Effective Logo Retrieval with Adaptive Local Feature Selection

Jianlong Fu, Jinqiao Wang, and Hanqing Lu
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
jasonfu1988@gmail.com, {jqwang,luhq}@nlpr.ia.ac.cn

ABSTRACT
Towards building a practical large-scale logo retrieval system, we propose a novel approach to extract and combine local features for effective logo retrieval. Instead of global feature extraction by modeling the web logo as a whole, we extract the local feature phrases to form a visual codebook and build an inverted file storing the features to accelerate the indexing process. Then we divide logos into several groups according to local feature type based on which feature can model the logo best and naming as “Point-type”, “Shape-type” and “Patch-type”. We develop a strategy of adaptive feature selection by a weight updating mechanism. To evaluate the performance, we have built a new challenging dataset which consists of 60 international corporations’ logos. Experiments and comparisons demonstrate the superior performance to previous retrieval algorithms.

Categories and Subject Descriptors
H.3.3 [Information Systems]: Information storage and retrieval

General Terms
Algorithms, Experimentation

Keywords
Logo Retrieval, Adaptive Feature Selection

1. INTRODUCTION
Content based logo retrieval is of high interest for many applications in daily life including logos detection in TV or other media, logo retrieval and automatic logo annotation. The main challenge for logo retrieval comes from the fact that logo images are usually very small which providing us with very little information. Although the bag-of-features based approaches are more popular, in our dataset, we discovered that there are only 1 or 2 feature points and even no feature points appear in some trademarks. Moreover, the combinations of scale, rotation, viewpoint, lighting, and many other factors also contribute to the challenge of this task.

To achieve robust object recognition from videos and images, a series of approaches [1-3] have been proposed aiming at exploit logos or street landmarks appearance and geometrical information. Since appearance and bag-of-features are surprisingly useful in general object recognition [4], it is reasonable for us to adopt both appearance and context information of logos to modeling. However, we believe that we don’t have to exploit all the information extracted from logos. Considering that each kind of brand has its unique features combination, in this paper, we are interested in the strategy of local feature selection for different type of logos in the web logo search task and we will describe it explicitly in the next section.

2. PROPOSED APPROACH
Based on our observation, we extract SIFT features, shape context and segmentation patches, cluster and build an inverted file for each dataset logo image at the first step offline. Then when a query image comes, we launch an initial retrieval by extracting the three types of features and assign them an identical weight. At the second step, a mechanism of result refinement is introduced to...
update the weight continuously for the next retrieval before we get the final results, as shown in Figure 1.

3. Local Feature Extraction
Local features are fundamental representation units of an object in images. We extract local key points, edges and patches. We utilize SIFT feature to represent local key points for its effectiveness and robustness. To exploit the shape information, sketches between spatial related strokes [5] and shape context method [6] are employed. Shape context has been used in recognition of conformation codes and handwritten digits. For its strength of outstanding shape representation, shape context has shown great potential in shape recognition. Each shape context is a log-polar histogram of the coordinates of the rest of the point set measured using the reference point as the origin. To obtain the sampled edge points, we firstly choose Canny edge detector [7] to extract edges. As patch feature, we choose the most popular segmentation algorithm, Mean-Shift [8], to get similar subsections of an image and calculating the gray histogram for each segment.

3.1 Local Feature based Logo Analysis
Web logos involve a massive dataset, in which we can divide hundreds of trademarks into several groups according to the type of feature which can model the logo best and naming as “Point-type”, “Shape-type” and “Patch-type”. Then combining different types in a suitable way, we will achieve more robustness with less effort. That is why we utilize the features combination strategy. Though SIFT key points have been demonstrated to be effective for matching problems, just one type of feature usually isn’t adequate to model a whole logo. Fortunately, we have found shape and appearance features can tolerate more scale variations. We are tempted to exploit robust feature combinations, bigram or triplet, which are unique to the logo.

Figure 2: Different types of logos.
Figure 2 shows three types of features for logo images. For some kinds of logos, such as Adidas, Canon and Hp, whose main component is English letters, we can’t extract sufficient SIFT key points from them, whereas shape context descriptors may work well in this situation. Similarly, the subsection gray histogram is a necessary complement. For logos in the second line, to achieve against viewpoints, scale and illumination changes, we combine the edges or appearance features with SIFT key points. The three types of features for McDonald’s are shown in Figure 3.

