LOGO RECOGNITION AND LOCALIZATION IN REAL-WORLD IMAGES BY USING VISUAL PATTERNS

Wei-Ta Chu and Tsung-Che Lin

National Chung Cheng University, Chiayi, Taiwan
wtchu@cs.ccu.edu.tw, congjhe@gmail.com

ABSTRACT

By describing spatial relationships between feature points, we present promising logo recognition and localization, which are verified based on two state-of-the-art datasets. Given features points on the query logo, similar features on test images are efficiently found by locality sensitive hashing. After filtering out outliers, candidate regions are found by the mean-sift algorithm, and each region is compared with the logo by jointly considering visual word histogram and visual patterns. Evaluation results show that visual patterns more appropriately describe logos and provide better performance than previous approaches.

Index Terms— Logo recognition, logo localization, visual patterns

1. INTRODUCTION

Logos are important symbols for organizations and business. They can be seen everywhere, such as sports games, TV series, and websites. If we can detect logo objects from media, many interesting applications can be developed. Many studies have been proposed to detect logos in document images [1]. However, logo objects in real-world images are often suffered from non-rigid deformation (on T-shirts or shoes) or reflection (on bottles or cars), which makes logo detection in real-world images much more challenging than that for document images.

Joly and Buisson [2] propose one of the first real-world image collections specific for logo detection. They also propose a SIFT-based matching approach [3] with a query expansion strategy to retrieve images with specific logos. However, their work solely utilizes local features and does not achieve satisfactory performance. Furthermore, they do not localize logos, without this many practical applications cannot be developed.

In this paper we propose a logo recognition method based on visual pattern matching. Visual patterns are constituted by local features with description of spatial relationship between them. To speed up process, locality-sensitive hashing and candidate logo region selection are adopted. With the proposed methods, we not only retrieve images containing specific logos, and also localize the detected logos in images. We conduct comprehensive experiments based on two recently-proposed datasets: the BelgaLogos dataset [2] and the FlickrLogos dataset [4].

In the following, Section 2 describes details of the proposed logo detection and localization system. Section 3 provides comprehensive evaluation results, and Section 4 concludes this paper.

2. LOGO DETECTION AND LOCALIZATION

2.1 Overview

Figure 1 shows the proposed system framework. We first extract SIFT features from test images and the logo image. Similar features between them are found by using the Locality Sensitive Hashing (LSH) [5]. Outlier test images are then detected and filtered out to speed up computation. According to candidate features in test images, we find regions that contain high-density candidate features, and from which we decide whether they contain the logo by matching bag of visual words and visual patterns.

We extract SIFT features from an image I, and represent this image as $I = \{p_i\}$. Each SIFT feature p_i [3] is represented by a 4-tuple $p_i = [\boldsymbol{x}_i, s_i, \theta_i, \boldsymbol{f}_i]$, where the 2D vector $\boldsymbol{x}_i$ represents this keypoint's location, s_i is scale of this keypoint, $\theta_i \in [-\pi, +\pi]$ is the main orientation of the neighborhood of this keypoint, and $\boldsymbol{f}_i$ is the 128-dim descriptor describing its appearance information.

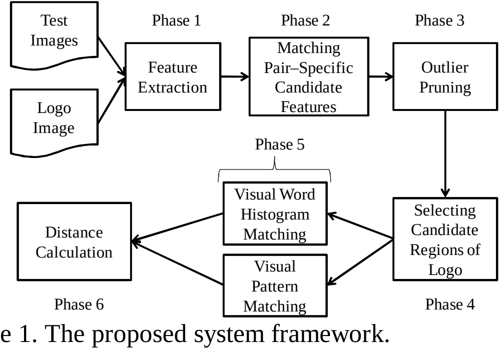


Figure 1. The proposed system framework.

2.2 Matching Pair-Specific Candidate Features

To find similar objects between a test image and the logo image, we transform this problem as an approximate nearest neighbor search problem. Based on the extracted feature points, this problem is processed by locality-sensitive hash

(LSH) [5]. The basic idea of LSH is to hash input items so that similar items are mapped to the same buckets with high probability. With the E2LSH (Exact Euclidean LSH) package [5], we set a parameter ϵ to find approximate local features ($\epsilon = 250$ in this work). For each local feature in the logo image, we find its ϵ-nearest neighbors in test images. Because E2LSH adaptively learns feature distributions between an image pair, conditions for finding the ϵ-nearest neighbor vary for different image pairs. This is why we call these features "pair-specific" candidate features.

