SYM-FISH: A Symmetry-aware Flip Invariant Sketch Histogram Shape Descriptor

Xiaochun Cao[†,‡], Hua Zhang[†], Si Liu[§], Xiaojie Guo[‡], Liang Lin[£]
[†]School of Computer Science and Technology, Tianjin University, Tianjin, China
[‡]Institute of Information Engineering, Chinese Academy of Sciences, Beijing,China
[§]Department of Electrical & Computer Engineering, National University of Singapore, Singapore
[£]School of Software, Sun Yat-Sen University, Guangzhou, China
caoxiaochun@iie.ac.cn, huazhang@tju.edu.cn, dcslius@nus.edu.sg, xj.max.guo@gmail.com,linliang@ieee.org

Abstract

Recently, studies on sketch, such as sketch retrieval and sketch classification, have received more attention in the computer vision community. One of its most fundamental and essential problems is how to more effectively describe a sketch image. Many existing descriptors, such as shape context, have achieved great success. In this paper, we propose a new descriptor, namely Symmetric-aware Flip Invariant Sketch Histogram (SYM-FISH) to refine the shape context feature. Its extraction process includes three steps. First the Flip Invariant Sketch Histogram (FISH) descriptor is extracted on the input image, which is a flip-invariant version of the shape context feature. Then we explore the symmetry character of the image by calculating the kurtosis coefficient. Finally, the SYM-FISH is generated by constructing a symmetry table. The new SYM-FISH descriptor supplements the original shape context by encoding the symmetric information, which is a pervasive characteristic of natural scene and objects. We evaluate the efficacy of the novel descriptor in two applications, i.e., sketch retrieval and sketch classification. Extensive experiments on three datasets well demonstrate the effectiveness and robustness of the proposed SYM-FISH descriptor.

1. Introduction

With the popularity of tablets, e.g. iPad and Microsoft Surface, sketch related studies become unprecedented popular nowadays. For instance, via such devices, people can easily draw any object in his/her mind by touching the screens. The sketches drawn by users are used as queries to feed into any of the sketch retrieval system. The sketches are essentially different with the real life images in many aspects. For example, the information of sketches is mostly represented by edges, in contrast, however, the things in reality are very likely to be with rich textures. That is to say, a huge gap exists between the simple stroke and the objects in the real world, and thus brings great challenges for solving the problem. Actually, beside sketch retrieval [5], many other edge related tasks, such as sketch detection [23] and sketch recognition [2] are also extensively studied.

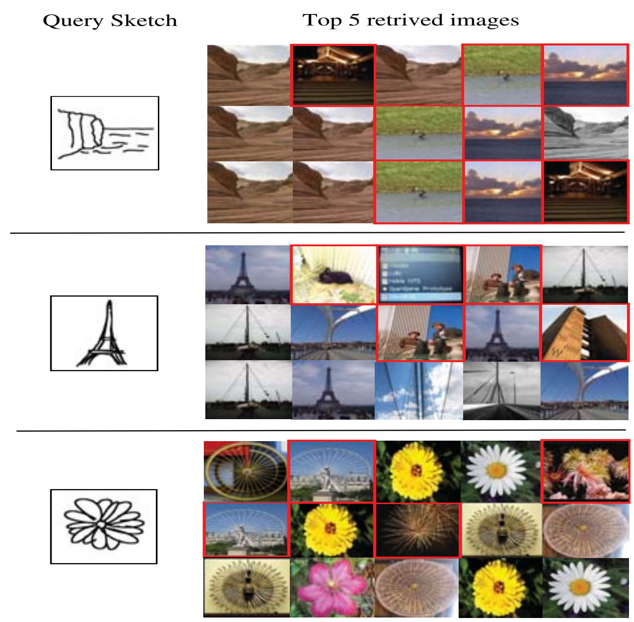


Figure 1. Several retrieval results of three query sketches are shown. The first query is non-symmetric, the second query is bilateral symmetric while the last one is rotation symmetric. For each query, the retrieval results of three kinds of shape descriptors: shape context, FISH and SYM-FISH are shown sequentially in different rows. The first column is the query sketch images, and the remaining columns are returned real life images. The incorrect retrieval images are highlighted by red bounding boxes.

To describe the shape information of the sketch, many descriptors are proposed. The most widely used one is shape context [2], which shows great success in practice. But it is unable to capture a very important property of images: symmetry. Symmetry is ubiquitous in both natural and man-made environments from galaxies, buildings to biological structures as well as in the arts. Moreover the symmetry is of scale invariance as well as translation invariance. Although there is a long history of symmetry study [9, 13, 21], it has rarely been integrated into descriptors in a unified framework.

