation algorithms (Hartley and Zisserman, 2004), (Dubrofsky, 2009) are based on finding fea-

ture correspondences in those images. Given the features from two input images, the putative matches are computed (Brown and Lowe, 2007). Evidently, a large amount of outliers will be present; thus, RANSAC (random sample consensus) algorithm (Fischler and Bolles, 1981) is employed to estimate H. The matching process is visually exemplified for a pair of crops in figure 3.

2.2.2. RANSAC

The RANSAC estimation of homography is widely used in object recognition and tracking applications (Klein and Murray, 2009), (Jin et al., 2008), (Sheikh et al., 2009), scene understanding (McLauchlan and Jaenicke, 2002), (Brown and Lowe, 2003), (Lin and Medioni, 2007), 3D modelling (Simon et al., 2000), (Snavely et al., 2008).

The algorithm iteratively randomly selects 4 feature correspondences, for which the homography H is found with the direct linear transformation (DLT) (Hartley and Zisserman, 2004). After the iterations are finished, the solution with the largest consensus set (i.e. the most inliers) is selected. The pseudocode for the used version of RANSAC estimation of homography is presented in table 2.

A high number of iterations signifies a larger probability to find the best match. If the inlier probability is p_i, then the probability of finding the correct transformation $p(H^{Correct})$ after n trials is:

$$p(H^{Correct}) = 1 - (1 - (p_i)^r)^n, \tag{2}$$

where r is number of samples extracted at each iteration; in our case $r = 4$.

The mathematical conclusion is that the number of iterations needed in order to obtain the correct homography with a certain probability $p(H^{Correct})$ must follow the rule (Raguram et al., 2008):

$$n \geq \frac{\log(1 - p(H^{Correct}))}{\log(1 - (p_i)^r)} \tag{3}$$

For panorama stitching (Brown and Lowe, 2007), 500 RANSAC iterations, with an inlier probability of $p_i = 50\%$, lead to a probability of finding the correct homography of $p(H^{Correct}) = 1 - 10^{-14}$. In our case, the inlier probability is much lower, since the logo areas are usually obstructed by various occlusions, many logos are warped or distorted due to their support, and especially for the full natural images (considered in the test phase), the background to logo area ratio is very high. Computing it on the training database (using full images and not crops) led to a value of $p_i \approx 11.5\%$. In order to obtain the same certainty of obtaining the correct homography $p(H^{Correct}) = 1 - 10^{-14}$, the number of needed trials is $184,295$. Consequently, we use $200,000$ iterations for the estimation of the homography through RANSAC.

In the test phase all the class models are presented to the test image and very often no homographies can be found between the test image and a certain model. Hence, there is no need to follow all the iterations in some cases. If the initial number of putative matches is below 20, experiments have shown that there is a high probability that there is no correct homography to be found, especially since training showed that a correct homography has at least 20 inlier matches. This means that out of those 20 putative matches, all of them should be inliers in order to output a correct homography. As an inlier probability of 100% is near impossible, we built an early termination criterion, stating that if the number of putative matches is below 20, the search stops.

2.3. The interconnections graph

After all the possible connections between the training images are done, the links created are organized in the form of a graph.

The result of the first stage of training is that for n crop images per class, $n(n-1)/2$ image pairs are potentially matched. In reality, because of occlusions, inverted colors, large variations or distortions in shape, not all the pairs of images have enough common information so to get a correct homography. This is not an issue, since a pair of unconnected images may still have a linkage, indirectly, through other images that were successfully matched. This means that all the images may be interconnected directly or indirectly. This aggregation forms a graph, in which the nodes are the images and only those generated by a homography have a link connecting them, similarly to the idea in (Romberg et al., 2011).

An interconnections graph holding the information about the paths between the images is created for each class. By exploiting the information from the class graph, the image with the highest number of connections to the others is found. The image will be called the *central image*. Clearly, this is the most representative image for the logos of the class and holds the largest quantity of information about its aspect. This image is thus chosen as a support for the template of the class, as the following stage of the training consists in aggregating the relevant information from the other images onto it.

A summarized representation of a class graph is in figure 4. The figure is presented at a smaller scale than the real one, since usually a class graph holds an average of 300 links for 30 training images.

Differing from our previous work (Boia and Florea, 2015),

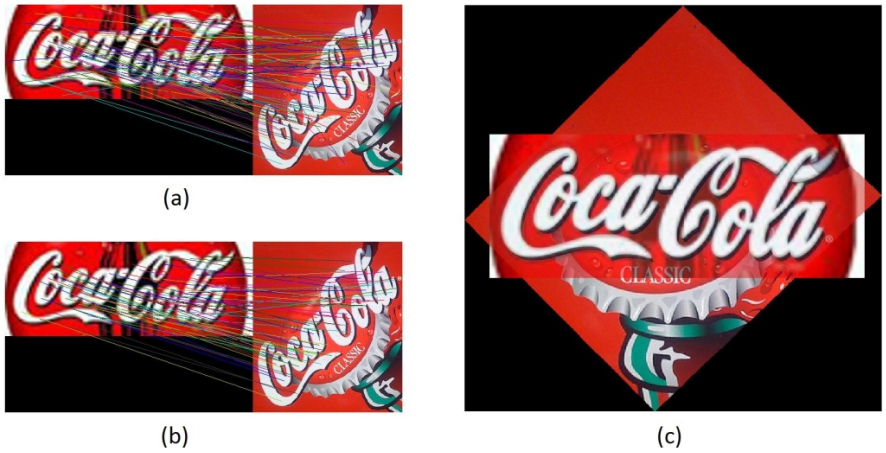


Figure 3. The matching process for two crops of the class "Coca Cola": (a) The matching of the descriptors, (b) the remaining pairs after RANSAC, (c) the two crops overlaid after the homography is applied