Figure 3: Three types of local features for McDonald’s Logo. (a) an image with McDonald’s logo; (b) SIFT key points; (c) Canny edges; (d) image segmentation patches.

From Figure 3, we display the three aspect of McDonald logo. We can extract the segmentation patches with gray level information combining with either SIFT or shape context descriptors for modeling the logo. In practice, we extract the three types of features for each logo, and use an adaptive weighting strategy in the web logo retrieval.

4. LOGO RETRIEVAL
4.1 Building Visual Vocabulary
For each image in our dataset, we apply the procedure to extract the SIFT descriptors, shape context and subsections’ gray histogram in the training stage. And then we cluster the three types of features and construct vocabularies for each type to obtain the visual codebook. In the inverted structure, each image is represented by a vector of standard weighting known as tf-idf. For a visual vocabulary of k words, each image is represented by a vector $V_d = (t_1, t_2, ..., t_k)^T$ of weighted word frequencies with components. The weight is calculated as follows:

$$t_i = \frac{n_{id}}{n_d} \log \frac{N}{n_i}$$

in which $n_{id}$ is the number of occurrences of word $i$ in the image $d$, $n_d$ is the total number of words in the image, $n_i$ is the number of occurrences of terms $i$ in the whole database and $N$ is the number of images in the whole database. The weight is a product of two terms: the word frequency and the inverse document frequency. This index structure can reflect in intuition that unique feature phrases occur on these kinds of logos, but rarely occur anywhere else.

4.2 Logo Retrieval
At this stage, images are ranked by their scalar product between the query vector and all document vectors in the database. Since we have combined different types of features into phrases in the feature selection stage, a combined score can be introduced to represent the similarity between each image in our dataset and the query image. For each image $d$ in the dataset, we define a combination score as follows:

$$S(d) = \sum_{m} w_m \text{Score}(f_m) + \sum_{(f_j, f_k) \in p} d_{f_j}(l_j, l_k)$$

where $f_j$ specifies the candidate type of matching feature and $w_j$ is a coefficient related to weight. Candidate feature type includes SIFT key points, shape context and segmentation patches information. $n=2$ for a bigram and $n=3$ for a triplet. $(f_j, f_k) \in p$ indicates $f_j$ and $f_k$ are combined into a feature phrase. We do not
use the scalar product as the score directly, and instead we normalize it for each feature type before we combine the different types into a group. To increase the ranking for images where the searched for feature appear close together in the retrieved images, a spatial consistency is introduced. Because we believe similar subparts of images should have similar spatial arrangement. The idea is implemented here by adding a geometric consistency score. For $d_k(l_i,l_j)$, we choose SIFT key points as a reference point for its stability with its matched position $l_i$. Then define $d_k(l_i,l_j)$ as the $L_2$ distance from $l_i$ to $l_j$. For triplet, this term is sum of the distance between the three features.

All returned images are ranked in decreasing order of the similar scores we have computed yet. Since we concentrate on a practical logo retrieval system, we want to rank the desired images in front as much as possible, especially the first 20 results.

### 4.3 Adaptive Weights Update for Result Refinement

We will highlight in this subsection the novel idea of adaptive weights $w_i$ which is used to compute the weights for each type of features. We find that if we give the three types of features an identical weight in the first iteration, the first 5 to 10 retrieved images are usually correct in a large database. We thus define the weights by the contributions of each type features in the first $n$ retrieval results under the hypothesis that they are all correct results. For each type of feature, we make statistics the number of images that whether we adopt this feature type only or combined feature types, they both appears in the first $n$ positions. We believe the appearance of more images indicates this type of feature make a greater contribution to the retrieval results. That is to say, this type of feature can model this kind of logo pretty well. $w_i$ is defined as follows:

$$w_i = \frac{\# \{ \{ I_j, I_j \in \Omega_n \} \cap \{ I_j, I_j \in \Omega_{n,i} \} \} }{ \sum_{j=1}^{n} \# \{ \{ I_j, I_j \in \Omega_n \} \cap \{ I_j, I_j \in \Omega_{n,i} \} \} }$$

(3)

$I_j$ is the image in the dataset of $N$ images, $j \in \{1,\ldots,N\}$

$\Omega_n$ is the first $n$ images ranking by combined feature types

$\Omega_{i,n}$ is the first $n$ images ranking by the $i$th feature type, $i=1,2,3$

After the first iteration, we have got updated weight for each. Then following the procedure we described to execute the next iteration.