In this paper, our goal is to design a symmetry-aware shape descriptor. Three steps are conducted sequentially. First, the image is represented by a flip invariant descriptor. More specifically, a Flip Invariant Sketch Histogram (FISH) descriptor is extracted (Section 3.1). It is rotation, translation, scale and flip invariant. Second, we detect the symmetry axis. To this end, a simple energy measurement based on the matching costs between different feature points are calculated (Section 3.2). The measurement is densely computed on the image patches under different orientations. Then we minimize the energy measurement to determine the symmetry directions in each image patch. Finally, we incorporate the symmetry character into image representation. We construct a graph named symmetry table to describe the symmetry character and generate the SYMmetric-aware Flip Invariant Sketch Histogram (SYM-FISH)(Section 4.1). Please note that, based on the graph, we can handle both the case with and without symmetry property in the image.

To validate the effectiveness of our proposed approach, we apply it on two applications: sketch retrieval [5] and sketch classification [15]. The sketch retrieval task is quite challenging because of the huge gap between the sketch query and real life repository images. Some of the representative works on sketch retrieval are MindFinder [4] [5] and Sketch2photo [6]. These works have achieved some impressive success. However, little attention is paid on studying the symmetry character of the images. Symmetric is very essential in image retrieval. For an instance in Figure 1, the second and the third query images are symmetric, so the retrieval results should also be symmetric. Furthermore, we do the experiment on two benchmark datasets: ETH shape dataset [8] and a large scale sketch retrieval dataset [17]. Promising results have been achieved. The second application is sketch classification. Intuitively, symmetry is useful for classification, i.e., sketch images from the same category may have certain common preference to symmetry. Such as apples are usually bilaterally symmetric and flowers tend to rotation symmetric. We conduct extensive experiments on the sketch dataset [15]. Experimental results show that the proposed SYM-FISH descriptor is more discriminating than standard descriptors, such as shape context, and can significantly improve sketch classification performance.

2. Related work

There are limited related works on sketch classification. One of the most representative work is [2], which is a histogram representation of the sample points. It maps interest points into a log-polar space based on the relative positions of the points. In addition, self-similarity [22] adopts a similar log-polar mapping function to compute the intensity differences among log-polar bins. Another work on sketch classification is [15]. The authors used a traditional bag of word method to classify the sketches. However, all the aforementioned descriptors do not enforce the descriptors to be flip invariant, while our propose SYM-FISH can handle the flip cases well.

Image retrieval has made a significant progress in recent years [12, 20]. Sketch retrieval is as a branch of image retrieval but with more difficulties. In MindFinder system [4], the edge position and gradient were used to represent the sketches. To make the representation translation invariant and more discriminative, [16] proposed a tensor descriptor which firstly divided the image into cells, and then computed the dominant orientation of each cell to construct a structure preserving descriptor. The above-mentioned descriptors are not robust to the image rotation. Therefore, we propose a novel sketch descriptors which can handle various transformations *e.g.* translation, rotation and scale. Furthermore, [17] proposed a bag of features framework based on local features. But the retrieval results would be influenced by the ambiguity of visual words. Thus, in our proposed method, we introduce the symmetry structure of the image to compensate such shortcoming.

Symmetry detection has been studied for many years [13]. A recent related work is [10], which proposed a symmetry score approach to find the symmetry feature points, afterwards constructed a symmetry descriptor for building matching. But our work is different from theirs because their method involves the score strategy, but SYM-FISH is based on the feature points matching.

To encode the full structure information in descriptor construction process will lead it sensitive to rotation. Therefore, Zhang *et al.* [24] propose to build a GVP (geometrical visual phase) to represent an image. In [24], the authors construct an offset space to compute the co-occurrences visual phase words in a particular spatial layout. However this method may work well for rigid construction e.g. buildings, when there existing some distortions the visual phase will also be changed. Our proposed symmetry visual word phase is robust to the distortions and noise.

3. The approach

The whole procedure of extracting SYM-FISH descriptor consists of three main components: 1) computing the FISH descriptor on the input image, 2) discovering the sym-

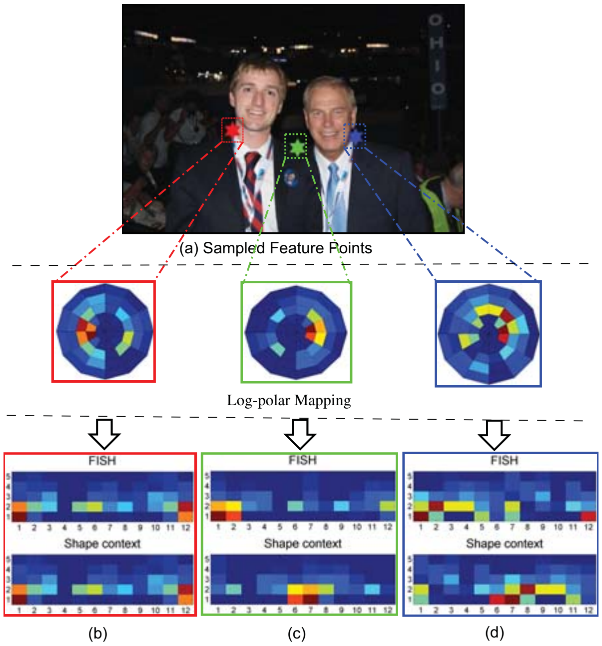


Figure 2. FISH descriptor construction process: sampling feature points, mapping sampled points in the log-polar coordinate, and developing the descriptor representation. We compare the proposed FISH descriptor with shape context descriptor. (b), (c) and (d) are the visualization of FISH and shape context descriptors of red, green and blue stars in the original image (a).

metry character of the image by analyzing matching scores and 3) constructing a symmetry table combined with FISH descriptor to finally generate the SYM-FISH.