After this process, we have a large number of pair-specific candidate features. To further speed up computation, we filter out the test images containing too few candidate features. If a test image contains less than $\gamma\% \times N$ pair-specific candidate features, it is claimed as an outlier. The parameter γ is set as 0.1, and N is the total number of local features in the logo image.

2.3 Candidate Region Selection

The logo object may appear anywhere in test images, and is often much smaller than the whole image. In this component, we cluster neighboring candidate features into regions where the logo object may locate. Based on xy coordinates of feature points, the mean-shift algorithm [6] is used to implement clustering. This algorithm locates the maxima of a density function given discrete data points. With this algorithm, we do not need to decide the targeted number of clusters in advance.

Assume that n clusters are generated after the mean-shift algorithm, denoted by $\mathcal{C} = \{C_1, C_2, ..., C_n\}$. We design an adaptive clusters selection algorithm (c.f. Figure 2) to select clusters according to their relative sizes. In this algorithm, two thresholds T_1 and T_2 are adaptively updated to determine relatively larger clusters. When the ratio of T_2 to T_1 is less than a threshold τ, this algorithm converges ($\tau = 1.5$ in this work). This algorithm selects appropriate candidate regions that contain more pair-specific candidate features and may contain the logo object.

```
Algorithm: Adaptive clusters selection
Input: The set of clusters $\mathcal{C} = \{C_1, C_2, ..., C_n\}$
1.  Threshold setting: $T_1 \leftarrow \sum_{C_i \in \mathcal{C}} \frac{|C_i|}{|\mathcal{C}|}$
2.  Discard $C_j$ from $\mathcal{C}$ if $|C_j| < 0.9 \times T_1$
3.  Update threshold: $T_2 \leftarrow \sum_{C_i \in \mathcal{C}} \frac{|C_i|}{|\mathcal{C}|}$
4.  Repeat
5.      If $\frac{T_2}{T_1} > \tau$, $T_1 \leftarrow T_2$ and goto step 2
6.      Else goto to step 8
7.  Until $\mathcal{C} = \emptyset$
8.  Output $\mathcal{C}$
```

Figure 2. The adaptive clusters selection algorithm.

2.4 Visual Word Histogram and Visual Pattern

To verify whether a candidate region contains a specific logo, we describe candidate regions by a visual word histogram and visual patterns.

2.4.1 Visual Word Histogram

We build up a standard bag-of word representation by clustering local features extracted from a set of training images [7]. The centroids of clusters form a visual dictionary $W = \{w_1, w_2, ..., w_K\}$. Given a set of local features $P = \{p_i\}$ extracted from a candidate region or the logo image, we consult the visual dictionary to identify each local feature's corresponding visual word by finding the nearest visual word w_{i^*} to a local feature p_i. After this step, the feature set P is transformed into bag-of-words representation. We count numbers of visual words and form corresponding visual word histograms after normalization.

2.4.2 Visual Patterns

- Description

Visual word histogram is a global representation that states statistics of visual word appearance and overlooks spatial information. In this work we also describe spatial relationship between feature points as visual patterns to increase robustness of detection results.

The spatial relationship between two local features p_i and p_j is characterized by $r_{ij} = [D_{ij}, S_{ij}, H_{ij}, H_{ji}]$, where

$$D_{ij} = \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\sqrt{s_i^2 + s_j^2}}, \tag{1}$$

$$S_{ij} = \frac{\min(s_i, s_j)}{\max(s_i, s_j)}, \tag{2}$$

$$H_{ij} = \Delta_\theta(\arctan(\boldsymbol{x}_i - \boldsymbol{x}_j) - \theta_i), \tag{3}$$

$$H_{ji} = \Delta_\theta(\arctan(\boldsymbol{x}_j - \boldsymbol{x}_i) - \theta_j). \tag{4}$$

The value D_{ij} denotes the normalized spatial distance between p_i and p_j , which is normalized by the corresponding scale to resist image scaling. The notation $\|\cdot\|$ denotes Euclidean distance. The value S_{ij} denotes the relative scale. The value H_{ij} is the relative heading from p_i to p_j, which makes it invariant to image rotation. The value H_{ji} is the relative heading from p_j to p_i. An example of relative headings is illustrated in Figure 3. The function $\Delta_\theta(\cdot)$ denotes the principle value, which is the value in the range $[-\pi, \pi]$. This representation is invariant to translation, scale and rotation, and is robust to small distortion.

