Bi-Similarity Mapping Based Image Retrieval Using Shape Features

P. Jemimah Jeba Priya¹, T. Nalini² and M. Padmavathy³

¹PG student, Department of Computer Science and Engineering, Bharath University, Chennai
²Professor, Department of Computer Science and Engineering, Bharath University, Chennai
³Research Scholar, Bharath University, Chennai

ABSTRACT

Shape based image retrieval is a very active research area in image processing. Effective and efficient retrieval of similar images from large image databases is still a challenging problem despite the high relevance that shape information can have in describing image contents. The goal is to develop an effective retrieval system to overcome a few limitations associated with existing systems. This paper proposes Co-Transduction algorithm that is a shape based image retrieval algorithm, which is to combine different similarity measures for robust shape based image retrieval through a semi-supervised learning framework. The accuracy of adopted similarity measures (distances or metrics) decides the performance of a retrieval system. In shape based image retrieval, intra-class shapes should have smaller distances than inter-class shapes. Given two similarity measures and a query shape, the algorithm iteratively retrieves the most similar images using one measure and assigns them to a pool for the other measure to do a reranking, and vice versa.

Keywords: Shape Based Image Retrieval, Shape matching, Shape Context, Inner Distance Shape Context

1. INTRODUCTION

Research in content-based image retrieval today is a lively discipline, expanding in breadth. At the current stage of content-based image retrieval research, it is interesting to look back toward the beginning and see which of the original ideas have blossomed, which haven’t and which were made obsolete by the changing landscape of computing.

Shape is a key feature for computer vision applications. For example, contour-based object recognition methods [1] have recently shown to outperform appearance-based methods, because shape is a strong feature for recognition as psychophysical studies [4] have shown. Object contours are invariant to extreme illumination conditions and large variations in texture or color and for some categories shape is more generic than appearance. Automated comparison and grouping of shapes is very often the basis in the areas of human detection [5] or action recognition [6]. Hence in this paper we use shape as a key feature for image retrieval.

The key part in all these applications is a robust and efficient shape matching method, which allows quantifying the similarity between two input shapes. The calculated similarity scores are the basis for different tasks, like retrieving most similar shapes, identifying clusters within databases or ending exemplary shape prototypes. These tasks are complicated by the fact that similarities of shapes do not lie in a metric space. Therefore Euclidean analysis is not sufficient for handling shape similarities. State-of-the-art shape matching performs pair wise analysis and ignores the fact that distances between all other shapes contain important information about the shape manifold. Therefore, recently some effort was put on post-processing the obtained similarity scores by analyzing the estimated similarities between all given shapes to increase the discriminability between different shape groups as e.g. in [7, 8].

Shape of the objects represented in images is one of the most significant properties used in CBIR and in recognition tasks. This is particularly due to the fact that shape is perceptually very relevant in order to recognize objects. Content-based image retrieval (CBIR) work includes selection, object representation, and matching. If a shape is used as feature, edge detection might be the first step of feature extraction. Invariance to translation, rotation, and scale is required by a good shape representation. Sustaining deformation contour matching is an important issue at the matching process. There are four important feature components for content-based image retrieval: color, texture, shape, and spatial relationship. Among these features, shape contains the most attractive visual information for human perception. In shape description, features are generally classified into two types. One is the contour-based shape feature, and the other is the region-based shape feature. Contour-based shape feature is the feature extracted from the shape boundary points only, while region-based shape feature is the feature extracted from the shape interior points. An important step before shape extraction is edge point detection.
2. RELATED WORK

Given a query shape, the most similar images 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 of critical importance 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. Ideally, such a similarity measure should result in smaller distances between the variants of a particular object than this object to any other ones, as well as smaller distances between intra class objects than interclass objects. The following Fig. 1 gives an illustration how to extract appropriate feature vectors to represent image content correctly, and the other is how to carry out the image retrieval based on the extracted feature vectors effectively.

![Fig. 1: A horse in (a) may look more similar to a dog in (b) than to another horse in (c)](image)

The semi-supervised learning problem has attracted an increasing amount of interest recently, and several novel approaches have been proposed. In this section, we discuss some of the previous efforts that are related to the approach proposed in this paper. The various approaches for shape based image retrieval in literature have the irrespective advantages and limitations with respect to the kind of input they can handle, computational complexity, etc. 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.