Table 2. The RANSAC algorithm as it was used for panorama stitching (Brown and Lowe, 2007) compared with the modifications proposed by us to match the specific of logo recognition problem. For the training part we select a single homography H (the best one), while for testing, as we detect multiple logos, we select all that have more inliers than `nAcceptable`. `nMatches` is the number of initial putative matches computed before applying RANSAC(Brown and Lowe, 2007).

Base-line RANSAC (Brown and Lowe, 2007)	Modified RANSAC

`nIter = 500;`
`nBest = 0;`
`for` `i:=0, i<nIter, i++,` **`do`**
`P4 = SelectRandomSubset(P)`
`Hi = ComputeHomography(P4)`
`nInliers = CompInliers(Hi)`
`if` `nInliers > nBest` **`then`**
`H = Hi`
`nBest = nInliers`
`end` **`if`**
`end` **`for`**
	`nIter = 200,000;`
	`nBest = 0;`
	`if` `nMatches < 20` **`then exit`**
	`for` `i:=0, i<nIter, i++,` **`do`**
	`P4 = SelectRandomSubset(P)`
	`Hi = ComputeHomography(P4)`
	`nInliers = ComputeInliers(Hi)`
	`if` `nInliers > nAcceptable` **`then`**
	`H.add(Hi)`
	`end` **`if`**
	`end` **`for`**

the graph is weighted, where the weight of an edge is inversely proportional to the number of pairs of points matched between the images that it connects. This choice brings an increase in the accuracy of the system and is explained more detailed in the next section.

2.4. The class model

The purpose of the training stage is to conglomerate all the representative keypoints and their corresponding descriptors on a common ground, which was chosen to be the central image. Since each image has its unique description, it is clear that every homographic connection between the pairs of images will output a different set of inlier matching descriptors and keypoints. Using the computed homographies, all the revealed descriptors and keypoints are projected on the plane of the central image and their reunion will form the model of the class. This will evidently be a more powerful description of the class than the plain usage of the central image, as it collects the most important information from all the instances of the class, thus leading to a more complete description of the logo. An example of two images whose keypoints and features are merged on the central image can be seen in figure 5.

Merging features onto a single image means projecting all the features (keypoints and descriptors) into the plane of that image. The projection does not affect descriptors as SIFT features are resistant to perspective change. Consequently, only the keypoints require a computation effort, while their corresponding descriptors are merged to the class model unchanged.

For images that are directly connected to the central image,

(a) (b) (c) (d) (e)

Figure 5. The merging onto the central image of the representative keypoints coming from 2 images. (a) the matching pairs of descriptors between the first image and the central one, (b) the important keypoints of the first image, (c) the matching pairs of descriptors between the second image and the central one, (d) the important keypoints of the second image, (e) the reunion of the important keypoints on the central image

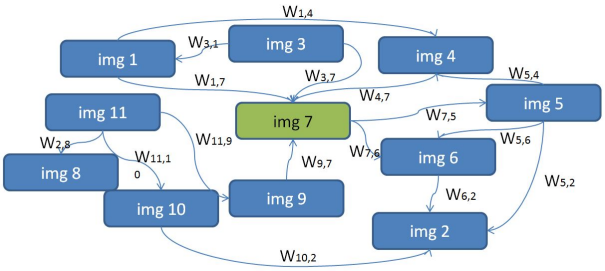


Figure 4. Small part of a class graph. The image most connected to the others is highlighted, and is the *central image*. The weights W_{ij} are proportional with the number of point pairs matched between images i and j.

a simple application of the homography transform between that image and the central one suffices for computing the new positions of the keypoints. This idea is similar to (Psyllos et al., 2010), but with differences in the key points of the training. While Psyllos et al. manually select the best image of the class, the proposed method automatically computes the central image. Moreover, their method is tested on a smaller database where all the images are connected to the reference and they also use prior knowledge for the approximate localization of the logos.

Contrary, the proposed technique takes into account the case of the images without a direct connection to the central image. The class graphs actually indicate that most often images are not directly connected to the reference one, thus leading to a high necessity of developing a method for extracting information from them as well. The gist of the technique used to compute the homography between two unconnected images is to compose the intermediary homographies that indirectly chain them. For example, in figure 6 images 1 and n are connected through images $2, 3, \ldots n-1$. The homography between images 1 and n is the composition of the homographies of the images connecting them:

Figure 6. Example of a chain of images connecting first image to the last one. H_{ij} projects image j onto image i while the inverse of H_{ij} projects i onto j.

$$H_{1,n} = H_{1,2} \circ H_{2,3} \circ \cdots \circ H_{n-1,n} \tag{4}$$

This actually implies using the paths created in the graph to connect the images to the central one. It is well known that a homography transform introduces a small error, named *re-projection* error. Thus, when applying a chain of homographies, these errors will accumulate. Evidently, the longer the path of intermediary images, the larger the final error. Choosing the central image as the one most connected to others means selecting the case with the highest possible number of images directly linked to the reference and thus smaller errors.

