Co-transduction for Shape Retrieval

Xiang Bai[1], Bo Wang[1], Xinggang Wang[1], Wenyu Liu[1], and Zhuowen Tu[2]

[1] Department of Electronics and Information Engineering,
Huazhong University of Science and Technology, China
{xbai,liuwy}@hust.edu.cn, {wangbo.yunze,wxghust}@gmail.com
[2] Lab of Neuro Imaging, University of California, Los Angeles
ztu@loni.ucla.edu

Abstract. In this paper, we propose a new shape/object retrieval algorithm, *co-transduction*. The performance of a retrieval system is critically decided by the accuracy of adopted similarity measures (distances or metrics). Different types of measures may focus on different aspects of the objects: e.g. measures computed based on contours and skeletons are often complementary to each other. Our goal is to develop an algorithm to fuse different similarity measures for robust shape retrieval through a semi-supervised learning framework. We name our method co-transduction which is inspired by the co-training algorithm [1]. Given two similarity measures and a query shape, the algorithm iteratively retrieves the most similar shapes using one measure and assigns them to a pool for the other measure to do a re-ranking, and vice-versa. Using co-transduction, we achieved a significantly improved result of 97.72% on the MPEG-7 dataset [2] over the state-of-the-art performances (91% in [3], 93.4% in [4]). Our algorithm is general and it works directly on any given similarity measures/metrics; it is not limited to object shape retrieval and can be applied to other tasks for ranking/retrieval.

1 Introduction

Shape-based object retrieval is an important task in computer vision. Given a query object, the most similar objects are retrieved from a database based on a certain similarity/distance measure, whose choice largely decides the performance of a retrieval system. Therefore, it is critically important to have a faithful similarity measure to account for the large intra-class and instance-level variation in configuration, non-rigid transformation, and part change. Designing such a measure is a very difficult task. Fig. (1) gives an illustration where a horse might have a smaller distance to a dog (based on their contours) than another horse, whereas our human vision systems can still identify them correctly.

In this paper, we refer to shape as the contour of an object silhouette. Our algorithm, however, is general and not limited to any particular similarity measure or representation. Building correspondences is often the first step in computing shape difference but it is challenging: two shapes may not have the direct correspondences in representation, regardless if they are represented by sparse points, closed contours, or parametric functions. For example, two shapes with the same

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part III, LNCS 6313, pp. 328–341, 2010.

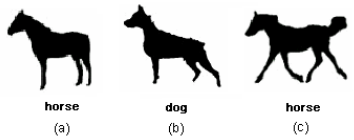

Fig. 1. A horse in (a) may look more similar to a dog in (b) than another horse in (c)

contour but different starting points typically are considered as the same one. Therefore, measuring the similarity between two shapes often can be done in two ways: (1) computing direct difference in features extracted from shape contours, which are invariant to the choice of starting points and robust to certain degree of deformation, such as moments and Fourier descriptors; (2) performing matching to find the detailed point-wise correspondences to compute the differences [5,6]. The latter recently becomes dominate due to their ability of capturing intrinsic properties, and thus leading to more accurate similarity measures. Recently, Yang et al. [3] explored the group contextual information of different shapes to improve the efficiency of shape retrieval on several standard datasets [2,7]. The basic idea was to use shapes as each others' contexts in propagation to reduce the distances between intra-class objects. The implementation was done by a graph-based transduction approach [8]. Later, several other graph-based transduction methods were suggested for shape retrieval [4,9]. Different similarity measures have different emphasis: for example, similarities computed on matching the skeletons of two objects may be robust against non-rigid transformation, but are hard to capture the rich variability in part change; similarities computed on matching the contour parts can capture subtle change but may not be robust against articulation. It would be natural to think to fuse/combine different complementary metrics together to achieve better performance. A straight-forward way is to linearly combine a few measures together. However, this often requires certain level of supervision or manual tuning and will not necessarily produce the best results (we will see a comparison in the experiments).

This paper provides a different way of fusing similarity/distance measures through a semi-supervised learning framework, *co-transduction*. The user input is a query shape and our system returns the most similar shapes by effectively integrating two distance metrics computed by different algorithms, e.g. Shape Contexts [5] and Inner-Distance [6]. Our approach is inspired by the co-training algorithm [1]. The difference though is that, in co-training, it requires having two conditionally independent views of the data samples. In our problem, each data only has one view but different algorithms report measures by exploring different aspects of the data. Therefore, they may lead to different retrieval results for the same query, which can be helpful to each other. For example, as shown in Fig. (2), the retrieval results of Shape Contexts (SC) [5] in the first row and Inner-Distance Shape Contexts (IDSC) [6] in the second row are very different as

their different shape representation, even they can gain the comparable Bull-eyes retrieval rate (SC: 86.8%[1], IDSC: 85.4%) in MPEG-7 Shape dataset [2].