3.1. Flip Invariant Sketch Histogram

Before we introduce the procedure of extracting FISH descriptor, we first briefly review how shape context descriptor is extracted. The procedure is shown in Fig. 2. Shape context is constructed based on the distribution of feature points sampled along the edge, which is detected by edge detector operator or by computing the gradient of the image. Then the log-polar mapping function (five bins for radius and twelve for angles in our figure representation) is applied and the number of feature points in each log-polar bin is computed. After normalization, shape context feature can be obtained. Although shape context has achieved great success, it cannot handle the flip case. For example in Fig. 2 (b) and (d), their shape context features are totally dissimilar even though their feature points corresponding to red and blue star in Fig. 2 (a) are quite similar (only under flip changes).

To handle the flip variations, we propose a FISH descriptor, which can be viewed as a post-processing procedure after shape context feature is extracted. More specifically, we re-order all the bins in the shape context by two steps: determine the reference bin and the rotation orientation sequentially. First, we determine the reference bin. It is set as the most dense bin (MDB) of the log-polar, *i.e.* bin marked by deepest color of the shape context feature in Fig. 2. After determining the MDB, we re-order all bins in shape context by putting the MDB in the first bin of the FISH descriptor. Second, the rotation orientation is determined by the orientation from MDB to the second most dense bin (SMDB). To sum up, we can roughly align the FISH features by re-ordering the bins of shape context according to the inferred reference bin and the rotation orientation. One possible problem of the above mentioned strategy is the MDB and SMDB bins may share the same polar angle. In this case, we skip the original SMDB and depend on the third MDB to determine the rotation orientation.

We show the effectiveness of the proposed FISH descriptor in Fig. 2. The red and blue stars in Fig. 2 (a) are quite similar, and thus their FISH are similar, shown in Fig. 2 (b) and Fig. 2 (d). To the contrary, the green star in Fig. 2 (a) looks quite dissimilar with the other stars. Therefore, its FISH Fig. 2 (c) is dissimilar with Fig. 2 (b) and Fig. 2 (d).

Since we map the feature points into log-polar space. Moreover, the relative distance is used to develop feature points distribution. Therefore, FISH is translation, rotation and scale invariant. After re-ordering, FISH is also flip invariant.

Table 1. The average evaluation results between FISH and Shape context, and the best are highlighted with bold.

	FISH		Shape Context	
Transformation	Precision	Recall	Precision	Recall
Rotation	**0.0588**	**0.3098**	0.0160	0.1062
Flip	**0.6357**	**0.8954**	0.0046	0.0453
Scale$_1$	**0.1847**	**0.1552**	0.1839	0.1550
Scale$_2$	0.1582	0.1316	0.1582	0.1316
Scale$_3$	0.1400	**0.1160**	0.1408	0.1158
Scale$_4$	0.1348	0.1080	**0.1353**	**0.1092**
Scale$_5$	0.**1260**	0.1028	0.1251	**0.1031**

Evaluation of Flip Invariant Sketch Histogram: We will quantitatively compare FISH with shape context in the image matching task. However there is no benchmark sketch dataset specially for matching, a sketch pairs database is collected by ourselves. The dataset is composed of 250 pairs. each pair is consisted of original image and its rotation, flip and scale version. In the sketch pair database, the orientation angle is randomly selected from (0,360). The scale parameter selected from the fixed 5 scales from $\frac{1}{\sqrt{2}}$ to 2. For the flip situation, we flip the whole original image.

The setting of the matching experiment is as follows: 300 feature points are sampled on two sketch images, then an adjacency matrix is constructed by computing the similarity among their FISH descriptors. Afterwards, the KM matching approach [18] is used to get the global one-vs-one correspondence of feature points. Finally a RANSAC [19] method is utilized to further improve the matching results.