For the images without a direct connection, a compromise must be set such as to keep errors as low as possible. The solution is to connect an image to the central one by the path with minimum weight. We use the Djikstra algorithm for computing this path. In our previous work (Boia and Florea, 2015) for the unweighted graph used, the shortest path meant literally using the path containing the fewest nodes between the source image and the target, that introduced the smallest error possible. However, this will not also significate the most complete description of the class. A shorter path between two images introduces the least number of errors, but the case might be that not enough information is extracted from the images connecting them. A longer path means much more data to be accumulated from the intermediary images. In order to find a balance in the problem of a small error versus a high amount of data, the class graph is modified as to be a weighted one.

The direct links between images can bear more or less in-

formation regarding the model of the class depending on the number of matching pairs of points they yield. Thus, a more descriptive model of the class is built by considering a more complete graph that carries information about the matching score (i.e. number of pairs) between the images it links. The weights of the edges will be equal to the inverse of the number of matching pairs of points. This ensures that the shortest path has indeed the most pairs of points to describe the logo. The resulting class model has more points, thus a richer description than in our previous algorithm (Boia and Florea, 2015), while keeping a low error level.

2.5. The error map

To avoid flooding the class model with irrelevant information, a pre-filtering of the list of keypoints is necessary. No homographic overlapping of images can be perfect, especially when parts of the objects are not perfectly planar or occlusions appear. Features situated in areas of imperfect matching must be identified such as to avoid adding them to the final class model.

The solution is to build an error map for each homographic matching of images, its target being to separate the regions of qualitative overlapping from the erroneous matching regions. The map values are directly related to the correctness of the matching in that area. This procedure is similar to shape matching score from (Florea et al., 2013): given a training set of shapes the joint distribution is computed; given an actual pair, the score is retrieved by back projecting the joint distribution and enriched by histogram of oriented gradients (Dalal and Triggs, 2005) for each position.

The map indicates for each pixel whether it is in an area of accurate projection of the images or it is in a region of occlusion and distortions of the shapes. Given that the common element in all images is the logo object, we can say that the error map will reveal the common usable parts of the logos to be added to the model. Figure 7 describes the matching process and the overlaid images after applying the homography. Figure 7(c) shows the error map created for this matching, where the darker regions in the map show the areas where the matching is correct. The areas of occlusion or difference in the shape of the logo are indicated by the lighter values in the map showing a poor quality of the matching.

2.6. Class descriptor

The final descriptor is formed by the reunion of the keypoints and the descriptor vectors over the central image. This model holds a powerful description of the class, far superior to the sole usage of the central image. This fact is illustrated in figure 8, where it can be seen the successful detection of a logo using the class model, compared to the failure obtained for the case of using just the regular central image.

(a) (b)

Figure 8. (a) Matching fails when using only the central image. (b) Successful detection when using the model of the class. We note that, for clarity, only the first detection is showed in the right hand plot, while in this case all Adidas logos were, in fact, found.

2.6.1. Quantization of descriptors

Some of the positions found originate from the same interest points of the logo and, thus, become adjacent on the model image. This leads to points in the model that are extremely similar as location and as description. Consequently, a quantization step is employed, to keep the unique keypoints and features describing the image. Brown and Lowe (Brown and Lowe, 2007) suggested to use K-D trees as it offers both various quantization levels and speed improvement while matching, thus reducing the testing computation time.

2.7. Luminance inverted logo classes

Since the logo images are taken under different illumination conditions and also the brand logos can undergo slight changes in color setups, clearly SIFT description of the grayscale image is appropriate for the description of the shapes. This will overcome the changes in color palette, but not a full inversion of the luminance levels. This issue cannot be overcome by the normal SIFT description, as the gradients' orientation will be totally inverted. Furthermore, while we describe this situation as "color inverted logo", we stress that due to the fact that SIFT descriptor is computed only the equivalent grayscale image, the inversion refers in fact to grayscale/luminance levels; yet inversion of the RGB colors implies inversion of the grayscale levels, maybe at a smaller extent.

For the classes of logos that are in this special case, such as Adidas, HP, Puma, Nvidia, Coca Cola etc., the need to create a luminance-inverted class model arises. Examples can be seen in figure 9 (a). What is particular for these classes is that in the training phase two distinct clusters are formed in the interconnections graph. The images in the training set having the same luminance logic will be able

(a) (b) (c)

Figure 7. Constructing the error map (a) The matching pairs of points (b) The mosaic of images after applying the found homography. (c) The error map. The darker areas show good quality of matching.

to match to each other through homographies, while the others with the inverted luminance levels will gather in a separate cluster.

The idea to create separate classes for logos that include both original and negative versions appeared in (Revaud et al., 2012), but we differ by the fact that while they created this duplicate for manually selected negative version of some logos, we achieve this by an automatic process, in which not necessarily the negative logo will be learned, but the more distinct versions. This step for handling inverted instances of logos is one of the key differences with respect to our previous work.