Fig. (3) shows another example for illustrating the motivation of the proposed method: In Fig. 3(1), the SC distances between query shape A and B/C are not small due to articulation. However, in Fig. 3(2), IDSC reports different result as it is more stable than SC for articulation changes (it uses the inner distance to replace the Euclidean distance in SC's representation). As shown in Fig. 3, the SC distance between B and C is small as they have the same pose. Even though C is thicker than B, the SC distance still finds a good match between C and B. We use IDSC to retrieval B out firstly, and then put B and query A together as labeled data; a new classifier based on SC distance trained by A and B will give high confidence to C as shown in Fig. 3(4). Our algorithm is

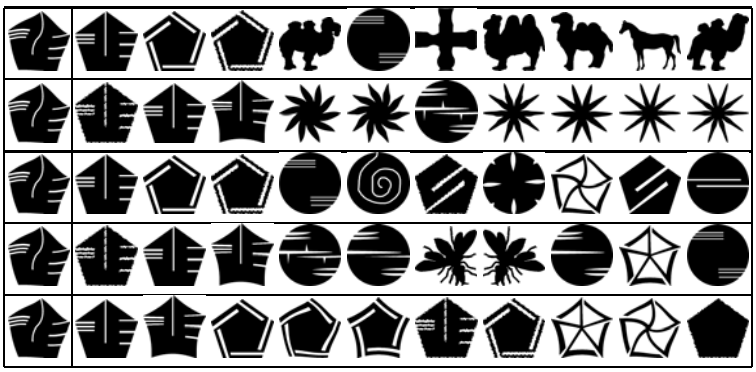

Fig. 2. The first column shows the query shape. The remaining 10 columns show the most similar shapes retrieved from the MPEG-7 data set. The 1st-4th rows are the retrieval results of SC [5], IDSC [6], SC+LP [3], IDSC+LP [3], respectively. The 5th row is the result of the proposed method by integrating two distance metrics computed by SC and IDSC.

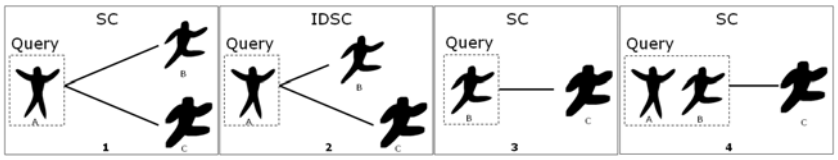

Fig. 3. The motivation of the proposed method

[1] Here we use Dynamic Programming (DP) to replace thin plate spline (TPS) as Belongie et al. did in [5] for the matching process and achieve 86.8% on MEPG-7 dataset. The new distance measure by DP based on SC descriptor is used as the input for our retrieval framework.

inspired by co-training [1]. However, unlike co-training in which two independent views (sets of features) are assumed, our algorithm deals with single-view but multiple classifiers; each transduction algorithm on a given similarity measure is a classifier and they help each other by sending most similar results to the others. Co-Transduction is also related to [10] but with the difference: (1) [10] tackles a regression problem; (2) kNN was used in [10]; (3) we focus on fusing different metrics for object retrieval.

2 Co-transduction Algorithm

We first briefly review the graph-based transduction algorithm (label propagation) [8] applied in shape retrieval [3]. Given a set of objects $X = \{x_1, ..., x_n\}$ and a similarity function sim: $X \times X \rightarrow R^+$ that assigns a positive similarity value to each pair of objects. Assume that x_1 is a query object (eg., a query shape), $\{x_2, ..., x_n\}$ is a set of known database objects(or a training set). Then by sorting the values $sim(x_1, x_i)$ in decreasing order for $i = 2, ..., n$ we can obtain a ranking for database objects according to their similarity to the query. A critical issue is then to learn a faithful sim. Yang et al. [3] applied label propagation (diffusion map) to learn a new similarity function sim_T that drastically improves the retrieval results of sim for the given query x_1. They let $w_{i,j} = sim(x_i, x_j)$, for $i, j = 1, ..., n$, be a similarity matrix, then obtain a $n \times n$ probabilistic transition matrix P as a row-wise normalized matrix w.

$$P_{ij} = \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}} \tag{1}$$

where P_{ij} is the probability of transit from node i to node j.