We use precision and recall to evaluate the matching per-

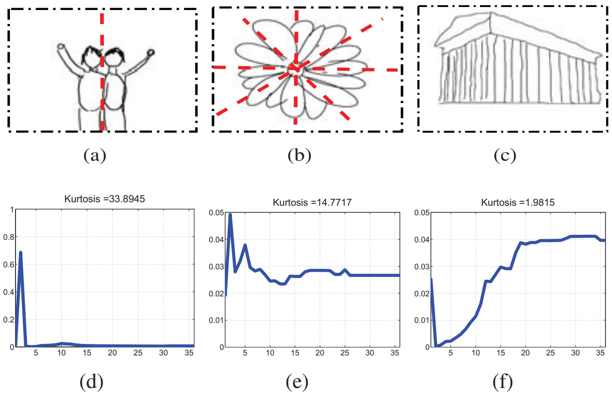


Figure 3. The Kurtosis values of different symmetry types. The red line indicates the detected symmetry axis of the original image. For bilateral symmetry image (a), the kurtosis value is large and the distribution has a peak (d). While the rotation symmetry image (b) has a small kurtosis value and a flat distribution (e). The non-symmetry image (c) has a smaller kurtosis value distribution (f).

formance. A match is considered as correct if the estimated matching points are less than 10 pixels away from the ground truth matching points. The results can be found in Tab. 1. In this table, we show the average matching results of 250 sketches. The reason is that the variation tendency of precision and recall is like other local descriptors. We think the average value can be used to measure the performance. In the rotation and scale situations, FISH achieves a competitive matching results comparing with shape context. And under flip situation, FISH obtains significant improvements.

3.2. Symmetry discovering

According to common sense and previous studies[10], the symmetry sketch can be divided into two categories[1] : bilateral symmetry and 2n-rotataion symmetry, the definition of which are:

Bilateral symmetry: A sketch contains only one symmetry axis or two (near) orthogonal axises. Specially, two separated parts could be mapped to each other by the angle of the symmetry axis. One example of bilateral symmetry is Fig. 3 (a).

2n-fold rotation symmetry: There exist more than two symmetry lines of a sketch, while these lines are intersected in one point. In our problem, if a sketch include two symmetry lines, which are not orthogonal, we also define it as rotation symmetry. One example of rotation symmetry is Fig. 3 (b).

Thus, discovering local symmetry of a region can be converted to search the symmetric axis of the sketch. To detect the symmetry axis on the input sketch, we propose a compact energy minimization method. The whole strategy is

Algorithm 1 The procedure for symmetry discovering.

1: **Input**: an image I_i, the all-zero 36-dim vector ℓ, threshold $\hbar_1$ and $\hbar_2$
2: **for** each sampled angle O within (0,360) degrees **do**
3: **for** each sampled point p in the image I **do**
4: calculate the symmetric score of the {O, p} pair by Eq. 2.
5: **end for**
6: set $\ell(o) = Score_{I_i}(O)$.
7: **end for**;
8: calculate kurtosis coefficient $Kurt(V_{Score})$ by Eq. 3.
9: **if** $Kurt(V_{Score}) \geq \hbar_1$ **then**
10: image I is bilateral symmetric
11: **else**
12: **if** $\#matchedpoints \geq \hbar_2$ **then**
13: image I_i is rotation symmetric
14: **end if**image I_i is not symmetric
15: **end if**
16: **Output**: its symmetry type and the symmetric points;

shown in Algorithm 1. Firstly, as shown in line 2 of Algorithm 1, we traverse all orientations/angles o, which are evenly distributed in (0,360) degrees, with the interval of 10 degrees. Thus, we have 36 sampled angles. Secondly, we traverse all sampled keypoints p in the image I_i, as shown in line 3. Till now, we get the sampled {o, p}(indicating {angle, point}) pair. Then we divide the sketch image into two parts based on current {o, p} pair. Then we calculate the symmetric score for each orientation/angle o. Smaller matching score means more symmetric. The symmetric score is defined as:

$$Score_{I_i}(O) = \sum_j min\ D(f_j^i(O), f_{c(j)}^i(O)), \quad (1)$$

$$\forall O \in \{10, 20, ..., 360\} \quad (2)$$

where $c(j)$ represents the corresponding feature points of j, and f_j^i is the feature representation of point j in the image i. $D(\cdot)$ displays the Euclidean distance. O represents the symmetry directions, which is fixed from 10 degree to 360 degree for every 10 degrees. Then, we select the minimum scores of each orientation o as the potential symmetry orientation, We accumulate all the scores to generate a 36-dim vector, shown in line 6 of Algorithm 1. We observe that for the bilateral symmetry sketch the matching score $Score_{I_i}(O)$ has an unimodal distribution, while for the rotation symmetry, $Score_{I_i}(O)$ has a multimodal distribution. The kurtosis coefficient [11] can discriminate the two cases: output lower value for a single peak distribution and higher value for the multi-peak distribution. The kurtosis coefficient is calculated by:

$$Kurt(V_{Score}) = \frac{1}{\sigma^4} \sum_\lambda (\lambda - \mu_\lambda)^4 Score_{I_i} \quad (3)$$

[1]The curved symmetry may be considered as another symmetry categories. But we think the curved symmetry is a kind of piecewise bilateral symmetry. Moreover in this paper we do not discuss such situation.