Thus, in order to automatically detect the classes that are in this condition, we use a class compactness criteria - by analyzing the class's graph: if two separate connected components are identified, then the class must have two class models. As the central image is taken to be the one with the most connections, it means that the principal model of the class is described by the largest connected component of the graph. If the images in the other cluster of the graph can match to the built model of the class once their luminance is inverted, then it means that indeed the class will need an inverted luminance model.

Building another model of the class by selecting a secondary central image from the second cluster in the graph and following exactly the same steps as for the first model will lead to a weak description of the class, since this cluster contains fewer images than the principal one, thus less information. The principal model has a high power of description and an inversion of it is the way to go for an appropriate representation of the inverted part of the class. This time, the easy task is when it comes to creating the keypoint vector, as they positions are kept intact, while for the descriptors, they must be inverted.

Inverting a SIFT descriptor actually means reorganizing for each of the 4x4 histograms the 8 bins in the following manner: if the initial histogram of a patch had the bins order 0,1,2,3,4,5,6,7, then the corresponding inverted grayscale patch will have the SIFT descriptor having the bins order 4,5,6,7,0,1,2,3. This is also pictured in figure 9 (b).

In the testing phase, for the classes that failed the class compactness test and proved that an inverted model is necessary, both the normal model and the inverted model will be tested against the images.

3. Implementation and Results

For implementation, we started from the basic image matching method based on the VLFeat open source library (Vedaldi and Fulkerson, 2010) and complemented with Matlab code.

3.1. Evaluation procedure

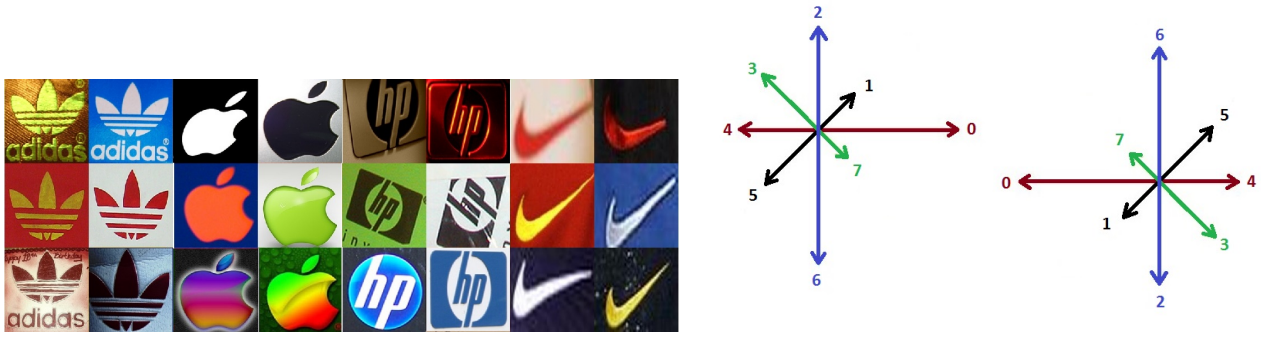
Given a query image, we count a true detection if the found logo is present in that image and if the intersection-over-union (i.e. the Jaccard index) is above 50% as defined in Pascal VOC protocol (Everingham et al., 2010).

For a detailed view, along with standard measure (True Positives (TP) rate, False Positives (FP) rate, True Negatives (TN) rate and False Negatives (FN) rate), we will report Precision (or positive predictive value - PPV) and Accuracy (ACC). We recall that these are defined as:

$$\begin{aligned} PPV &= \frac{TP}{TP+FP}; \\ ACC &= \frac{TP+TN}{TP+FP+TN+FN} \end{aligned} \tag{5}$$

3.2. Databases. Training sets

FlickrLogos-32. For a realistic evaluation of the proposed method, we first choose the FlickrLogos-32 database (Romberg et al., 2011), which was formed by carefully selecting images from collections of photos in a real word environment, depicting brand logos. The testing/ training scheme is the same as in the case of Romberg et al. (Romberg et al., 2011): 30 images per class for training



(a) (b)

Figure 9. (a) Example of classes that have great variations in color and luminance levels, including color inversion (which implies grayscale level inversion). (b) SIFT bins arrangement before and after inverting the grayscale level.

and 30 images per class for testing phase for a total of 32 classes. For the training phase, we used only the crops of the logos in the images, while for the test part, we scan the entire images.

Taking into account the average object size, when compared to other databases for object detection, the FlickrLogos-32 can be considered a small-object dataset. This adds up to the other challenges brought by the database: the great variance of object sizes - from tiny logos in the background to image-filling views, perspective tilt, important rotations, color and shape changes inside the same class of objects, different occlusions and variable background.

BelgaLogos. To further validate the method, we also used the database BelgaLogos (Joly and Buisson, 2009), that was originally used for logo retrieval rather than for classification. The images are captured from sports transmissions, so most of the logos are usually on the sportswear of the players thus they have very small sizes and have even partial warping. Others are situated on the boards behind the stadium and consequently are often occluded by the players or blurred. Another challenge comes from the fact that many classes have a small number of images.

For our experiments we used only the annotated part of the database: 1937 images containing logos from 37 classes. The training/testing scheme is two-fold, class-wise.