A a new similarity measure s is computed based on P. Since s is defined as similarity of other elements to query x_1, we denote $f(x_i) = s(x_1, x_i)$ for $i = 1, \ldots, n$. A key function is f and it satisfies

$$f(x_i) = \sum_{j=1}^{n} P_{ij}\ f(x_j) \tag{2}$$

Thus, the similarity of x_i to the query x_1, expressed as $f(x_i)$, is a weighted average over all other database objects, where the weights sum to one and are proportional to the similarity of the other database objects to x_i. In other words a function $f : X \rightarrow [0, 1]$ such that $f(x_i)$ is a weighted average of $f(x_j)$, where the weights are based on the original similarities $w_{i,j} = sim(x_i, x_j)$.

Note that LP is not limited to only one query object, which also can be used for 2 or more queries as it's a classification method (see the case in Fig. 3(4), there are two query objects A and B). Assume that $\{x_1, ..., x_l\}$ is a group of query objects, and $\{x_{l+1}, ..., x_n\}$ is a set of known database objects. Then the LP algorithm for computing the new similarity can be shown in Fig. 4.

In a general situation, graph-based transduction can be viewed as performing manifold regularization [11]. $f^* = \arg\min_{f \in \mathcal{H}_K} \sum_{i=j}^{l} V(x_j, y_j, f) + \lambda_1 ||f||^2_{\mathcal{H}_K} +$

Input: The $n \times n$ row-wise normalized similarity matrix P with the query $\{x_1 ..., x_l\}$, $f_1(x_i) = 1$ for $i = 1, ..., l$, and $f_1(x_i) = 0$ for $i = l+1, ..., n$.
while: $t < T$.
  for $i = l+1, ..., n$,
    $f_{t+1}(x_i) = \sum_{j=1}^{n} P_{ij}\ f_t(x_j)$
  end
  $f_{t+1}(x_i) = 1$ for $i = 1, ..., l$.
end
Output: The learned new similarity values to the query $\{x_1, ...x_l\}$: f_T.

Fig. 4. The pseudo-code of LP algorithm when the query includes a group of objects

$\lambda_2 \mathbf{f}^T L$ which is an approximation to the continuous function space of f based on the labeled (query objects in our case) and unlabeled data (database objects). L is the Laplacian map computed from the similarity measures P. $V(x_j, y_j, f)$ measures classification error of f on the supervised data and $||f||^2_{\mathcal{H}_K}$ is a regularize of f. Now we view LP as a tool to improve an input similarity function by taking the contextual information between objects. *The key problem we want to address in this paper is how to build a robust retrieval system, if there are two (even more) input similarity measures.* A straight-forward solution is to linearly combine different measures and use LP to gain further improvement. We will later show that this yields less encouraging results than the proposed algorithm, co-transduction.

Input: the labeled training set L
  the unlabeled training set U
Process:
  Create a pool U' of examples by choosing u examples at random from U
  Loop for k iterations:
    Use L to train a classifier h_1 that considers only the x_1 portion of x
    Use L to train a classifier h_2 that considers only the x_2 portion of x
    Allow h_1 to label p positive and n negative examples from U'
    Allow h_2 to label p positive and n negative examples from U'
    Add these self-labeled examples to L
    Randomly choose $2p + 2n$ examples from U to replenish U'

Fig. 5. Co-training Algorithm by Blum and Mitchell [1]

Fig. (5) and Fig. (6) give the pseudo-code for co-training [1] and the proposed co-transduction algorithm respectively. Same as in Yang et al. [3], a query object x_1 and database objects $\{x_2, .., x_n\}$ are respectively considered as labeled and unlabeled data for graph transduction. In spirit, co-transduction is in the co-training family; unlike the original co-training algorithm, co-transduction emphasizes single view but different metrics (in a way classifiers). It uses one metric to pull out confident data for the other metric to refine the performance. In implementation, the nearest neighbors of the query object are added to labeled data set for graph transduction in the next iteration based on the other shape

similarity. The final similarity sim_F of co-transduction is the average of all the similarities: $sim_F = \frac{1}{2}(sim_1^m + sim_2^m)$.

