Neurocomputing 156 (2015) 211–220

Contents lists available at ScienceDirect

Neurocomputing

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

An error-tolerant approximate matching algorithm for labeled combinatorial maps

Tao Wang [a,b,*], Hua Yang [c], Congyan Lang [a,b], Songhe Feng [a,b]

[a] School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China
[b] Beijing Key Laboratory of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing 100044, China
[c] Institute of Information Engineering, Kaifeng University, Kaifeng 475000, China

ARTICLE INFO

Article history:
Received 26 March 2014
Received in revised form
15 November 2014
Accepted 21 December 2014
Communicated by X. Li
Available online 7 January 2015

Keywords:
Combinatorial map
Similarity measure
Pattern recognition
Graph

ABSTRACT

Combinatorial maps are widely used in image representation and processing, and measuring distance or similarity between combinatorial maps is therefore an important issue in this field. The existed distance measures between combinatorial maps based on the largest common submap and the edit distance have high computational complexity, and are hard to be applied in real applications. This paper addresses the problem of inexact matching between labeled combinatorial maps, and aims to find a rapid algorithm for measuring distance between maps. We first define joint-tree of combinatorial maps and prove that it can be used to decide of isomorphism between combinatorial maps. Subsequently, a distance measure based on joint-trees and an approximate approach are proposed to compute the distance between combinatorial maps. Experimental results show that the proposed approach performs better in practice than the previous approach based on approximate map edit distance.

1. Introduction

Image representation, segmentation and retrieval are import issues in the field of computer vision. Traditionally, in many image processing applications [1–3], Region Adjacency Graphs (RAGs) use vertices to describe the maximal homogeneous sets of connected pixels and edges to describe the adjacency relationships. Compared with traditional graph model, combinatorial maps are more powerful structures for modeling topological structures with subdivided objects. The concept was first introduced informally for modeling planar graphs [4,5], and was later extended to represent higher-dimensional subdivided objects [6,7]. Compared with traditional graph-based representations, combinatorial maps have some natural advantages. First, combinatorial maps are more precise for explicitly encoding the orientations of edges around vertices. Second, it is easy to descript high-dimensional patterns using combinatorial maps. In literatures combinatorial maps have been utilized in 2D and 3D image representation and processing [8–14]. And many practical problems in these fields can be formulated as the combinatorial map matching problem.

Matching combinatorial maps is therefore an important problem in the field of image analysis and processing. There have been some previous works on the map matching problem. The early research can be traced back to Cori who discussed the computation of the automorphism group of a topological graph embedding in his report [15]. Liu defined sequence descriptions for combinatorial maps [16], which are subsequently used for map isomorphism and map automorphism [17]. Gossenlin et al. proposed two map signatures which are used to efficiently search for a map in a database [18]. Damiand et al. proposed a polynomial algorithm for searching compact submap in planar maps [19], and then extended this work to n-dimensional open combinatorial maps [20]. Wang et al. proposed a quadratic algorithm for submap isomorphism based on sequence searching [21].

All these works described above are only for the exact map matching problem, which aim for finding an exact one-to-one mapping between two combinatorial maps. In real applications, two objects having small structural differences are usually considered as matched. Also, real world objects are usually affected by noises so that map representations extracted from identical objects at different time are rarely exactly equal. Therefore, it is necessary to integrate some degree of error-tolerance into the map matching process. Combier et al. defined the first error-tolerant distance measure for comparing generalized maps by means of the size of a largest common submap [22], and then related maximum common submaps with the map edit distance by introducing

* Corresponding author at: School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China.
E-mail addresses: twang@bjtu.edu.cn (T. Wang), yanghua@kfu.edu.cn (H. Yang), cylang@bjtu.edu.cn (C. Lang), shfeng@bjtu.edu.cn (S. Feng).

http://dx.doi.org/10.1016/j.neucom.2014.12.058
0925-2312/

special edit cost functions [23]. This approach cannot be directly used for comparing labels on the maps, while in most scenarios maps extracted from real world objects are always labeled. Wang et al. defined edit distance of combinatorial maps and proposed an optimal algorithm to compute map edit distance based on tree search [24]. This approach is more flexible in terms of the ability of comparing labels on maps. However, this measure also has exponential computational complexity in the worst case and is difficult to be applied directly in real applications. An alternative way is to find approximate optimization methods at the cost of acceptable sub-optimal solutions, e.g. Wang at al. also proposed an approximate algorithm to compute map edit distance based on Greedy strategy [24].

In this paper, we address the problem of measuring distance between two combinatorial maps, which is one of the most important and fundamental issues of the inexact map matching problem. In particular, we aim to efficiently solve the labeled map matching problem via relaxing the problem to tree matching. We first define *Joint-tree* of combinatorial maps and propose an efficient algorithm for construction of joint-trees. Furthermore, we prove that joint-trees can be used to decide of map isomorphism, and subsequently propose a distance measure between combinatorial maps by means of edit distance between joint-trees. By this way, the problem of combinatorial map matching is relaxed to the problem of ordered tree matching, which can be solved by a simple dynamic programming algorithm. As shown in Experiments (Section 5), compared with the previous approach based on approximate map edit distance, the proposed approach is not only more computationally efficient but also provides more accurate results.

2. Background

In this section, we first recall some basic notions of combinatorial maps, and then introduce briefly the map edit distance defined by Wang et al. [24]. Concepts and terminologies not mentioned here can be found in Refs. [6,7,24].