3.3. Testing

The purpose of testing is to locate logos and classify them. Given a model for each class, the testing phase tries to match the query image against all the class models as it is shown in figure 10. The matching is done as in the training phase: using SIFT feature matching and RANSAC search for the correct homography. Since now the logos are part of natural images, with large areas of non-uniform background yielding a considerable amount of keypoints, the ratio of outliers per inliers is higher than in training phase, where we used only the logo crops. This motivates the use of a high number of iterations in the RANSAC stage.

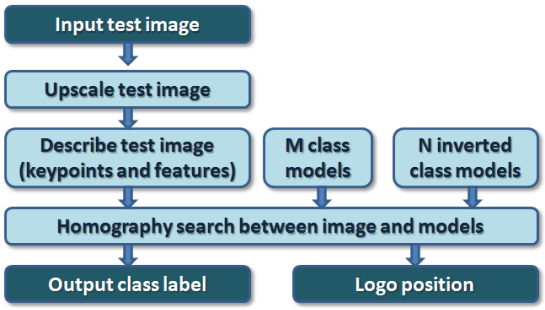


Figure 10. The schematic of the system used to locate and classify a logo.

The error map is built for each matching result and its average is used as an indicator of the quality of the image matching. If the score is small enough, the decision is taken that the logo is present. The system indicates the logo position and the homography pointing to the model of the logo class chosen. If after being confronted to all the class models, no error score is small enough, then the test image will be classified as ”no-logo”.

In the training phase we learned that small sized logos contain too few features to be correctly represented and then classified. Since there is no information about their sizes or locations, we first try to localize the logos in the images using their natural size. However, many of the test images might contain also very small logos, especially in BelgaLogos dataset. Consequently, if the detection fails, the matching is tried using an up-scaled version of the test image by the factor 4. The idea behind this up-scaling is

(a) (b)

Figure 11. (a) The failed detection of the very small logo. (b) The successful detection after resizing the test image.

Table 3. Recognition rates (i.e. localization+classification - TP) on the FlickrLogos-32 database. Lu et al. (Lu et al., 2014) report results for several techniques of acceleration: BF - brute force, FFT - Fast Fourier transform using the acceleration technique from Dubout et al. (Dubout and Fleuret, 2012) and respectively using only 4K KD trees. The proposed method includes both weighted graph and inverted colors. Results detailed on individual classes may be seen in figure 12.

Method	**Recognition Rate** [%]
Romberg et al. (2013)	61.14
Central image model	60.1
Lu et al. (2014) - BF	61
Lu et al. (2014) - FFT	61
Lu et al. (2014) - 4K	58
Li et al. (2014)	78.71
Boia & Florea (2015)	84.06
No color inversion	88.43
No logo resize	76.04
Proposed method	***90.62***

to increase the logo detectable size to the minimum training logo size; the precise value was found empirically, as a compromise between an increase of the features corresponding to the logos and an introduction of resizing artifacts. Figure 11 presents a situation of failed detection for a small logo, which the system is able to detect after applying the upscaling.

3.4. Results and Discussions

FlickrLogos-32. Testing on the FlickrLogos-32 has illustrated 100% integrated recognition rate for 18 classes, equal to or over 90% for 27 classes. The average recognition rate for the entire dataset is **90.62%**[1]. Performance for each class is presented in figure 12. The false positives rate is 0% since the detection is restricted by the error map threshold that never allows degenerate homographies. The false negatives rate is 9.38%. The Precision is 100% while Accuracy is 95.31%.

Comparative results with related work may be seen in table 3. To show the benefits of the proposed homography based construction, we considered the central image as class model as proposed in (Psyllos et al., 2010) for the vehicle logo recognition. As one may see, we outperformed the previously introduced methods with more than **10%** and the baseline method of Romberg et al. (Romberg and Lienhart, 2013) with near 30%.

The two major improvements brought to our previous work (Boia and Florea, 2015) are: (1) the usage of a graph with weighted interconnections as it gathers more points for each class description and (2) the introduction of inverted class models. In order to distinguish between the benefits introduced in the system by each of them, we present the recognition rate of the system that only contains the weighted interconnections graph improvement in table 3. It can be clearly seen that the introduction of the new model based on weighted graphs brings an increase of **4.37%** in the Accuracy of the system, while the introduction of inverted class models further improves with **2.87%**.

BelgaLogos. The total Accuracy of classification in this database is **78.09%**. To our best knowledge there is no other method reporting integrated recognition on this database, as it was originally created for logos retrieval. The individual class results are detailed in figure 13. Usually the logos in BelgaLogos dataset are very small since the images present sport images with logos on players sportswear. Illustrative detections are shown in figures 14(d),(e),(f).

The BelgaLogos dataset is an unbalanced database which for certain classes has a very small number of examples: 15 classes have less than 10 training /testing images, while "Gucci" and "Roche" hold just one image. In such cases the method gives poor results, since it cannot gain, during the training process, enough information about the logo's appearance such as to identify it in the natural test images. This fact is better seen in the plot in figure 15 which clearly shows that less than 5 training images is a clear indicator for failing, while for cases having between 5 and 10 image the results are mixed. If we remove from the testing part (but not from training as we still try to match them) the classes with less than 10 training images, the performance of the system increases to **80.33%** over 22 classes.