Input: a query object x_1 (a labeled data)
the database objects $X = \{x_2, ...x_n\}$ (unlabeled data)
Process:
Create a $n \times n$ probabilistic transition matrix P_1 based on one type of shape similarity (eg. SC)
Create a $n \times n$ probabilistic transition matrix P_2 based on another type of shape similarity (eg. IDSC)
Create two sets Y_1, Y_2 such that $Y_1 = Y_2 = \{x_1\}$
Create two sets X_1, X_2 such that $X_1 = X_2 = X$
Loop for m iterations:
Use P_1 to learn a new similarity sim_1^j by graph transduction when Y_1 is used as the query objects ($j = 1, ..., m$ is the iteration index)
Use P_2 to learn a new similarity sim_2^j by graph transduction when Y_2 is used as the query objects
Add the p nearest neighbors from X_1 to Y_1 based on the similarity sim_1^j to Y_2
Add the p nearest neighbors from X_2 to Y_2 based on the similarity sim_2^j to Y_1
$X_1 = X_1 - Y_1$
$X_2 = X_2 - Y_2$
(Then X_1, X_2 will be unlabeled data for graph transduction in the next iteration)

Fig. 6. Co-transduction algorithm

When the database of known objects is large, computing all the n objects becomes impractical; in practice, we construct similarity matrix w using the first $M << n$ most similar objects to the query x_1 according to the original similarity, which is similar to Yang et al. [3]. Let S denote the first M similar objects to the query x_1. As different shape similarity often have different S, we use S_1 and S_2 to represent the first M similar objects to x_1 according to two kinds of shape similarity respectively. Then the Pseudo code of an efficient version of Co-Transduction algorithm is shown in Fig. 7, which is used in all our experiments. In our experiments, M is always setting as 300.

Theoretical justification

Next, we provide a brief theoretical discussion of our algorithm. We borrow the analysis from [12], which mostly follows the PAC (probably approximately correct) learning theory. Let H_1^0 and H_2^0 be two classifiers (the two transduction algorithms on different metrics in our case) at round 0. They are respectively bounded by generalization errors $a_0 < 0.5$ and $b_0 < 0.5$ with high probability, $1 - \delta$, in PAC. Then H_1^0 selects u number of unlabeled data samples (database objects) and put them into σ_2 for training H_2^1 using transduction. Let l be the number of labeled data and $G = u \times a_0$. If $l \times b_0 \leq e \sqrt[G]{G!} - G$, then

$$Pr[d(H_2^1, H^*) \geq b_1] \leq \delta,$$

Input: a query object x_1 (a labeled data)
the database objects $X = \{x_2, ...x_n\}$ (unlabeled data)
Process:
Create a $M \times M$ probabilistic transition matrix P_1 based on one type of shape similarity with the data from S_1
Create a $M \times M$ probabilistic transition matrix P_2 based on another type of shape similarity with the data from S_2
Create two sets Y_1, Y_2 such that $Y_1 = Y_2 = \{x_1\}$
Create two sets X_1, X_2 such that $X_1 = X_2 = X$
Loop for m iterations:
Use P_1 to learn a new similarity sim_1^j by graph transduction when Y_1 is used as the query objects ($j = 1, ..., m$ is the iteration index)
Use P_2 to learn a new similarity sim_2^j by graph transduction when Y_2 is used as the query objects
Add $N_1 \bigcap S_2$ (N_1 denotes the p nearest neighbors from X_1 to Y_1 based on the similarity sim_1^j) to Y_2
Add $N_2 \bigcap S_1$ (N_2 denotes the p nearest neighbors from X_2 to Y_2 based on the similarity sim_2^j) to Y_1
$X_1 = X_1 - Y_1$
$X_2 = X_2 - Y_2$
(Then X_1, X_2 will be unlabeled data for graph transduction in the next iteration)

Fig. 7. Co-transduction algorithm for a large database

where H^* is the ideal classifier to retrieve all the correct answers, and $d(H_2^1, H^*)$ measures the difference between learned H_2^1 and H^*. The new error is then

$$b_1 = \max[\frac{l \times b_0 + u \times a_0 - u \times d(H_1^0, H_2^1)}{l}, 0].$$

As we can see, the general guidance to achieve a small b_1 is to: (1) reduce the errors of the original learners (good input metrics); (2) increase the complementariness of the metrics. Our algorithm does not necessarily improve the overall performance if the input metrics are not so good at the first place and they are not so different from each other.