### 5. EXPERIMENTS

#### 5.1 Our Dataset

Our logo dataset is composed of 12,000 images covering natural scenes, animals, advertisements, automobiles, politics, economics and all aspects of human activities. All images are in JPEG format and have been resized properly. The 12,000 images in our dataset have been manually annotated for 60 logos of different brand names. Each image is labeled for each logo with 1 if the trademark is present in the image and with 0 if it is not. The list of logos that were annotated is given in table 1 with an illustration of the targeted logo.

#### 5.2 Features Combination Test

In our experiments, we extract local key points, Canny edges and segmentation patches. The parameter of the scales' number per octave and DoG threshold is set according to David Lowe's paper. We choose Canny edge detector to extract edges and collect sampled points to compute shape context descriptor. Shape context descriptors provide us millions of feature points for about 100 images. In practice, we cannot afford the heavy computational process. Edges whose length is greater than 10 pixels and smaller than 500 pixels, area is greater than 25 square pixels and smaller than 160000 square pixels are accepted. By trimming the edges, we can achieve efficiency with limited time consuming. For the Mean-shift Segmentation algorithm, we set the related parameters as follows: $\text{spatial\_band} = 7$ and $\text{color\_band} = 14$.

![Table 1: Annotated logos in our dataset.](image)

To explain our feature combination strategy outperforms the method using one type of feature only, we firstly apply SIFT, shape context, Mean-Shift segmentation to our dataset. Then we use our novel algorithm evaluating with and without adaptive weight method. Non-adaptive weight means the three terms has an identical weight. We select 26 logos as the query image from 60 annotated logos computing the Mean Average Precision for each query and their average. Table 2 then gives the Mean Average Precision for the single type of feature and for the two combination strategy with and without adaptive weight method of the 26 logos.

The results show that the MAP of our combination strategies with and without adaptive weights are higher than any method using alone with up to 10.8% MAP improvements for SIFT, 7.0% MAP improvements for shape context and almost 8 times improvements for Mean-shift segmentation. Note that about half of the MAP of our combination strategy is less than either one feature type of the two, SIFT or shape context, for some queries. This is partly oriented from the fact that our algorithm is a combination of three methods with stronger robustness for most queries but not targeting for a specific trademark. Moreover, comparing the adaptive weights method with the non-adaptive weights one we
can find the adaptive method work well for half queries with up to 17% MAP improvements and 3.4% MAP improvements for the total queries. The result verifies our original starting point that it is possible to model an image only by its unique features and exploit corresponding type of features to represent it. To some extent, we can view an image as “Point-type” image, “Shape-type” image or “Patch-type” image.

Table 2: Experiment Results of Logo retrieval.