2.1. Recalls on combinatorial maps

A 2D combinatorial map may be understood as a graph explicitly encoding the orientation of edges around a given vertex. The basic element in combinatorial maps is called *dart*, and each edges is composed of two darts with different direction. The fact that two darts stem from the same edge is recorded in the involution α. A permutation σ defines the rotation of darts around a vertex. Each cycle of σ is associated to one vertex and encodes the orientation of darts encountered when turning counterclockwise around this vertex (e.g. the σ-cycle $(3, 4, -1)$ in Fig. 1).

Definition 1. (2D labeled combinatorial map). A 2D labeled combinatorial map G is a 4-tuple $G=(D, \alpha, \sigma, \mu)$ where

- D is a finite set of darts,
- α is the involution on D,
- σ is the permutation on D,
- and μ is a dart label function.

Fig. 1 demonstrates the derivation of a combinatorial map from a plane graph, where $D=\{1,-1, 2, -2, 3, -3, 4, -4, 5, -5, 6, -6\}$, $\alpha=(1, -1)(2, -2)(3, -3)(4, -4)(5, -5)(6, -6)(7, -7)$ and $\sigma=(1, 2)(3, 4, -1)(5, -4)(7, -2)(6, -3, -7)(-5, -6)$. Usually, μ is a partial function mapping darts to a finite set of integers, characters or vectors. A labeled map $G=(D, \alpha, \sigma, \mu)$ is connected if for any two darts x and y in D, y can be reached from x by successive applications of the involution α and the permutation σ. For the sake of simplicity, maps in this paper are connected and vertices are unlabeled unless otherwise stated.

Compared with graph isomorphism problem, checking isomorphism of maps needs to integrate additional constraints on preserving topological relationships between edges. By considering this, the map isomorphism problem becomes simple.

Definition 2. (map isomorphism). Given two labeled maps $G_1=(D_1, \alpha_1, \sigma_1, \mu_1)$ and $G_2=(D_2, \alpha_2, \sigma_2, \mu_2)$, if there is a one-to-one mapping $\psi: D_1 \to D_2$ such that for any $x \in D_1$, there are

$$\psi(\alpha_1(x)) = \alpha_2(\psi(x)), \quad \psi(\sigma_1(x)) = \sigma_2(\psi(x)), \quad \mu_1(x) = \mu_2(\psi(x))$$

then G_1 and G_2 are considered isomorphic.

Combinatorial maps explicitly encode the information of the orientation of darts around vertices. This information enables us to differentiate between configurations in the graph matching problem, and it is not encoded by region adjacency graphs. Moreover, combinatorial maps may be defined in any dimensions. Indeed, the 3D image representation and processing based on combinatorial maps is an active research field [12–14].

The following notations may be used in the rest of this paper. Given a map G, let $E(G)$ and $V(G)$ denote the edge set and the vertex set of G, respectively. Given a dart x, let $t(x)$ and $h(x)$ denote the tail vertex and head vertex of x, respectively, and let $e(x)$ denote the corresponding edge of x.

2.2. Map edit distance

Edit distance is one of the most flexible methods for error-tolerant matching of structures. It was initially introduced for string-to-string comparison [25], and was later extended to compare trees [26–28], graphs [29–31] and combinatorial maps [24].

A standard set of edit operations includes insertion, deletion and substitution of both darts and vertices. Denote the substitution of two darts x and y by $(x \to y)$, the deletion of dart x by $(x \to \Lambda)$, and the insertion of dart y by $(\Lambda \to y)$. Edit operations on vertices are implied by edit operations on their darts, i.e. whether a vertex is substituted, deleted, or inserted, depends on the edit operations of its darts. Given a source map G_1 and a target map G_2, a sequence of edit operations $S=s_1, \ldots, s_k$ that transforms G_1 completely into G_2 is called an edit path between G_1 and G_2. Let γ be a distance metric that assigns a nonnegative real number $\gamma(x \to y)$ to each edit operation $(x \to y)$, where x and y may be darts or vertices. Note that γ is defined on the whole set of operations and is not specific to two given maps. Extend γ to the edit path S by

$$\gamma(S) = \sum_{i=1}^{k} \gamma(s_i),$$

and to the combinatorial map $G=(D, \alpha, \sigma, \mu)$ by

$$\gamma(G) = \sum_{x \in D} \gamma(x \to \Lambda).$$

Definition 3. (map edit distance). Let G_1 be the source map and G_2 be the target map. The map edit distance between G_1 and G_2 is defined by

$$d(G_1, G_2) = \min\{\gamma(S) \mid S \text{ is an edit path between } G_1 \text{ and } G_2\}.$$

Given two maps $G_1=(D_1, \alpha_1, \sigma_1, \mu_1)$ and $G_2=(D_2, \alpha_2, \sigma_2, \mu_2)$, and a triple (M, G_1, G_2) where M is a set of pairs of darts (x, y) $(x \in D_1$ and $y \in D_2)$, we say that x and y are *touched by a line* in M if $(x, y) \in M$, and that x_1 *follows* x_2 $(x_1 \in D_1$ and $x_2 \in D_1)$ in M if $x_1=\sigma_1^k(x_2)$ $(k>0)$ and none of the darts $\sigma_1(x_2), \sigma_1^2(x_2), \ldots, \sigma_1^{k-1}(x_2)$ is touched by any line in M.