Again, the false positives rate is 0% since the detection is restricted by the error map threshold that never allows degenerate homographies. The false negatives rate is 21.91%, Precision is 100%, while the Accuracy is 89.04%.

Successful detections of logos in difficult situations such as

[1]Confusion matrix, and other supplementary results may be retrieved from the project page `imag.pub.ro/common/staff/rboia/logoRecognition/`.

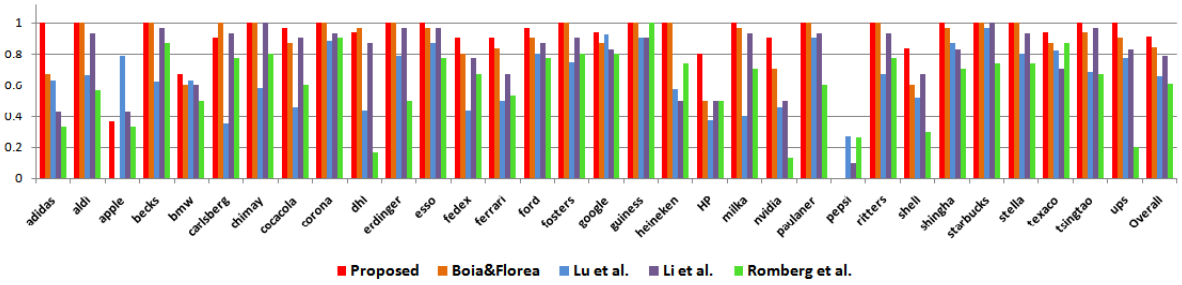


Figure 12. Comparative results on the **FlickrLogos-32** database. We report the results achieved with the proposed method, by our older method (Boia&Florea (Boia and Florea, 2015)), and by the methods introduced by Li et al. (Li et al., 2014), Lu et al. (Lu et al., 2014) and respectively Romberg et al. (Romberg and Lienhart, 2013).

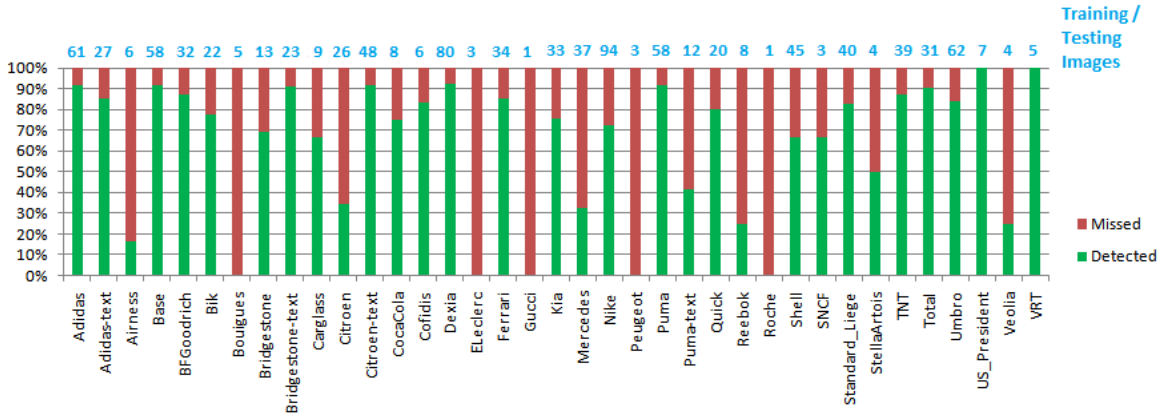


Figure 13. Results (detection rate vs missed detection) on the **BelgaLogos** database. On top of the detected/missed percentage, we marked (with blue) the number of images available in each class. One may note that poor performance is associated with classes with few number of images. This fact is shown in figure 15

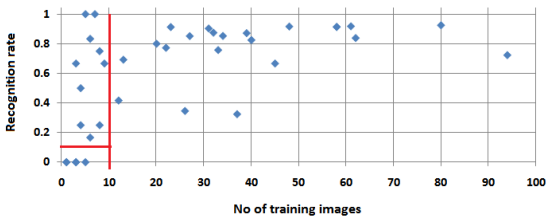


Figure 15. Recognition rate vs number of available training images in the Belga Logos database. To ensure a high recognition rate, a high number of training images is needed. In the case of very few training images, the performance of the system is unreliable.

highly occluded or very blurred logos or very small logos from the FlickrLogos-32 database are presented in figure 14(a)(b)(c).

The algorithm cannot detect logos in cases when the logo instance is very blurred and at a small scale, since the keypoints extracted in its area are not enough to represent its shape. This is the case of the failed detection in figure 16 (a) and (b). Also, in the cases when the logo is extremely small or the viewing perspective on the logo is very extreme, the detection cannot be done such as in figure 16 (c).

Multiple logos/image. The algorithm is also able to detect multiple logos in one image (showed in figure 17). If more instances of the same class are present in the same image, all the homographies output by the RANSAC matching that have a high enough number of inliers are tested with the error map average score. Those that pass the test give the localizations of the logos in the test image.

