Pattern Recognition 41 (2008) 3179 – 3191

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Image categorization: Graph edit distance + edge direction histogram

Xinbo Gao [a], Bing Xiao [a], Dacheng Tao [b], Xuelong Li [c,*]

[a] School of Electronic Engineering, Xidian University, Xi'an 710071, PR China
[b] Biometrics Research Centre, Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, PR China
[c] School of Computer Science and Information Systems, Birkbeck College, University of London, London WC1E 7HX, UK

ARTICLE INFO

Article history:
Received 1 January 2008
Received in revised form 15 March 2008
Accepted 18 March 2008

Keywords:
Inexact graph matching
Graph edit distance (GED)
Edge direction histogram (EDH)
Earth mover's distance (EMD)
Image categorization

ABSTRACT

This paper presents a novel algorithm for computing graph edit distance (GED) in image categorization. This algorithm is purely structural, i.e., it needs only connectivity structure of the graph and does not draw on node or edge attributes. There are two major contributions: (1) Introducing edge direction histogram (EDH) to characterize shape features of images. It is shown that GED can be employed as distance of EDHs. This algorithm is completely independent on cost function which is difficult to be defined exactly. (2) Computing distance of EDHs with earth mover distance (EMD) which takes neighborhood bins into account so as to compute distance of EDHs correctly. A set of experiments demonstrate that the newly presented algorithm is available for classifying and clustering images and is immune to the planar rotation of images. Compared with GED from spectral seriation, our algorithm can capture the structure change of graphs better and consume 12.79% time used by the former one. The average classification rate is 5% and average clustering rate is 25% higher than the spectral seriation method.

1. Introduction

When we get images of real world objects, the acquirement is often affected by noise and distortion. Graph representations for identical objects may not match exactly, so integration of error correction into the matching process is necessary; thus inexact graph matching [1–5] has been the focus of research in the areas of computer vision and structural pattern recognition for over two decades. One of its key issues is the similarity measurement between pairwise graphs. Among many kinds of approaches for measuring graph similarity, graph edit distance (GED) has attracted researchers' attention greatly because of its toleration to noise and distortion. GED between two graphs is defined as the least cost of edit operations that are needed to transform a graph into another one.

Initially, Sanfeliu and Fu [6] introduced edit distance into graph in 1980s, which is computed by counting node and edge relabelings together with the number of node and edge deletions and insertions necessary to transform a graph into another, and then GED received significant attention. It can be computed directly. Extending the idea of Sanfeliu and Fu [6], Messmer and Bunke defined the subgraph edit distance by the minimum cost for all error-correcting subgraph isomorphisms [7,8], in which common subgraphs of different model graphs are represented only once and the limitation of inexact graph matching algorithms working on only two graphs once can be avoided. There have been attempts to extend the edit distance to trees and graphs. But direct GED algorithms lack some of the formal underpinning of string edit distance, so there is considerable current effort to make the underlying methodology rely on a rigorous footing. There have been some development for overcoming this drawback, for instance, the relationship between GED and the size of the maximum common subgraph has been demonstrated [9], the uniqueness of the cost function is commented [10], a probability distribution for local GED has been constructed, and graphs can be converted into strings first, etc. As mentioned above, GED can be computed indirectly with the idea of string alignment after graphs are converted into strings. For non-attributed graphs, GED is usually computed in this way. As a result, the role of edit distance [5,11] cannot be neglected in the development of GED, which is used to compare coded patterns of graphs and promotes the birth of new GED algorithms. Hancock et al., used Levenshtein distance, an important kind of edit distance, to evaluate the similarity of pairwise strings which are derived from graphs [12]. The edit distance between strings can also be evaluated by dynamic programming [5], which has been extended to compare trees and graphs on a global level [13,14]. Recently, Marzal and Vidal normalized the edit distance so that it may be consistently applied across a range of objects in different size [15] and this idea has been used to model the probability distribution for edit path between pairwise graphs [16]. The Hamming distance between two strings is another special case of the edit distance, with which Hancock [17] measures the GED between

* Corresponding author.
E-mail address: xuelong@dcs.bbk.ac.uk (X. Li).

0031-3203/$30.00 © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.03.025

structural units of graphs together with the size difference between graphs.

Although the research of GED has been developed flourishingly, most of the existing algorithms are very dependent on cost function and similarity criterion of corresponding nodes and edges in two graphs which are difficult to be defined reasonably. In order to avoid these unsolved problems, we combine the EDH and EMD for computing GED. Edit operation sequence consists of edge/node insertion, edge/node deletion and edge/node substitution. Node operations are involved in edge operations; thus the GED is related to the connectivity difference of graphs, i.e., edge direction and edge length. While edge direction histogram (EDH) is adept at characterizing these features well, so the GED of graphs is converted into distance of EDHs, which is computed with earth mover distance (EMD). This method is completely independent of cost function and similarity criterion. It can be proved that it is available for classifying and clustering images and is invariable with image planar rotation. From Section 2 and 4, EDH, EMD and this new algorithm are introduced in detail, respectively; in Section 5, a set of experiments is presented to show the performance of this method; complexity of the algorithm is analyzed in Section 6; conclusion is given in Section 7.