From a different perspective, different measures explore different aspects about similarity; the top M most similar objects w.r.t each measure are often not all correct; however, the most similar one (nearest neighbor) is likely be the case; pulling out the best match by one measure to the other helps to further retrieve similar ones by the other complementary measures. This intuition explains why co-transduction works. Our work is also related to the diffusion map [13] which obtains improved similarity measures for clustering by performing Markov random walks. Our transductive learning component improves similarity measures, just like the diffusion map algorithm, and the fusion of different metrics gives further improvement. By exchanging the improved similarity measures of two

transductive learning algorithms, we gradually achieve a fused similarity by letting two originally different measures to meet with each other, which realizes a fusion process. A more detailed theoretical analysis will be left in a longer version.

3 Experimental Results

In this section, we show results on three datasets: MPEG-7 shape dataset [2], Tari's shape dataset [14], and Wei's trademark dataset [15]. In addition, we show our algorithm has a potential to bag-of-feature image search.

3.1 Results on Shape Datasets

The MPEG-7 shape dataset consists of 1400 silhouette images grouped into 70 classes with class having 20 different shapes. Usually the retrieval rate for this dataset is measure by "Bull's eyes test". Every shape in the database is compared to all other shapes, and the number of shapes from the same class among the 40 most similar shapes is reported. The bulls eye retrieval rate is the ratio of the total number of shapes from the same class to the possible number (which is 20×1400). We use the similarities computed by SC [5] and IDSC [6] as the original distance measures. The new similarity obtained by co-transduction resulted in 97.72% on Bull's eyes test, which outperforms existing state-of-the-art algorithms; to further illustrate that our algorithm is independent of specific algorithms, we also use the similarity computed by data-driven general model (DDGM) [16] proposed by Tu and Yuille together with SC or IDSC as the distance measures for co-transduction; we achieve scores 97.45% and 97.31% respectively. These improvements show that the performance gain of our method is general, and not tied to any specific similarity measures. Our results and the scores by several other recent methods on the MPEG-7 dataset are shown in Table 3.1. We observe that co-transduction significantly outperform the alternatives. This demonstrates that integrating different shape similarities is an important direction for shape recognition.

In order to visualize the gain in retrieval rates by our method compared to SC or IDSC , we plot the percentage of correct results among the first k most similar shapes in Fig. 8(a). For example, we plot the percentage of the shapes from the same class among the first k-nearest neighbors for $k = 1, ..., 40$. Recall that each class has 20 shapes and this is the reason for curve $k > 20$. We observe that not only does the proposed method increase the bull's eye score, but also the ranking of the shapes for all $k = 1, ..., 40$ gets improved. In Fig. 8(a), we also plot the curves of retrieval rates for SC/IDSC with graph transduction [3] (eg. SC + LP and IDSC + LP).

Tari's dataset [14] consists of $1,000$ silhouette images grouped into 50 classes with 20 images per class. Tari's dataset has more articulation changes within each class than MPEG-7 dataset as shown in Fig. 9, and consequently IDSC achieved better results than SC on this dataset (see Table 3.1). The retrieval

Table 1. Bull's eyes scores on MPEG-7 dataset [2] and Tari's dataset [14]

Algorithm	MPEG-7 dataset	Tari's dataset
SC [5] (DP)	86.8%	94.17%
IDSC [6]	85.4%	95.33%
DDGM [16]	80.03%	
Planar Graph Cuts [17]	85%	
Shape-tree [18]	87.7%	
Contour Flexibility [19]	89.31%	
IDSC + LP [3]	91%	99.35%
SC + LP [3]	92.91%	97.79%
IDSC + LCDP[9]	93.32%	99.7%
SC + GM + Meta [20]	92.51%	
IDSC + Mutual Graph [4]	93.40%	
SC + IDSC + Co-Transduction	**97.72%**	**99.995%**
IDSC + DDGM + Co-Transduction	**97.31%**	
SC + DDGM + Co-Transduction	**97.45%**	