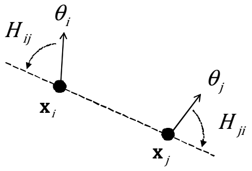


Figure 3. An illustrative example of relative headings.

We compare two relationships r_{ij} and $r_{i'j'}$ by their quantized heading values. Given a spatial relationship r_{ij}, its two heading values are quantized into a pair of specific indices by using the quantization function. That is,

$$QH(r_{ij}) = \left[\frac{conv(H_{ij})}{2\pi/NUMBINS}, \frac{conv(H_{ji})}{2\pi/NUMBINS}\right], \tag{5}$$

where the function $conv(\cdot)$ converts a principle value ranging $[-\pi, \pi]$ to $[0, 2\pi]$, and the constant $NUMBINS$ denotes the number of bins to quantize the interval $[0, 2\pi]$. After quantization, two relationships r_{ij} and $r_{i'j'}$ are considered consistent if $QH(r_{ij}) = QH(r_{i'j'})$.

- Visual pattern discovery

We adopt data mining techniques to unsupervisedly find visual patterns. By using a graph to represent image features and their spatial relationships, a visual pattern can be treated as a connected subgraph embedded in this graph.

Given a set of local features $P = \{p_i\}$, we build an undirected graph (called the root graph) $G_r = \langle V_r, E_r \rangle$. The vertex $v_i \in V_r$ corresponds to p_i, and is represented by a 2-tuple $v_i = [i, \ell_i]$, where ℓ_i is the vertex label indicating the visual word index for p_i. Each edge is represented by a three-tuple $e_{ij} = [i, j, \ell_{ij}]$, where ℓ_{ij} is the edge label determined by $QH(\cdot)$. We have $QH(r_{ij}) = QH(r_{i'j'}) \leftrightarrow \ell_{ij} = \ell_{i'j'}$.

Given a root graph G_r, any connected subgraph would potentially be a visual pattern. To decrease complexity of the mining process, some criteria are applied to construct appropriate edges. First, a pattern cannot have two vertices with the same visual word index. Second, we assume that the spatial scatter of a pattern would be in a range. Two features that are highly overlapped or far apart can't be linked by an edge. Third, we should consider a pattern's repeatability across different images. Only features in nearby scales are connected. Overall, we construct an edge between two vertices with different vertex labels, if their spatial relationship fulfills the following equation.

$$E_r = \{e_{ij} | \ell_i \neq \ell_j, D_{ij} \in [T_{D_{min}}, T_{D_{max}}], S_{ij} > T_{S_{min}}\}. \quad (6)$$

The values $T_{D_{min}}$ and $T_{D_{max}}$ are thresholds for D_{ij} (eqn. (1)), and the value $T_{S_{min}}$ is the threshold for S_{ij} (eqn. (2)).

Given a root graph $G_r = \langle V_r, E_r \rangle$ constructed from a candidate region or from the logo image, we exhaustively find subgraphs in it. With the constraints mentioned above, the root graphs from candidate regions are much smaller than the graph from the whole image. Furthermore, in this work we constrain that the order of a pattern (i.e. the number of vertices) should be three. Note that a larger-size visual pattern has more discriminability but less repeatability across images. Therefore, too large patterns cannot serve as good models to detect visual patterns across images.

A set of subgraphs are finally obtained from each root graph and serve as the representation of visual patterns. To compare any two subgraphs, we encode a subgraph into a string code called the canonical label. Two subgraphs are isomorphic and called matched if they have identical canonical label. Given a subgraph, its canonical label is obtained by concatenating all its vertex labels and the upper-triangular entries of its adjacency matrix. In order to make this string invariant to vertex ordering, a naïve way is to try all possible permutations of vertices, produce a set of strings from all such permutations and its corresponding adjacency matrix, and then choose the lexicographically largest one as the canonical label for this subgraph [8]. Efficient implementation of this process please refers to [8].