**Table 1:** Retrieval rates of different existing algorithms for the MPEG7 CE-shape-1 database.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MPEG-7</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC[1]</td>
<td>86.8%</td>
</tr>
<tr>
<td>IDSC[3]</td>
<td>85.4%</td>
</tr>
<tr>
<td>DDGM[2]</td>
<td>80.03%</td>
</tr>
<tr>
<td>IDSC + CDM[6]</td>
<td>88.30%</td>
</tr>
<tr>
<td>IDC + LP[9]</td>
<td>91%</td>
</tr>
<tr>
<td>SC + LP[9]</td>
<td>92.91%</td>
</tr>
<tr>
<td>SC + GM + Meta[5]</td>
<td>92.51%</td>
</tr>
<tr>
<td>IDSC + Mutual Graph[13]</td>
<td>93.40%</td>
</tr>
</tbody>
</table>

Since a large number of shape similarity methods have been proposed in the literature, we focus our attention on methods that reported retrieval results on the MPEG-7 shape data set (part B of the MPEG-7 Core Experiment CE-Shape-1 [8]. The results in Table 1 were reported in the following corresponding references. This allows us to clearly demonstrate the retrieval rate improvements obtained by the proposed method.


Bai et al. [9] explored the group contextual information on different shapes to improve the efficiency of shape retrieval on several standard data sets [10], [11]. The basic idea was to use shapes as each other’s contexts in propagation to reduce the distances between intra class objects. The implementation was done by a graph-based transduction approach, named label propagation (LP) [12].
Later, several other graph-based transduction methods were suggested for shape retrieval [13], [14]. Egozi et al. [5] proposed a contextual similarity function, named Meta similarity, which characterizes a given object by its similarity to its k-nearest neighbor (k-NN) objects.

An interesting distance learning method called contextual dissimilarity measure (CDM) [6] is motivated by an observation that a good ranking is usually not symmetrical in image search, which is mainly designed for the image search problem. CDM significantly improves the distance measure using bag-of-features; however, its improvement on shape retrieval is not as obvious as the shape distance measures have different properties than bag-of-features.

In the context of image retrieval and shape similarity, several shape descriptors have been proposed, ranging from moments and Fourier descriptors to Hausdorff distance and the medial axis transform. It should be emphasized that our approach is generically applicable as opposed to most shape matching techniques that are restricted to silhouettes and closed curves. In our framework shape refers to any type of boundary information and in consequence, our algorithm is applicable for a large variety of recognition problems.

At its core, shape contexts can be understood as a point set matching technique. Most closely related is the work of [3] which proposes an iterative optimization algorithm to jointly determine point correspondences and underlying image transformations, where typically some generic transformation class is assumed. This formulation leads to a difficult non-convex optimization problem which is solved using deterministic annealing.

Shape contexts will greatly simplify the matching part, leading to a very robust point registration technique. It is invariant to scale and translation and to a large extent robust to rotation and deformation. Extension incorporating rotational invariance and local appearance features may be found.

3. PROPOSED SYSTEM

Our shape based image retrieval system consists of database construction part and image retrieval. The database construction part is intended to ensure high retrieval efficiency by extracting a feature set for each of the images and storing the feature set along with its corresponding image in the database.

To access the database, the user initiates the image retrieval process by providing a query image as input, and then the system starts with extracting the features from the query image as shown in Fig.2. Afterwards, the system measures the similarity between the feature set of the query image and those of the images stored in the database through Bi-Similarity measure process. Finally, the system ranks the relevance based on the similarity and returns the results to the user.

This paper provides a different way of fusing similarity/distance measures through a semi supervised learning framework. The user input is a query shape, and our system returns the most similar images by effectively integrating two distance metrics computed by different algorithms, e.g., shape contexts (SC) [1] and inner-distance shape contexts (IDSC) [3]. Our approach is inspired by the co-training algorithm [4], which assumes views (sets of features) with two conditions: 1) Each view is strong enough to describe the data (a good classifier can be learned based on enough training samples); and 2) each view is conditionally independent given the labels.

However, unlike co-training, in which two independent views (sets of features) are assumed, our algorithm deals with single-view but multiple-input similarities; we deal with the retrieval/ranking, whereas co-training is focused on the classification problem.
4. Experiments and Results

The feature extraction has to be done in both database construction and image retrieval processes. Feature extraction is about extracting from an image a set of attributes/features that can feasibly describe/represent the image in order to facilitate the ensuing feature matching and ranking processes. The feature extraction stage in the proposed method involves image pre-processing and feature representation.

The purpose of the image pre-processing module is to detect the edges in order to pave the way for the feature representation module. The task of the feature representation module is to extract a set of feature descriptors (also called feature vector) of the images from their corresponding edge maps generated by the image pre-processing module.