where σ is the standard deviation and λ represents the angel of the symmetry axis. μ_λ is the expected value of λ and V_{score} is the distribution of confidence symmetry. The step corresponds to line 8 in Algorithm 1. Usually, the bilateral symmetry produces much higher *Kurt* score comparing with rotation symmetric and non-symmetric, as shown in Fig. 3 (d). We take advantage of this property, and set a threshold $\hbar_1$ shown in line 9 of Algorithm 1. Since both rotation symmetric and non-symmetric produce similar lower *Kurt* score, shown in Fig. 3 (e) and Fig. 3 (f). We have to judge the type by another criterion, i.e., the number of matching feature points $\hbar_2$. Intuitively, non-symmetric images have small number of matched feature points, as shown in line 12 of Algorithm 1. Till now, we can classify all three kinds of symmetric types.

Evaluation of symmetry discovering: We test the effectiveness Algorithm 1 on a subset of sketch database [17]. The validation database is composed of 31 human drawing sketches, which contain bilateral symmetry sketch, rotation symmetry sketch and non-symmetry sketch. We would like to know whether the symmetry type can be correctly classified. We find that the total classification accuracy is 67.7%, which is much higher than random guess 33.3%.

3.3. Symmetry-aware Flip Invariant Sketch Histogram

In image retrieval and classification the local descriptors, such as SIFT [14], shape context [2] and FISH, will not be directly used for the image representation. Usually, we summarize all the local descriptors in a sketch image with the Bags of words (BoWs) representation. Thus, in this section, we will illustrate how to fuse the symmetry property among feature points into the visual word representation. Traditionally BoWs features ignore the relationships between different visual words. A lot of works have put attention on adding the spatial relation between different visual words [3] [24]. In this paper we only focus on symmetry and do not exploit other spatial structure of the sketch images.

We propose to use a symmetry table to capture the symmetry relations among visual words. The whole process contains four steps. Firstly, the k-means is used to cluster the feature points to get the dictionary. And we map all the feature points into its nearest visual words to get the visual word representation. This step is same with traditional BoWs framework. Secondly, we detect the symmetric feature points by the method introduce in Sec 3.2. Thirdly, we map the symmetry of feature points to symmetry of visual words. This step is similar with [3] whose purpose is to transfer the feature points spatial distribution to visual word representation. Finally, we construct a symmetry table $Y \in \{0, 1\}$, which is a N×N matrix, where N is the number of visual words. Y is an index matrix, whose element $Y_{i,j}$ indicates whether the visual word V_i and V_j are symmetric in the sketch images. More specifically, $Y_{i,j} = 1$ if v_i and v_j are symmetry, otherwise $Y_{i,j} = 0$. With the symmetry table, the symmetry relationship is transferred from feature points level to the visual words level. To sum up, besides the original BoWs feature, for each sketch image, we have a new structural feature called SYMmetry-aware Flip Invariant Sketch Histogram (SYM-FISH) shape feature. It is the combination of original FISH feature and a symmetry table. The SYM-FISH feature is easy to compute and all its values are binary. Thus the distance between two symmetry table is just humming distance.

4. Applications of SYM-FISH

4.1. SYM-FISH descriptor in sketch retrieval

Searching the real life images by using a sketch query is not an easy task. The sketches are essentially different with the real life images in many aspects. For example, sketch images convey information mostly by edges while real life images always have rich texture.

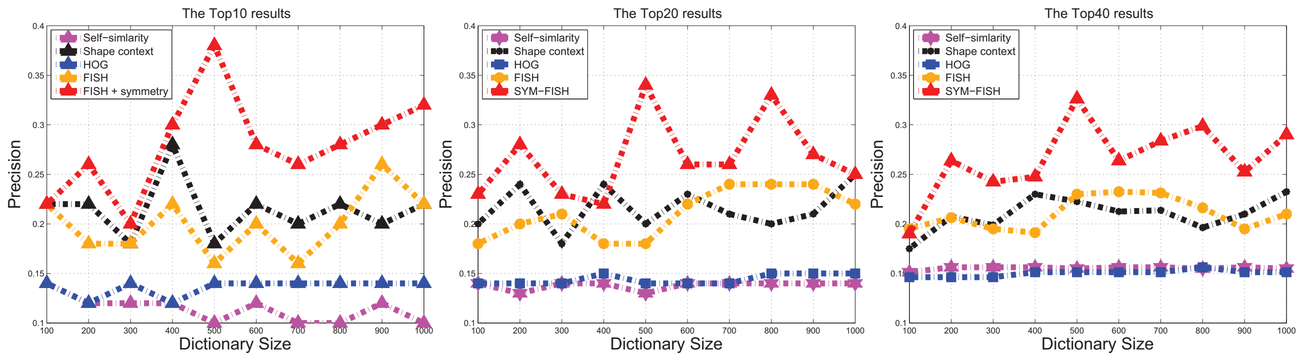
The SYM-FISH is used in the sketch retrieval task by re-ranking the original ranking list. The original list could be generated by any retrieval method, such as using the inverted file structure. For the SYM-FISH reranking, we first extract the symmetry table for all the images in the repository. Then the original list is reorder by the distances between symmetry table. We use Eq. 4 to compute the distance between symmetry tables.

$$D(I_q, I_r) = \|ST_q - ST_r\|_F, \tag{4}$$

where I_q and I_r display the query image and repository image. ST is the symmetry table for each image. The subscript F represents the Frobenius norm.