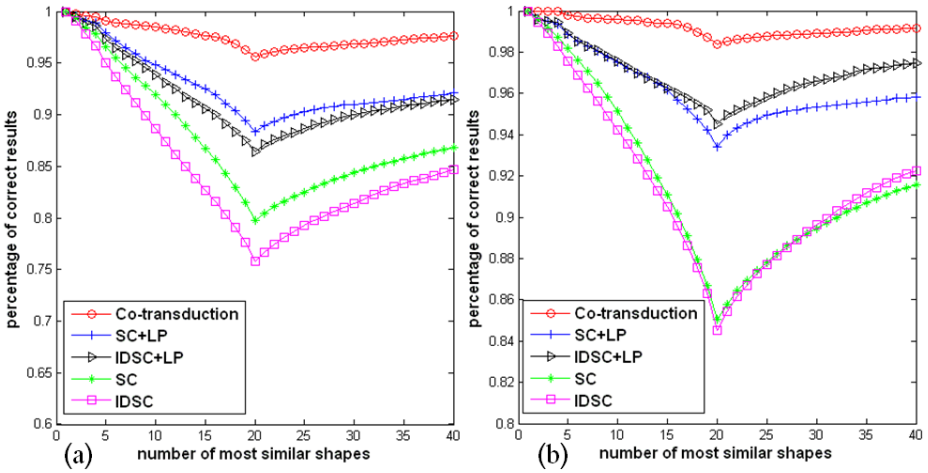

Fig. 8. The curves of retrieval rates for SC, IDSC, SC+LP, IDSC+LP, and Co-Transduction on MPEG-7 shape dataset (a) and Tari's dataset (b)

Fig. 9. Sample images in Tari dataset

performance on this dataset is also measured by "Bull's eyes test". Only one error was made when retrieving all the shapes from the dataset, which means we achieve nearly perfect retrieval rate: 99.995%. Table 3.1 also lists several results of Tari's dataset in comparison with other approaches; we observe that the second highest result by IDSC+LCDP [9] is 99.7% with 60 errors. Same as in Fig. 8(a), the retrieval curves in Fig. 8(b) are plotted to clearly show the performance gain by co-transduction algorithm.

3.2 Results on Trademark Images

We also tested our method on a trademark dataset [15] consisting of 14 different classes with $1,003$ trademark images in all. Fig. 10 shows typical some examples from the trademark dataset. To evaluate the performance of trademark retrieval, we use the precision-recall curves. The x-axis and y-axis represent recall and precision rates, respectively. Precision is the ratio of the number of relevant images retrieved to the total number of images retrieved while recall is the number of relevant images to the total number of relevant images stored in the database. For each query image input to the system, the system returns 11 pages of hits with descending similarity rankings, each page containing nine trademark images. This allows the performance of our system to be evaluated on a page-wise manner. Since there are only five classes containing more than 99 images, we only report the precision-recall graph on these five classes. Each curve consists of 11 data points, with the ith point from the left corresponding to the performance when the first i pages of hits are taken into consideration. A precision-recall line stretching longer horizontally and staying high in the graph indicates that the corresponding algorithm performs relatively better. Here, we use two distance measures: moment invariants [21], Zernike [22], which are two kinds of region-based shape features. Then we use our method on these two distance measures. In Fig. 11, the data points shown on the curve for co-transduction are the average precisions and recalls over the five classes. The curves shows that our method can improve the performance of trademark retrieval significantly, which also prove that co-transduction algorithm is good fit for trademark images and different shape distance measures.

Fig. 10. Sample images in Wei's trademark dataset

3.3 Improving Bag-of-Features Image Search with Co-transduction

In this section we show that co-transduction can improve the accuracy of image search. Bag-of-features image representation [23,24] is usually suggested for image search problem. Recently, Jegou et al. [25] proposed a distance learning method called contextual dissimilarity measure (CDM), which can significantly

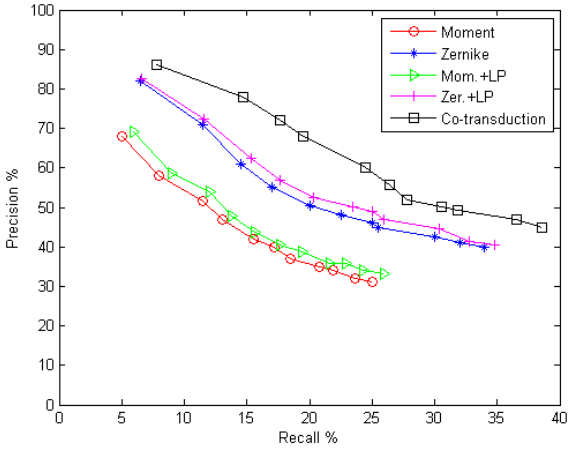

Fig. 11. The precision/recall curves for trademark images

Fig. 12. Sample images of N-S dataset [26]

improve the similarity computed by bag-of-features. We compare our method with CDM on the Nistér and Stewénius (N-S) dataset [26]. The N-S dataset consists of 2,550 objects or scenes, each of which is imaged from 4 different viewpoints. Hence the dataset has 10,200 images in total. A few example images from N-S dataset are shown in Fig. 12.