4.1. Image Pre-processing

Images come in different sizes and shapes and their key components appear in different locations. An effective retrieval system should be invariant to the scaling, rotation and translation of images. To achieve these objectives, the proposed CBIR system performs the following steps to normalize each image Canny edge detector [7] to find the edge points in an image. The resulting edge map represents the contour of the underlying image.

Canny Edge Detection Algorithm

The canny edge detection algorithm was developed to detect edge lines and gradients for the purpose of image processing. This algorithm provides good detection and localization of real edges while providing minimal response in low noise environments. The main stages of the Canny Algorithm are as follows:

1. **Smoothing:** Blurring of the image to remove noise.
2. **Finding gradients:** The edges should be marked where the gradients of the image has large magnitudes.
3. **Non-maximum suppression:** Only local maxima should be marked as edges.
4. **Double thresholding:** Potential edges are determined by thresholding.
5. **Edge tracking by hysteresis:** Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

The main stages are explained below:

1. **Smoothing:**

   It is inevitable that all images taken from a camera will contain some amount of noise. To prevent that noise is mistaken for edges, noise must be reduced. Therefore the image is first smoothed by applying a Gaussian filter. The kernel of a Gaussian filter with a standard deviation of $\sigma = 1.4$ is shown in Equation (1).

   $B = \frac{1}{128} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$

   (1)

2. **Finding gradients:**

   The Canny algorithm basically finds edges where the greyscale intensity of the image changes the most. These areas are found by determining gradients of the image. Gradients at each pixel in the smoothed image are determined by applying what is known as the Sobel-operator. The gradient magnitudes (also known as the edge strengths) can then be determined as an Euclidean distance measure by applying the law of Pythagoras as shown in Equation (3). It is sometimes simplified by applying Manhattan distance measure as shown in Equation (4) to reduce the computational complexity. The Euclidean distance measure has been applied to the test image. The computed edge strengths are compared to the smoothed image in Figure 3.

   $|G| = \sqrt{G_x^2 + G_y^2}$

   (3)

   $|G| = |G_x| + |G_y|$

   (4)

   Where $G_x$ and $G_y$ are the gradients in the x and y directions respectively.
It is obvious from Figure 3, that an image of the gradient magnitudes often indicate the edges quite clearly. However, the edges are typically broad and do not indicate exactly where the edges are. To make it possible to determine this, the direction of the edges must be determined and stored as shown in Equation (5).

\[ \theta = \arctan \left( \frac{|G_x|}{|G_y|} \right) \]  

(5)

3. Non-maximum suppression:

The purpose of this step is to convert the “blurred” edges in the image of the gradient magnitudes to “sharp” edges. Basically this is done by preserving all local maxima in the gradient image, and deleting everything else. The algorithm is for each pixel in the gradient image:

1. Round the gradient direction \(\theta\) to nearest 45\(^\circ\), corresponding to the use of an 8-connected neighbourhood.
2. Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient direction. I.e. if the gradient direction is north (\(\theta = 90^\circ\)), compare with the pixels to the north and south.
3. If the edge strength of the current pixel is largest; preserve the value of the edge strength. If not, suppress (i.e. remove) the value.

A simple example of non-maximum suppression is shown in Fig. 3. Almost all pixels have gradient directions pointing north. They are therefore compared with the pixels above and below. The pixels that turn out to be maximal in this comparison are marked with white borders. All other pixels will be suppressed.

![Fig. 3: Illustration of non-maximum suppression. The edge strengths are indicated both as colors and numbers, while the gradient directions are shown as arrows. The resulting edge pixels are marked with white borders.](image)

4. Double thresholding:

The edge-pixels remaining after the non-maximum suppression step are (still) marked with their strength pixel-by-pixel. Many of these will probably be true edges in the image, but some may be caused by noise or color variations for instance due to rough surfaces. The simplest way to discern between these would be to use a threshold, so that only edges stronger that a certain value would be preserved. The Canny edge detection algorithm uses double thresholding. Edge pixels stronger than the high threshold are marked as strong; edge pixels weaker than the low threshold are suppressed and edge pixels between the two thresholds are marked as weak.

5. Edge tracking by hysteresis:

Strong edges are interpreted as “certain edges”, and can immediately be included in the final edge image. Weak edges are included if and only if they are connected to strong edges. The logic is of course that noise and other small variations are unlikely to result in a strong edge (with proper adjustment of the threshold levels). Thus strong edges will (almost) only be due to true edges in the original image. The weak edges can either be due to true edges or noise/color variations. The latter type will probably be distributed independently of edges on the entire image, and thus only a small amount will be located adjacent to strong edges. Weak edges due to true edges are much more likely to be connected directly to strong edges.