<table>
<thead>
<tr>
<th>Logo name</th>
<th>SIFT</th>
<th>Shape context</th>
<th>Mean-shift segmentations</th>
<th>Non-adaptive weights</th>
<th>Adaptive weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burberry</td>
<td>0.62</td>
<td>0.04</td>
<td>0.13</td>
<td>0.88</td>
<td>1.05</td>
</tr>
<tr>
<td>Coberry</td>
<td>0.62</td>
<td>0.04</td>
<td>0.13</td>
<td>0.88</td>
<td>1.05</td>
</tr>
<tr>
<td>Canon</td>
<td>0.62</td>
<td>0.04</td>
<td>0.13</td>
<td>0.88</td>
<td>1.05</td>
</tr>
<tr>
<td>Citron</td>
<td>0.56</td>
<td>0.96</td>
<td>0.17</td>
<td>0.85</td>
<td>0.86</td>
</tr>
<tr>
<td>Ferrari</td>
<td>0.91</td>
<td>0.73</td>
<td>0.19</td>
<td>0.88</td>
<td>0.95</td>
</tr>
<tr>
<td>GE</td>
<td>0.81</td>
<td>1.00</td>
<td>0.61</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>Hermes</td>
<td>0.81</td>
<td>1.00</td>
<td>0.61</td>
<td>0.94</td>
<td>0.94</td>
</tr>
<tr>
<td>Hongkongia</td>
<td>0.72</td>
<td>0.95</td>
<td>0.17</td>
<td>0.76</td>
<td>0.74</td>
</tr>
<tr>
<td>HP</td>
<td>0.98</td>
<td>0.95</td>
<td>0.19</td>
<td>0.89</td>
<td>0.98</td>
</tr>
<tr>
<td>Intex</td>
<td>0.78</td>
<td>1.00</td>
<td>0.61</td>
<td>0.88</td>
<td>0.96</td>
</tr>
<tr>
<td>Jordan</td>
<td>0.81</td>
<td>0.90</td>
<td>0.19</td>
<td>0.89</td>
<td>0.86</td>
</tr>
<tr>
<td>KFC</td>
<td>0.93</td>
<td>0.72</td>
<td>0.13</td>
<td>0.85</td>
<td>0.86</td>
</tr>
<tr>
<td>Kodak</td>
<td>0.86</td>
<td>0.86</td>
<td>0.29</td>
<td>0.95</td>
<td>0.96</td>
</tr>
<tr>
<td>LG</td>
<td>0.91</td>
<td>0.95</td>
<td>0.27</td>
<td>0.91</td>
<td>0.90</td>
</tr>
<tr>
<td>Mediolinon</td>
<td>0.91</td>
<td>0.81</td>
<td>0.21</td>
<td>0.97</td>
<td>0.96</td>
</tr>
<tr>
<td>Melelion</td>
<td>0.97</td>
<td>0.94</td>
<td>0.64</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>Peugeot</td>
<td>0.78</td>
<td>0.87</td>
<td>0.13</td>
<td>0.87</td>
<td>0.87</td>
</tr>
<tr>
<td>PO</td>
<td>0.89</td>
<td>0.93</td>
<td>0.65</td>
<td>0.87</td>
<td>0.87</td>
</tr>
<tr>
<td>Panas</td>
<td>0.81</td>
<td>0.90</td>
<td>0.13</td>
<td>0.92</td>
<td>0.94</td>
</tr>
<tr>
<td>Ralional</td>
<td>1.00</td>
<td>0.96</td>
<td>0.13</td>
<td>0.85</td>
<td>1.00</td>
</tr>
<tr>
<td>Red Dot Chi Pepears</td>
<td>0.88</td>
<td>0.83</td>
<td>0.67</td>
<td>0.98</td>
<td>0.96</td>
</tr>
<tr>
<td>Skoda</td>
<td>0.76</td>
<td>0.66</td>
<td>0.44</td>
<td>0.76</td>
<td>0.76</td>
</tr>
<tr>
<td>Unlevier</td>
<td>0.84</td>
<td>0.88</td>
<td>0.19</td>
<td>0.98</td>
<td>0.92</td>
</tr>
<tr>
<td>UPS</td>
<td>0.91</td>
<td>0.98</td>
<td>0.14</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Volkswagen</td>
<td>0.75</td>
<td>1.00</td>
<td>0.66</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td>Waykhide</td>
<td>0.81</td>
<td>0.90</td>
<td>0.61</td>
<td>0.86</td>
<td>0.86</td>
</tr>
<tr>
<td>All</td>
<td>0.83</td>
<td>0.86</td>
<td>0.19</td>
<td>0.89</td>
<td>0.92</td>
</tr>
</tbody>
</table>

5.4 Results on BelgaLogos
To compare our logo retrieval method with [3], we have launched an experiment on the BelgaLogos dataset. As an illustration in Figure 5, the results show that the MAP of our method is higher than the query expansion method for most queries especially for some logos which is full of text information, such as Adidas and TNT logos. The improvement is limited by the fact that insufficient images for some brands in the BelgaLogos dataset. There are only 5 images covering the Peugeot brand and for Gucci and Roche the number is 2.

![Figure 4: Effect of cluster size on logo retrieval.](image)

6. CONCLUSION
In this paper, we have proposed an effective logo retrieval approach with adaptive weights updating strategy for multiple local features. In particular, we have demonstrated its outstanding performance on the field of web logo search. By observing different kinds of logos and analyzing its unique feature type, we introduced a concept to divide hundreds of trademarks into several groups according to the type of feature which can model the logo best and naming as “Point-type”, “Shape-type” and “Patch-type”. Then we applied a practical updating mechanism to dynamic assign different weights to each feature type in the iteration process. The experimental results have demonstrated its superior performance to previous logo retrieval algorithms.

7. ACKNOWLEDGEMENT
This work is supported by National Natural Science Foundation of China (Grant No. 60905008, 60833006) and National Basic Research Program (973) of China under contract No.2010CB327905.

8. REFERENCES