2.5 Distance Calculation

We jointly consider visual word histograms and visual patterns to more appropriately measure distance between a candidate region and the logo. Based on visual word histograms, we define a distance D_{vw} as

$$D_{vw} = \sqrt{\sum_{k=1}^{n} (h_i[k] - h_j[k])^2}, \quad (7)$$

where h_i and h_j denote the visual word histogram of the logo and a candidate region, respectively. The value n denotes numbers of visual words.

Based on visual patterns, the distance D_{vp} between a logo i and a candidate region j is calculated as

$$D_{vp} = \frac{1}{m(i,j) + \rho}, \quad (8)$$

where $m(i,j)$ denotes the number of matched visual patterns, and ρ is set as 1 to avoid zero denominator. If more visual patterns are matched, the candidate region j is more similar to the logo i.

We jointly consider two clues to decide whether a logo exists in a test image. A weighting α is set to prioritize two measures mentioned above. The integrated measure is

$$D_{all} = \alpha \times D_{vw} + (1 - \alpha) \times D_{vp}. \quad (9)$$

If the integrated distance is less than a threshold φ, the logo is claimed to be detected in the candidate region (φ = 0.3 in this work).

3. EXPERIMENTS

We evaluate the system on the BelgaLogos dataset [2], which includes 10,000 images and 26 different logos. Each image can contain one or several logos or no logo at all. There are totally 22,572,764 SIFT features in the dataset. To extract visual word histograms and visual patterns, we construct a size-50 visual vocabulary. The size-50 vocabulary is constructed based on 1% of total SIFT feature points in the dataset. For visual pattern discovery, the parameter $NUMBINS$ is set as 8, and the thresholds $T_{D_{min}}$, $T_{D_{max}}$, and $T_{S_{min}}$ are set as 2, 30, and 0.6, respectively.

- Performance of logo detection

We use precision and recall to demonstrate detection performance. If multiple logos are found in the same image, we only count them once for calculating precision and recall. Table 1 shows evaluation results of our system and [2], while they only provide precision values. Only average results are shown for the 26 logos due to space limitation. From Table 1 we clearly see the superior precision value for our work. Our system works badly for some logos, such as Addidas, CocaCola, Ecusson, and Nike. Our work is built based on local features, and if we do not extract enough local features from a query logo image, our system has little ability to detect logos.

We also compare with the works in [9] (first round result) and [10]. Only six logo detection results were specially reported in their results, as shown in Table 2. We again can find that the proposed approach is better than other methods. Our approach not only utilizes features similarity but also considers spatial relationship between features, while other approaches only use feature similarity. If enough local features can be extracted from a query logo, we can clearly describe logo characteristics by using visual patterns, such as President and Dexia.

We also test our system based on the FlickrLogos dataset and compare it with Romberg's work [4]. Overall, we obtain 0.58 in precision, and Romberg et al. obtained 0.61 in precision. However, we do not need to individually train a model for each logo. Instead, simply one query logo is used for logo recognition. More detailed experimental results would be provided in the future due to space limitation.

Table 1. Detection results of our system and [2].

	Joly [2]	Our Precision	Our Recall
Average	0.257	**0.300**	0.190

Table 2. Detection results of our system, [9], and [10].

		Coca Cola	Dexia	Ferrari	Mercedes	Peugeot	President
[10]	P	0.00	0.43	0.02	0.25	0.00	0.09
	R	0.00	0.02	0.03	0.09	0.00	0.64
[9]	P	0.00	0.81	0.01	0.92	0.01	0.05
	R	0.00	0.03	0.01	**0.15**	0.17	0.36
Ours	P	0.00	**0.90**	**1.00**	**1.00**	**0.05**	**0.67**
	R	0.00	**0.48**	**0.01**	0.04	**0.40**	**0.85**

Table 3. Average overlap ratio for different logos.