2. Edge direction histogram

EDH is computed by grouping the edge pixels which fall into edge directions and counting the number of pixels in each direction. Given an image, its EDH computation steps are represented in Fig. 1. Edge points are extracted by edge detection operator and each of them can be represented with the vector $\vec{p}_i = \{dx_i, dy_i\}$, where dx_i and dy_i are, respectively, horizontal and vertical differences of the point. Each point's edge direction (i.e., gradient direction) is calculated with the equation $ang_i = \arctan(dy_i/dx_i)$, according to which discrete directions expected are specified. Each of these discrete directions corresponds to a bin in histogram. Edge direction of each point is quantified as one of the discrete directions, that is to say, each point falls into a bin. Finally, the number of edge points falling into the same bin is accumulated.

Since edge points are related to shape information closely, EDH is a very simple and direct way to characterize shape information of

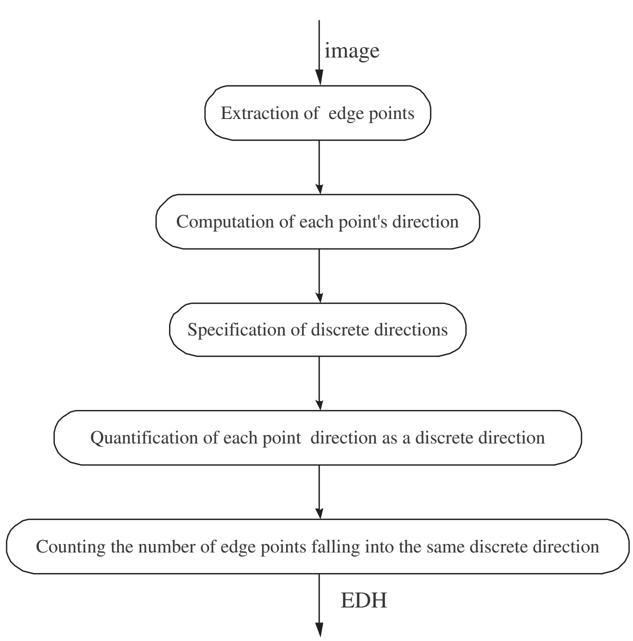

Fig. 1. The flow chart of EDH computation.

an object. It has been applied successfully to image retrieval [18–22], classification [23] and quality assessment [24]. In addition, Kim used EDH to watermarking text document images [25] based on the idea that sub-images have similar-shaped EDHs. EDH is usually normalized to be scaling invariant, but Zhang et al. [26] compute the 1-D FFT of the normalized EDH to obtain rotation invariance and take it as the final signature of image. In a word, with the help of EDH, high-level pattern recognition problem is to be solved with relatively simple low-level features. However, EDH is seldom used to graph matching. The EDH for a graph is computed according to the slope and length of edges; thus it can reflect the connectivity information of a graph directly and sufficiently, for example, the EDH for a graph is shown in Fig. 2. So, we intend to measure the graph similarity based on the EDHs of graphs.

In Fig. 2, a graph is shown as the left figure and points on each edge are of the same direction which is the degree of edges (arc tangent of slopes). EDH of this graph is shown in the right one in which x-axis denotes the centers of bins, that is the degree of edges, and y-axis denotes the number of pixels falling into each bin. Correspondence of edges and bins is explained in Table 1. The sum of edge (a, d)'s length and edge (b, c)'s length is the largest, so the bin centered on 0 arc corresponds to the most pixels. The length of edge (a, b) is the least and the bin centered on $-\pi/2$ arc corresponds to the fewest pixels. In a word, number of pixels falling into a bin coincides with the length of the edge.

3. Earth mover's distance

EMD [27] is required to address the dissimilarity measure between two signatures and solved based on the transportation simplex method. Its idea is the minimal amount of work that must be performed to transform one signature into the other one by moving "distribution mass" around. It has been applied widely to image retrieval [28], image clustering and text clustering [27], shape matching [29,30], query of video [31], etc.

For two signatures:

$$X = \{(x_1, w_1), (x_2, w_2), \ldots, (x_n, w_n)\} \quad \text{and}$$
$$Y = \{(y_1, w'_1), (y_2, w'_2), \ldots, (y_m, w'_m)\},$$

where (x_i, w_i) represents that the distributed mass at position x_i is w_i and the same to (y_j, w'_j), if transformation from distribution X into Y is performed, the computation of EMD is a linear programming problem and can be described as below:

The objective function is $\min_{F=\{f_{ij}\}}\{\sum_{i=1}^{n}\sum_{j=1}^{m} d_{ij}f_{ij}\}$.

Subject to the following constraints:

(1) $f_{ij} \geqslant 0$, where $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m$,
(2) $\sum_{j=1}^{m} f_{ij} \leqslant w_i$,
(3) $\sum_{i=1}^{n} f_{ij} \leqslant w'_j$,
(4) $\sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij} = \min(\sum_{i=1}^{n} w_i, \sum_{j=1}^{m} w'_j)$.

In the formulas above, d_{ij} represents the cost of removing unit mass from i to j and can be set to be $d_{ij} = |x_i - y_j|^2$, and f_{ij} denotes the quantity of mass removed from i to j. According to optimal F, EMD is defined as: $EMD(X, Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} d_{ij}f_{ij} / \sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij}$.

4. GED based on EDH

Our algorithm computes GED on the basis of EDH. EDHs of graphs which are derived from images are computed, EMD is used to measure the distance between pairwise histograms. All edit operations are transformed into edge operations; thus the GED of graphs is related to the connectivity difference of graphs that can be well