We adopt the method in [25] to compute the similarity for image search. The image descriptor is a combination of Hessian-Affine region detector [27] and SIFT descriptor [28]. A visual vocabulary is obtained using the k-means algorithm on the sub-sampled image descriptors. As co-transduction requires two kind of similarity measures, we proposed a new similarity named **reverse similarity** based on the one by [25]. Let $w_{i,j}$ denote the similarity between objects i and j computed by [25], the reverse similarity $w^r_{i,j} = \frac{1}{d^\beta}$, where d is the ranking number of i when using j as a query for the dataset, and β is a weight factor setting with a constant 10. Reverse similarity is motivated the phenomenon pointed out by [25]: a good ranking is usually not symmetrical in image search, which tells us two objects can be very likely from the same category when they both obtain a good ranking position when using each other as a query. With w and w^r, we can apply co-transduction to image search on N-S dataset, and the measure score is the average number of correct images among

Table 2. The results on N-S dataset

number of distinct visual vocab.	vocab. size	original N-S score	N-S score with CDM	N-S score with co-transduction
1	30000	3.26	3.57	3.66

the four first images returned. Table 3.3 lists the results on N-S dataset. We can observe that co-transduction significantly increases the score from 3.26 to 3.66, which is also better than CDM's result when the number visual vocabulary is 1 and vocabulary size is 30000. Our result demonstrates that co-transduction is also able to improve the performance of image search problem.

3.4 The Parameter Setting and Discussion

As introduced in [29], there are two key parameters for label propagation: α and K. Beside α and K, there are two additional parameters for co-transduction: the iteration number m and the number of nearest neighbors p. For the MPEG-7 and Tari's dataset, we use the following parameter settings: $\alpha = 0.25$, $K = 14$, (which are consistent with the setting in [29]), $m = 4$, and $p = 3$. For the trademark dataset, since the input distance measures are different from the ones for MEPG-7 dataset, the parameter setting is $\alpha = 8$, $K = 8$, $m = 2$, and $p = 2$. For the N-S dataset, the parameters are $\alpha = 0.25$, $K = 10$, $m = 3$, and $p = 1$. Since [29] has introduced a supervised learning method for determining the parameters α and K in details, we no longer review it here. We only need to focus on m and p. As both m and p are integer, the values of them are very easily to set. Table 3.4 shows the scores on MPEG-7 dataset when setting m, p with the integers from 1 to 5. We observe that all these scores are around 97%, which demonstrates the insensitiveness of co-transduction to parameter tuning.

Now we want to discuss why co-transduction is essential. We iteratively run LP on MPEG-7 dataset based on only one type of similarity with the same parameter setting for co-transduction (the p most similar objects will be added into the query set for the next iteration), and we get the bull's eyes scores 92.68% and 91.79% based on SC and IDSC respectively. Compared to the LP's results in Table 3.1, there is not so much change. Let sim'_{SC} and sim'_{IDSC} denote the similarities obtained in the above experiments. We obtain a new similarity sim'_c by linearly combining sim'_{SC} and sim'_{IDSC} as follows: $sim'_c =$

Table 3. The bull's eyes scores on MEPG-7 dataset with different parameter setting

	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
$p = 1$	96.89%	97.05%	97.30%	97.32%	97.34%
$p = 2$	97.06%	97.24%	97.36%	97.45%	97.36%
$p = 3$	97.20%	97.54%	97.63%	97.72%	97.67%
$p = 4$	97.13%	97.30%	97.42%	97.37%	97.32%
$p = 5$	97.24%	97.55%	97.58%	97.20%	96.92%

$\lambda sim'_{SC}+(1-\lambda)sim'_{IDSC}$, where λ is a weight factor. We tuned λ, and the highest score based on sim'_c is 92.0% when λ is 0.9. These results are much lower than the ones by co-transduction, and this illustrates that the performance achieved by co-transduction can not be reached by simply combining the similarities.

4 Conclusion

We have proposed a shape retrieval framework named co-transduction which combines two (our algorithm is actually not limited to just two) different distance metrics. The significant performance improvement on four large datasets has demonstrated the effectiveness of co-transduction for shape/object retrieval.Our future work includes the extension to other problems and providing deeper understanding of the approach.