From the experiments, we observe that SYM-FISH not always output good rerank results. The main reason is that it is very difficult to extract the edges of real life images . And the SYM-FISH descriptor is sensitive to edge detection errors. To partially solve the problem, we use subwindows method.

The candidate subwindows are chosen based on the their objectness [1]. The objecness of an image is defined to find the regions which most likely include the objects. Supposing we have m candidate windows in the query image, and n candidate windows in one repository image. Next, we should identify how to measure the distance between query and repository images. Till now, the original image-image distance has been transfered to a set-set distance, where all candidate windows in an image form a set. To solve the problem, we assume that if two images are similar, there exist some quite similar subwindows too. Therefor, we measure the image-image distance by averaging the distance of



(a) (b) (c)

Figure 4. The sketch retrieval results on ETH dataset. (a)-(c) are average retrieval results for top 10, 20 and 40 cases.

top 3 similar subwindows:

$$D(I_q, I_r) = \frac{1}{3}\sum_{i=1}^{3} d(i, I_q^p, I_r^p), \tag{5}$$

where $d(i, I_q^p, I_r^p)$ denoted the i-th similar subwindows between query image I_q and reposotyr image I_r.

4.2. SYM-FISH descriptor in sketch classification

In the traditional classification approach [15] , chi-square distance is usually used to compute the similarity between different images while the symmetry character of the s-ketches is not considered. The chi-square distance is just based on the similarity of the bags of visual words representation. In our approach, we combine the distance between visual word representation and the similarity of symmetry table. Formally, we have:

$$D(I_i, I_j) = \chi^2(I_i, I_j) + \lambda * ST(I_i, I_j) \tag{6}$$

where $\chi^2(I_i, I_j)$ computes the chi-square distance between different images, and $ST(I_i, I_j)$ is the similarity of symmetry table between different images. However, there exists variation within the category, this distance may increment the distance within the class. In fact, we observe that the distance within the category is increasing, but the distance between different categories is growing more larger. To validate the performance of such combination, the experiment results are shown in Section 5.2.

5. Experiments

We evaluate our proposed descriptor FISH and SYM-FISH on two applications : sketch retrieval and sketch classification. In the sketch retrieval experiment, we first check the performance of FISH and SYM-FISH on ETH shape database [8]. The further validation of these two descriptors is on a large scale sketch retrieval dataset [17]. In the sketch classification experiment, we test the performance of proposed descriptors on sketch classification dataset [15].

5.1. Sketch Retrieval Results

5.1.1 Evaluation on the ETH dataset

In this experiment, we compare FISH, SYM-FISH with shape context [2], self-similarity [22] and HOG [7] on ETH shape dataset. This dataset contains five classes (bottles, swans, mugs, giraffes and apple logos) with a total of 255 images collected from the web. It is very challenging, as the objects appear in a wide range of scales. And there is considerable intra-class shape variations, and many images are severely cluttered, with objects comprising only a fraction of the whole image. The ETH dataset provides one representative sketch image for each class, which is used as the input query in our experiment. All the real life images in the dataset are used as repository data. We use precision to measure the performance of different descriptors, which is the ratio of corresponding images in the top n returned images: *Precision = #corresponding images/ n*.

The results are shown in Fig. 4, we test all the five descriptors in top 10, 20 and 40 cases. And we conclude that SYM-FISH achieves the best performance. Note that it is significantly better than shape context, the most widely used shape descriptor. The possible explanation is that the proposed SYM-FISH descriptor becomes more robustness because of the novel encoding approach and can better handle the flip situation. Moreover, the introduced symmetry property enables the representation more discriminative. We also test the performance w.r.t. different vocabulary sizes from 100 to 1000. From Fig. 4, we can see that in all cases, the maximal precision is achieved when the dictionary size is 500, which is a trade-off between descriptors' discrimination and quantization error.

5.1.2 Evaluation on the Large Scale Image Database

We have demonstrated the effectiveness of FISH and SYM-FISH on a relatively small ETH database. In this experiment, we validate their performances on a large scale dataset [17]. This database is composed of two parts, namely a benchmark dataset and a distractor image dataset. The benchmark dataset contains 31 benchmark sketches as well as 40 corresponding images for each sketch while the distractor image dataset contains 100,000 creative commons images. We mix the benchmark dataset and the distractor images together and use each of the 31 benchmark sketches as the query. For each query, a list which contains the ranking of the corresponding 40 benchmark images is stored. We use Kendalls rank correlation [17] to measure the performance ranging in [-1, 1] and higher value means the higher consistency:

$$\tau = \frac{\#concordant\ pairs - \#discordant\ pairs}{\frac{1}{2}n(n-1)}, \qquad (7)$$

where # concordant pairs evaluates the consistency between two lists and # discordant pairs measure the inconsistency between two lists. n is the length of the rank lists.