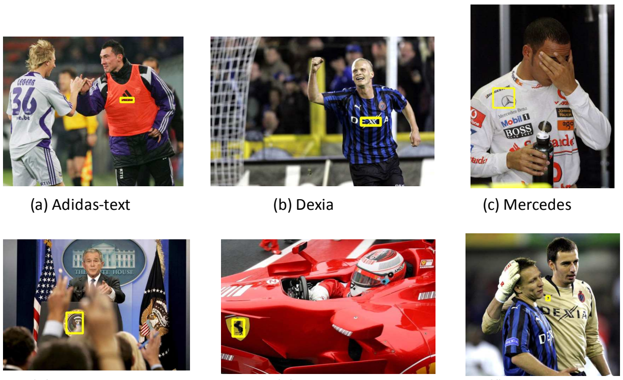
Logo name	**Average**	**Logo name**	**Average**
Adidas-text	0.921	Peugeot	1.000
Base	0.889	President	0.973
Citroen	0.946	Puma	0.852
Cofidis	0.900	Puma-text	1.000
Dexia	0.928	Quick	0.836
Eleclerc	0.759	SNCF	0.671
Ferrari	1.000	Stella	0.800
Kia	0.888	VRT	0.685
Mercedes	0.669		
Overall	0.866		

- Performance of logo localization

If a candidate region is claimed to contain logos, we determine positions of logos by locations of candidate regions. In the BelgaLogos dataset, logos are usually smaller relative to the whole image, and locations of logos in images are not well defined. Therefore, we manually define coordinates of minimum bounding boxes of the detected logos for the localization experiment. The ratio of the overlap area to the detected region is calculated:

$$\text{overlap ratio} = \frac{\text{overlap area}}{\text{detected area}}. \tag{10}$$

Table 3 shows localization results for images retrieved by our system. The average overlap ratio is 0.866. Some examples of logo localization are shown in Figure 4.



(a) Adidas-text (b) Dexia (c) Mercedes

(d) President (e) Ferrari (f) Puma

Figure 4. Example results of the detected logos.

4. CONCLUSION

We have presented an approach to automatically detect and localize logo objects in images. The pair-specific concept facilitates us to only capture relevant features between a query logo and a test image. Candidate regions are found by using the mean-shift method, and visual word histograms and visual patterns jointly describe logo objects. Our system works well for two large-scale databases (i.e., BelgaLogos and FlickrLogos). In the future, more comprehensive experiments will be conducted and new features will be designed to avoid bad performance for some logos.

Acknowledgement

The work was partially supported by the National Science Council of Taiwan, Republic of China under research contract NSC 100-2221-E-194-061.

5. REFERENCES

[1] Z. Li, M.S. Austum, and M. Neschen, "Fast logo detection and recognition in document images," *Proc. of ICPR*, pp. 2716-2719, 2010.

[2] A. Joly and O. Buisson, "Logo retrieval with a contrario visual query expansion," *Proc. of ACM Multimedia*, pp. 581-584, 2010.

[3] D.G. Lowe, "Object recognition from local scale-invariant features," *Proc. of ICCV*, pp. 1150-1157, 1999.

[4] S. Romberg, L.G. Pueyo, R. Lienhart, and R.V. Zwol, "Scalable logo recognition in real-world images," *Proc. of ICMR*, 2011.

[5] M. Datar, N. Immorlica, P. Indyk, and V. Mirroknu, "Locality-sensitive hashing scheme based on P-stable distributions," *Proc. of Annual Symposium on Computational Geometry*, pp. 253-262, 2004.

[6] K. Fukunarga and L. D. Hostetler. "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on information Theory*, vol. 21, no. 1, pp. 32-40, 1975.

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

[8] M. Kuramochi and G. Karypis, "An efficient algorithm for discovering frequent subgraphs," Technical Report, Department of Computer Science, University of Minnesota, 2002.

[9] J. Meng, J. Yuan, Y. Jiang, N. Narasimhan, V. Vasudevan, and Y. Wu, "Interactive visual object search through mutual information maximization," *Proc. of ACM Multimedia*, pp. 1147-1150, 2010.

[10] C. H. Lampert, "Detecting objects in large image collections and videos by efficient subimage retrieval," *Proc. of ICCV*, pp. 987-994, 2009.