Acknowledgement

We thank Herve Jegou for the help on the experiments on N-S dataset. This work was jointly supported by NSFC 60903096, NSFC 60873127, ONR N000140910099, NSF CAREER award IIS-0844566, and China 863 2008AA01Z126.

References

1. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proc. of COLT, pp. 92–100 (1998)
2. Latecki, L., Lakámper, R., Eckhardt, U.: Shape descriptors for non-rigid shapes with a single closed contour. In: Proc. of CVPR, pp. 424–429 (2000)
3. Yang, X., Bai, X., Latecki, L., Tu, Z.: Improving shape retrieval by learning graph transduction. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 788–801. Springer, Heidelberg (2008)
4. Kontschieder, P., Donoser, M., Bischof, H.: Beyond pairwise shape similarity analysis. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) Computer Vision – ACCV 2009. LNCS, vol. 5996. Springer, Heidelberg (2010)
5. Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Trans. PAMI 24, 522–705 (2002)
6. Ling, H., Jacobs, D.: Shape classification using the inner-distance. IEEE Trans. PAMI 29, 286–299 (2007)
7. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. IEEE Trans. PAMI 25, 116–125 (2004)
8. Zhu, X.: Semi-supervised learning with graphs. In: Doctoral Dissertation, Carnegie Mellon University, CMU–LTI–05–192 (2005)
9. Yang, X., Koknar-Tezel, S., Latecki, L.: Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In: Proc. of CVPR (2009)
10. Zhou, Z.H., Li, M.: Semi-supervised regression with co-training. In: Proc. of IJCAI (2004)

11. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. J. of Machine Learning Research 7, 2399–2434 (2006)
12. Wang, W., Zhou, Z.H.: Analyzing co-training style algorithms. In: Kok, J.N., et al. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 454–465. Springer, Heidelberg (2007)
13. Coifman, R., Lafon, S.: Diffusion maps. Applied and Comp. Harmonic Ana. (2006)
14. Aslan, C., Erdem, A., Erdem, E., Tari, S.: Disconnected skeleton: Shape at its absolute scale. IEEE Trans. PAMI 30, 2188–2201 (2008)
15. Wei, C.H., Li, Y., Chau, W.Y., Li, C.T.: Trademark image retrieval using synthetic features for describing global shape and interior structure. Pattern Recognition 42, 386–394 (2009)
16. Tu, Z., Yuille, A.L.: Shape matching and recognition - using generative models and informative features. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3023, pp. 195–209. Springer, Heidelberg (2004)
17. Schmidt, F.R., Toeppe, E., Cremers, D.: Efficient planar graph cuts with applications in computer vision. In: Proc. of CVPR (2009)
18. Felzenszwalb, P.F., Schwartz, J.: Hierarchical matching of deformable shapes. In: CVPR (2007)
19. Xu, C., Liu, J., Tang, X.: 2d shape matching by contour flexibility. IEEE Trans. PAMI 31, 180–186 (2009)
20. Egozi, A., Keller, Y., Guterman, H.: Improving shape retrieval by spectral matching and meta similarity. IEEE Trans. Image Processing 19, 1319–1327 (2010)
21. Gonzalez, R., Woods, R., Eddins, S.: Digital image processing using matlab. Prentice-Hall, EnglewoodCliffs (2004)
22. Kim, Y.S., Kim, W.Y.: Content-based trademark retrieval system using a visually salient feature. Image and Vision Computing 16, 931–939 (1998)
23. Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: Proc. CVPR, pp. 2161–2168 (2006)
24. Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: Proc. ICCV, pp. 1470–1477 (2003)
25. Jegou, H., Schmid, C., Harzallah, H., Verbeek, J.: Accurate image search using the contextual dissimilarity measure. IEEE Trans. PAMI 32, 2–11 (2010)
26. Stewénius, H., Nistér, D.: Object recognition benchmark, `http://vis.uky.edu/%7Estewe/ukbench/`
27. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. IJCV 60, 63–86 (2004)
28. Lowe, D.: Distinctive image features from scale-invariant key points. IJCV 60, 91–110 (2004)
29. Bai, X., Yang, X., Latecki, L., Liu, W., Tu, Z.: Learning context sensitive shape similarity by graph transduction. IEEE Trans. PAMI 32, 861–874 (2010)