In this experiment, we compare the description performance of FISH and SYM-FISH with shape context. All results are illustrated in Table 2. We can observe that FISH is marginally better than shape context and SYM-FISH is the best one. We also show several qualitative retrieval examples in Fig. 5. The results are obtained by using SYM-FISH descriptor. We can see that, in most cases, the returned results are visually and semantically similar with the query.

Table 2. The retrieval results on large scale image benchmark [17]. For all dictionary size, the best results are achieved by SYM-FISH.

Dictionary Size	Shape context	FISH	SYM-FISH
100	0.12395	0.11683	**0.1261**
300	0.10911	0.11041	**0.1409**
500	0.10914	0.10815	**0.1688**
700	0.10391	0.10225	**0.1762**
900	0.11594	0.12225	**0.1829**
1000	0.11066	0.1243	**0.1801**

5.2. Sketch Classification Results

In this section, we test the effectiveness of the proposed SYM-FISH on the sketch classification task. In this part, a subset of human sketch dataset [15] is used, which contains 24 categories: airplane, bicycle, car(sedan), cat, chair, computer monitor, couch, cow, dog, flying bird, horse, motorbike, person setting, person walking, potted plant, race car, sailboat, sheep, speed boat, table, table lamp, train, tv and wine-bottle. Each sketch class contains about 80 images with different styles. To train the sketch model, we randomly divide the dataset into 2 subset: 58 images from each category are randomly selected as the training set and the remaining images are used as testing set. We train 24 SVM classifiers one for each category in a one-vs-all manner. We use the self-defined kernel converted from Eq. 6.

In this experiment, we compare three descriptors: shape context, FISH, SYM-FISH. The comparison results are

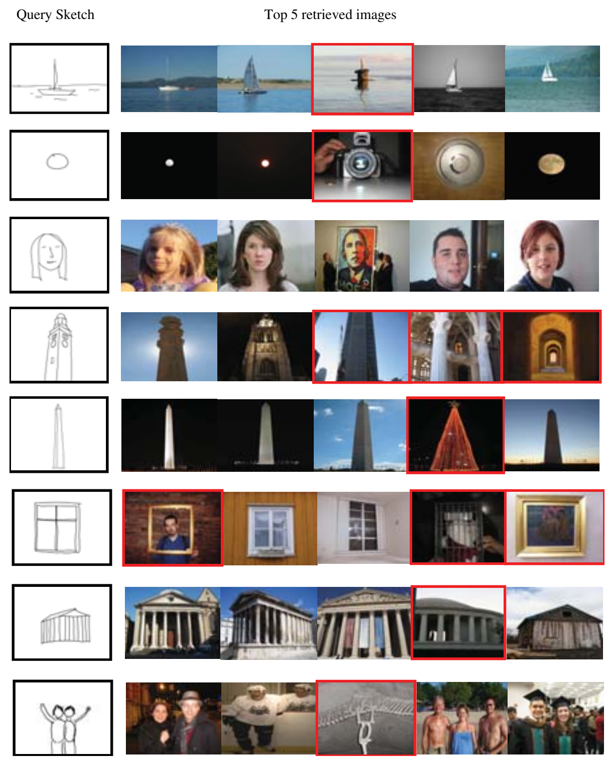


Figure 5. Examples of the retrieval results using the proposed SYM-FISH descriptor in the large scale dataset [17]. The first column is the query sketch image, while the remaining columns correspond to the retrieved real life images. The incorrect results are highlighted by red bounding boxes.

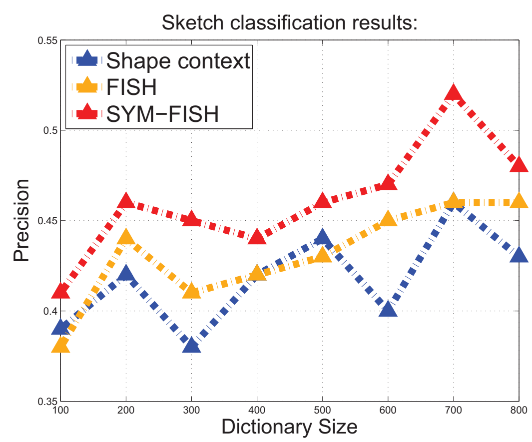


Figure 6. The classification results w.r.t. different dictionary size.

shown in Fig. 6. The evaluation metric is category's recognition accuracy. We can observe that in most cases, SYM-FISH achieves best performance. We also show the accuracies for each category in Tab. 2. Again, in 10 out of total 24 categories, SYM-FISH achieves best results. The average result of 0.53 is much higher than 0.47, which is the result of shape context. The reasons can be summarized as follows: firstly there exists many flip situations in the dataset

Table 3. The comparisons between our descriptors and baselines. The best results are highlighted in bold.