Edge tracking can be implemented by BLOB-analysis (Binary Large OBject). The edge pixels are divided into connected BLOB’s using 8-connected neighbourhood. BLOB’s containing at least one strong edge pixel is then preserved, while other BLOB’s are suppressed.

Implementation of Canny Edge Detection:

As noted in Section 1, all images in this worksheet (except the original) are produced by our implementation. A few things should be noted with regards to this:

1. The (source) image and the thresholds can be chosen arbitrarily.
2. Only a smoothing filter with a standard deviation of \(\sigma = 1.4\) is supported (the one shown in Equation 1).
3. The implementation uses the “correct” Euclidean measure for the edge strengths.
4. The different filters cannot be applied to edge pixels. This causes the output image to be 8 pixels smaller in each direction.

The last step in the algorithm known as edge tracking can be implemented as either iterative or recursive BLOB analysis [4]. A recursive implementation can use the grass-fire algorithm. However, our implementation uses the iterative approach. First all weak edges are scanned for neighbour edges and joined into groups. At the same time it is marked which groups are adjacent. Then all of these markings are examined to determine which groups of weak edges are connected to strong edges (directly or indirectly). All weak edges that are connected to strong edges are marked as strong edges themselves. The rest of the weak edges are suppressed. This can be interpreted as BLOB analysis where only BLOB’s containing strong edges are preserved (and considered as one BLOB).

4.2. Bi-Similarity Measure Process

This section deals with feature matching, which is about measuring the similarity between the feature vectors of the query image and the database images. Using co-transduction algorithm, the system returns the most similar images by effectively integrating two distance metrics computed by different algorithms such as shape contexts (SC) and inner-distance shape contexts (IDSC).

The following gives the pseudo codes for Co-Transduction algorithm. Same as in Bai et al. [9], a query shape $x_1$ 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. It uses one metric to pull out confident data for the other metric to refine the performance. The final similarity $\text{sim}_f$ of co-transduction is the average of all the similarities, i.e.,

$$\text{sim}_f = (1/2m) \sum_{j=1}^{m} (\text{sim}_1^j + \text{sim}_2^j)$$

(6)

For $j = 1,\ldots,m$ is the iteration index where $\text{sim}_1^j$ is the similarity measure of shape by using SC and $\text{sim}_2^j$ is the similarity measure of shape by using IDSC.

Algorithm: Co-Transduction algorithm for large databases

**Input:** A Query shape $x_1$ (a label data) the database shapes $X = \{x_2,\ldots,x_n\}$ (an unlabeled data)

**Process:**
Create a M x M probabilistic transition matrix $P_1$ based on one type of shape similarity with the data from $S_1$.
Create a M x 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 $\text{sim}_1$ by graph transduction when $Y_1$ is used as the query objects ($j=1,\ldots,m$ is the iteration index).
Use $P_2$ to learn a new similarity $\text{sim}_2$ by graph transduction when $Y_2$ is used as the query objects.
Add $N_1_i \cap S_2$ ($N_1$ denotes the $p$ nearest neighbours from $X_1$ to $Y_1$ based on the similarity $\text{sim}_1^j$ to $Y_2$)
Add $N_2_i \cap S_1$ ($N_2$ denotes the $p$ nearest neighbours from $X_1$ to $Y_1$ based on the similarity $\text{sim}_2^j$ 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)

When the database of known objects is large, computing all $n$ shapes becomes impractical; in practice, we construct similarity matrix using the first $M << n$ most similar shape to query according to the original similarity, which is similar to Bai et al. [9]. Let $S$ denote the first $M$ similar shapes to query $x_1$. As different shape similarities often have different $S$, we use $S_2$ and to represent the first $M$ similar objects to $x_1$ according to two kinds of shape similarity, respectively.

Thus this method integrates two distance metrics computed by two different algorithms for retrieving the top most similar images. Using Bi-Similarity measure, an improved result on the MPEG-7 data set was achieved.

5. Conclusion

Several methods have been proposed for retrieving the image based on shape by using similarity measure. The existing retrieving techniques identify the low accuracy results on the MPEG-7 data set and few of them are applied to the small databases. So, to overcome this drawback the algorithm called Co-Transduction is proposed. We have presented novel shape retrieval and clustering method that, applied on top of any existing shape matching algorithm,
significantly improve results in shape retrieval and clustering. Since our approach is generic and can be applied to any similarity matrix, we plan to further evaluate our method on standard machine learning databases.

References