Connections in class graph. The experiments performed on the FlickrLogos-32 database reveal that the proposed method falls short for symmetric and circular logos, as the keypoints on the surface of such logos have a low dis-

Figure 14. Examples of detections: (a) with blurry and shadowed logo, (b) with occluded over the logo or (c) for a very small logo (30×30). (d) (e) (f) for very small logos in sport images

Figure 16. (a) Failed detection due to very blurred and small logo (b) Failed detection due to the very small size of the logo. (c) Failed detection due to the very sharp perspective angle of the logo instance.

tinctiveness from the others in the same shape. Thus, the classical RANSAC homography matching is unable to find correct correspondences for these types of shapes, just as for the issues of matching repeated patterns (Torii et al., 2013),(Hauagge and Snavely, 2012). In our case this problem occurs especially for the class "Pepsi", where the logo is fully round and symmetrical, leading to the inability to describe the shape and recognize it. This issue appears to a lesser degree for the "Apple" class as well, since its logo exhibits certain distinctive points that assure that the shape is not symmetrical with respect to all axes, unlike the "Pepsi" logo. The weakness of the system in detecting these two logo classes can be foreseen from the training process, when building the interconnections graph: for these classes less than 10 connections can be achieved per class, while for the rest more than 200 pairs of images that match are found. This aspect is shown in figure 18. If we removed these classes from the system, the performance increases to **95.44%**.

Logos with inverted color. Creating additional virtual classes for cases which present high variation in color (including both original and negative version) permits simultaneous detection of both, even in the same image. A visual example is shown in figure 19. Numerically for Adidas (FlickrLogo-32) the recognition rate is 100%, Adidas (BelgaLogos)–91.8%, HP(FlickrLogo-32) 80% and Nike (BelgaLogos) - 72.34%.

4. Conclusion

We have proposed an effective method for logo localization and recognition. It works for the majority of the tested cases, surpassing many challenges of logo recognition in natural images. The achieved results improve with more than 10% the best previous work and with 6% compared to our older work. Continuation envisages the cases of failure (identified by the lack of connection in graphs) by changing the matching process so to take into account the vicinity of

Figure 17. Detections of logo instances from 3 different classes in the same image.

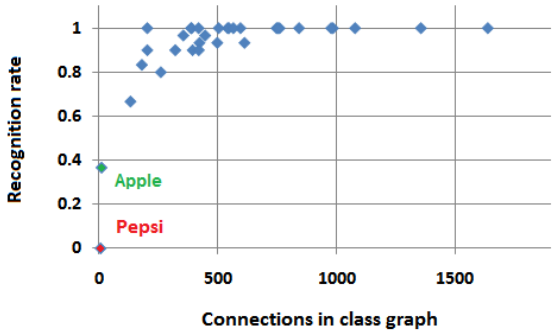


Figure 18. Recognition rate vs connections in class graph in the training phase for FlickLogo-32 database. The proposed method under-performs for Pepsi and Apple classes which exhibit very few interconnections graph.

the points, thus improving the homography building.

Acknowledgment

This work was supported by the Romanian Sectoral Operational Programme Human Resources Development 2007-2013 through the European Social Fund Financial Agreements POSDRU/159/1.5/S/132395 and POSDRU /159/1.5/S/134398.

References

Bagdanov, A., Ballan, L., Bertini, M. and Del Bimbo, A. (2007). Trademark matching and retrieval in sports video databases, *ACM MIR*, pp. 79–86.

Boia, R. and Florea, C. (2015). Homographic class template for logo localization and recognition, *Proc. of IbPRIA*.

Brown, M. and Lowe, D. (2003). Recognising panoramas, *ICCV*, pp. 1218–1225.

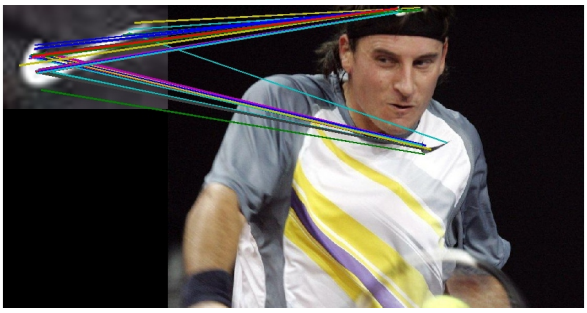


Figure 19. Successful detection of the logo in the normal luminance model and in the inverted one.

Brown, M. and Lowe, D. (2007). Automatic panoramic image stitching using invariant features, *Int. J. of Computer Vision* **74(1)**: 59–73.

Chan, D., Ge, R., Gershony, O., Hesterberg, T. and Lambert, D. (2010). Evaluating online ad campaigns in a pipeline: causal models at scale, *ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, pp. 7–16.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection, *CVPR*, pp. 886 – 893.

Dorko, G. and Schmid, C. (2005). Object class recognition using discriminative local features, *Technical report*, INRIA.

Dubout, C. and Fleuret, F. (2012). Exact acceleration of linear object detectors, *ECCV*, pp. 310 – 311.

Dubrofsky, E. (2009). *Homography estimation*, Master's thesis, Carleton University.

Everingham, M., Van Gool, L., Williams, C., Winn, J. and Zisserman, A. (2010). The pascal visual object classes (voc) challenge, *Int. J. of Computer Vision* **1**: 303–338.

Fergus, R., Perona, P. and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning, *CVPR*, pp. 264–271.