Categories	[15]	SC[2]	FISH	SYM-FISH
airplane	0.38	0.44	0.30	**0.63**
bicycle	**0.77**	0.56	0.44	0.59
car(sedan)	0.30	**0.63**	0.48	0.52
cat	0.22	0.26	0.15	**0.37**
chair	0.67	0.52	0.74	**0.78**
computer monitor	**0.81**	0.48	0.59	0.59
couch	0.62	0.59	0.63	**0.67**
cow	**0.5**	0.22	0.2	0.19
dog	**0.41**	0.19	0.11	0.37
flying bird	0.19	0.19	0.11	**0.22**
horse	0.54	0.52	**0.63**	0.44
motorbike	0.44	**0.48**	0.44	0.41
person setting	0.52	0.44	**0.59**	0.46
person walking	**0.69**	0.52	0.59	0.56
potted plant	0.81	0.67	0.81	**0.81**
race car	**0.22**	0.22	0.11	0.07
sailboat	**0.93**	0.59	0.74	0.67
sheep	**0.58**	0.41	0.44	0.56
speed boat	0.35	0.22	0.11	**0.37**
table	**0.89**	0.7	0.74	0.70
table lamp	0.27	0.59	0.70	**0.67**
train	**0.44**	0.37	0.33	0.41
tv	0.56	0.48	0.67	**0.67**
wine-bottle	0.81	0.89	0.70	**0.89**
Average	0.538	0.47	0.48	0.53

and our proposed descriptor is flip-invariant. Secondly, the symmetry table can better preserve the symmetry properties of the sketches which both decrease the intra-category distances and increase inter-category distances.

6. Conclusion and Future Work

In this paper, we propose a novel shape descriptor SYM-FISH which can handle the flip changes and encode image's symmetric property. It is low-dimensional and easy to compute. We thoroughly analyze its characteristics on two applications: sketch retrieval and classification. Experiments validate that SYM-FISH is significantly and consistently better than the shape context descriptors in most cases. Although we only validate the effectiveness of the descriptor on sketch retrieval and classification tasks in this paper, we believe that it can also be used in other tasks, such as sketch detection. We leave it as our future work.

7. Acknowledgement

This work was supported by National Natural Science Foundation of China (No. 61332012, 61170178), National High-tech R&D Program of China (2012*AA*011503), 100 Talents Programme of The Chinese Academy of Sciences, and Strategic Priority Research Program of the Chinese Academy of Sciences (*XDA*06030601).

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *TPAMI*, 2012.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *TPAMI*, 2002.

[3] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang. Spatial-bag-of-features. *CVPR*, 2010.

[4] Y. Cao, H. Wang, C. Wang, Z. Li, and L. Zhang. Edgel inverted index for large-scale sketch-based image search. *CVPR*, 2011.

[5] Y. Cao, H. Wang, C. Wang, Z. Li, L. Zhang, and L. Zhang. Mindfinder: Interactive sketch-based image search on millions of images. *MM*, 2010.

[6] T. Chen, M. Cheng, P. Tan, A. Shamir, and S. Hu. Sketch2photo: Internet image montage. *TOG*, 2009.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005.

[8] V. Ferrari, F. Jurie, and C. Schmid. From images to shape models for object detection. *IJCV*, 2010.

[9] L. G. and E. J. Detecting symmetry and symmetric constellations of features. In *ECCV*, 2006.

[10] D. Hauagge and N. Snavely. Image matching using local symmetry features. *CVPR*, 2012.

[11] K.P.Balanda and H.L.MacGillivray. Kurtosis: A critical review. *The American Statistician*, 1988.

[12] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *CVPR*, 2012.

[13] Y. Liu, H. Craig, S. Kaplan, and L. Gool. Computational symmetry in computer vision and computer graphics. *Foundations and Trends in Comp*, 2010.

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[15] E. Mathias, H. James, and A. Marc. How do humans sketch objects? *TOG*, 2012.

[16] E. Mathias, H. Kristian, B. Tamy, and A. Marc. A descriptor for large scale image retrieval based on sketched feature lines. *Eurographics Symposium on Sketch-Based Interfaces and Modeling*, 2009.

[17] E. Mathias, H. Kristian, B. Tamy, and A. Marc. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *TVCG*, 2011.

[18] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 1957.

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

[20] B. S., R. S. F., and L. S. D. Image ranking and retrieval based on multi-attribute queries. In *CVPR*, 2011.

[21] L. S. and L. Y. Skewed rotation symmetry group detection. In *TPAMI*, 2010.

[22] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. *CVPR*, 2007.

[23] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *TPAMI*, 2008.

[24] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry preserving visual phrases. *CVPR*, 2011.