Fischler, M. and Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography., *Communications of the ACM* **24**: 381–395.

Florea, L., Florea, C., Vranceanu, R. and Vertan, C. (2013). Can your eyes tell me how you think? a gaze directed estimation of the mental activity, *BMVC*.

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*, Cambridge University Press.

Hauagge, D. C. and Snavely, N. (2012). Image matching using local symmetry features, *CVPR*, pp. 206–213.

Jin, Y., Tao, L., Di, H., Rao, N. and Xu, G. (2008). Background modeling from a free-moving camera by multi-layer homography algorithm, *ICIP*, pp. 1572–1575.

Joly, A. and Buisson, O. (2009). Logo retrieval with a contrario visual query expansion, *ACM MM*, pp. 581–584.

Kleban, J., Xie, X. and Ma, W.-Y. (2008). Spatial pyramid mining for logo detection in natural scenes, *IEEE ICME*, pp. 1470 – 1477.

Klein, G. and Murray, D. (2009). Parallel tracking and mapping on a camera phone, *Int. Symp. on Mixed and Augmented Reality*, pp. 83–86.

Krapac, J., Perronnin, F., Furon, T. and Jegou, H. (2014). Instance classification with prototype selection, *ACM ICMR*, pp. 431 – 4.

Kumar, S. and Hebert, M. (2004). Discriminative fields for modeling spatial dependencies in natural images, *NIPS*, pp. 1531–1538.

Lewis, R., Rao, J. and Reiley, D. (2014). Measuring the effects of advertising: The digital frontier, *Economics of Digitization*, pp. 1–5.

Li, K., Chen, S., Su, S., Duh, D., Zhang, H. and Li, S. (2014). Logo detection with extendibility and discrimination, *Multimed Tools Appl.* **72**: 1285 – 1230.

Lin, Y. and Medioni, G. (2007). Map-enhanced uav image sequence registration and synchronization of multiple image sequences, *CVPR*, pp. 1–7.

Lowe, D. (1999). Object recognition from local scale-invariant features, *ICCV*, pp. 1150–1157.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints, *IJCV* **62(2)**: 91 – 110.

Lu, V., Endres, I., Stroila, M. and Hart, J. (2014). Accelerating arrays of linear classifiers using approximate range queries, *IEEE WACV*, pp. 255–260.

McLauchlan, P. F. and Jaenicke, A. (2002). Image mosaicing using sequential bundle adjustment, *Image and Vision Computing* pp. 751–759.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T. and Gool, L. V. (2005). A comparison of affine region detectors, *IJCV* **62(1-2)**.

Ng, A. and Jordan, M. I. (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes, *NIPS*, pp. 841–849.

Opelt, A., Fusseneger, M., Pinz, A. and Auer, P. (2006). Generic object recognition with boosting, *IEEE T. PAMI* **28(3)**: 416 – 431.

Psyllos, A. P., Anagnostopoulos, C. N. E. and Kayafas, E. (2010). Vehicle logo recognition using a sift-based enhanced matching scheme, *IEEE Trans. Intel. Transp. Syst.* **11(2)**: 322 – 328.

Raguram, R., Frahm, J.-M. and Pollefeys, M. (2008). A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus, *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, Berlin, Heidelberg, pp. 500–513.

Revaud, J., Douze, M. and Schmid, C. (2012). Correlation-based burstiness for logo retrieval, *ACM MM*, pp. 965–968.

Ries, C., Richter, F., Romberg, S. and Lienhart, R. (2014). Automatic object annotation from weakly labeled data with latent structured svm, *CBMI*, pp. 1–4.

Romberg, S., Garcia Pueyo, L., Lienhart, R. and van Zwol, R. (2011). Scalable logo recognition in real-world images, *ACM ICMR*, pp. 965–968.

Romberg, S. and Lienhart, R. (2013). Bundle min-hashing for logo recognition, *ACM ICMR*.

Schneiderman, H. and Kanade, T. (2003). A statistical method for 3d ob-ject detection applied to faces and cars, *CVPR*, pp. 746–751.

Sheikh, Y., Javed, O. and Kanade, T. (2009). Background subtraction for freely moving cameras, *ICCV*, pp. 1219–1225.

Simon, G., Fitzgibbon, A. and Zisserman, A. (2000). Markerless tracking using planar structures in the scene, *Augmented Reality*, pp. 120–128.

Snavely, N., Seitz, S. M. and Szeliski, R. (2008). Modeling the world from internet photo collections, *Int. J. of Computer Vision* pp. 189–210.

Torii, A., Sivic, J., Pajdla, T. and Okutomi, M. (2013). Visual place recognition with repetitive structures, *CVPR*.

Torralba, A., Murphy, K. and Freeman, W. (2004). Sharing visual fea-tures for multiclass and multiview object detection, *CVPR*, pp. 762–769.

Vedaldi, A. and Fulkerson, B. (2010). Vlfeat: An open and portable library of computer vision algorithms, *ACM MM*, pp. 1469–1472.

Viola, P. and Jones, M. (2004). Rapid object detection using a boosted cascade of simple features, *CVPR*, Vol. 1, pp. 511–518.

Zhu, G. and Doermann, D. (2007). Automatic document logo detection., *ICDAR*, pp. 864–868.