Pattern Recognition Letters 32 (2011) 1108–1118

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Fast palmprint identification with multiple templates per subject

Feng Yue [a,b], Wangmeng Zuo [a], David Zhang [a,c,*], Bin Li [b]

[a] Harbin Institute of Technology, School of Computer Science and Technology, No. 92 Xi Da-Zhi Street, Harbin 150001, China
[b] Beijing Institute of New Technology Applications, No. 28 Bei Yuan Street, Beijing 100012, China
[c] The Hong Kong Polytechnic University, Biometrics Research Center, Department of Computing, Hung Hom, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 2 November 2009
Available online 4 March 2011
Communicated by G. Borgefors

Keywords:
Palmprint identification
Tree structure
Competitive code
Ordinal code
Virtual template

ABSTRACT

Palmprint identification system commonly stores multiple templates for each subject to improve the identification accuracy. The system then recognizes a query palmprint image by searching for its nearest neighbor from all of the templates. When applied on moderate or large scale identification system, it is often necessary to speed up this process. In this paper, to speed up the identification process, we propose to utilize the intrinsic characteristics of the templates of each subject to build a tree, and then perform fast nearest neighbor searching with assistance of the tree structure. Furthermore, we propose a novel method to generate the 'virtual' template from all the real templates of each subject. The tree constructed by the virtual template and the real templates can further speed up the identification process. Two representative coding-based methods, competitive code and ordinal code, are adopted to demonstrate the effectiveness of our proposed strategies. Using the Hong Kong PolyU palmprint database (version 2) and a large scale palmprint database, our experimental results show that the proposed method searches for nearest neighbors faster than brute force searching, and the speedup becomes larger when there are more templates per subject in the database. Results also show that our method is very promising for embedded system based moderate scale and PC based large scale identification systems.

1. Introduction

As an emerging biometric technology, palmprint recognition has received considerable research interests because of its low-price capture device, fast execution speed, and high accuracy. Early researches on palmprint recognition mainly focus on the extraction of structural features like ending points of lines and all the points belonging to palmprint lines (Shu and Zhang, 1998). With the increasing interests on the development of online biometric technology, low-resolution palmprint recognition, which is more suitable for civil and commercial applications, has gradually been the focus of recent research interests (Zhang et al., 2003). A number of palmprint recognition systems have already been successfully applied for access control in community, campus, office building, etc.

Palmprint recognition may operate in either of two modes, verification or identification. In verification, an identity is validated by matching a test palmprint feature against a stored template of claimed identity and then deciding whether the claim is true or not. Verification is thus a typical one-to-one matching task. In identification the task is more complex, as determining the identity of a query palmprint requires the system to search for the most similar one from all the templates in the database. The subject which the resultant template belongs to is recognized as the identification result. Identification is thus a one-to-many matching task and is more challenging than verification. Obviously, a central difficulty in developing a moderate (several thousands registered subjects) or large scale (more than ten thousands registered subjects) palmprint identification system is that identification can be very time-consuming. For example, a typical palmprint identification system used in an office building or a campus for access control may have around one thousand registered subjects. If three templates per subject are stored in the database and the average matching time is 10 ms, then about 30 s are required for personal identification. Thus it is often necessary to speed up this process in identification systems.

Two strategies have been proposed for fast palmprint identification, hierarchical matching and palmprint classification. Hierarchical matching (Li et al., 2005; You et al., 2002; You et al., 2004) typically involves extracting multiple kinds of features and then searching in a layered fashion. Simpler features which can be quickly extracted and matched are used in higher layers because they allow a large number of candidates to be discarded. The drawback of this is that the templates discarded at higher layers may contain the nearest neighbor of the query. Classification strategies (Wu et al., 2004; Zhang and Zhang, 2004) proceed by dividing palmprints into several classes and the query palmprint is matched

* Corresponding author at: Harbin Institute of Technology, School of Computer Science and Technology, No. 92 Xi Da-Zhi Street, Harbin 150001, China.

E-mail addresses: csfyue@gmail.com (F. Yue), cswmzuo@gmail.com (W. Zuo), csdzhang@comp.polyu.edu.hk (D. Zhang), lbn@hit.edu.cn (B. Li).

0167-8655/$ - see front matter Crown Copyright © 2011 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.02.019

only with other templates in its class. The drawback here is that the initial classification may have put the query and its nearest neighbor into different classes, making a match impossible. Thus, while both strategies speed up the searching process, they do so at the expense of accuracy.

Recently Yue et al. (2009) proposed a fast palmprint identification strategy based on competitive code and the cover tree method (Beygelzimer et al., 2006). To meet the requirement of the cover tree method, this approach first demonstrated that the angular distance adopted in competitive code can be treated as a set of metrics and then exchanged the minimum value searching processes in palmprint matching and palmprint identification. Finally it constructed a set of cover trees that used all the templates in the database, and took advantage of them to facilitate fast nearest neighbor searching. This approach could get identification results identical with those obtained by brute force searching yet, according to their experiments, does so between 33% and 50% faster.

A common way to make palmprint identification systems more accurate and more robust against noise is to store multiple templates for each subject in the database. In some cases where only one template per subject is available, some strategies, such as rotating that template at different angles, are often adopted to enlarge the training set (Jia et al., 2008; Martinez, 2002). The templates of the same subject are usually very similar because they are either collected in the same session or generated from the same template. This characteristic can be utilized to facilitate fast palmprint identification.

In this paper, based on the work presented in (Yue et al., 2009), we propose an approach to speed up any coding-based palmprint recognition method by using the templates of each subject to build a tree and then using the tree structure to perform fast nearest neighbor searching. Furthermore, we propose a novel way to construct the virtual template from all the real templates of each subject, and build a tree using this virtual template and all the real templates, which can further speed up the identification process. We tested our approaches using two representative coding-based methods, competitive code and ordinal code, on both the Hong Kong PolyU palmprint database (version 2) and a large scale palmprint database. The results show that the proposed method searches for nearest neighbors faster than brute force searching while giving identical results. The speedup becomes even higher as more templates per subject are added to the database.

The remainder of this paper is organized as follows. Section 2 briefly describes two coding-based palmprint recognition methods. Section 3 considers the pros and cons of five search strategies that could be used in palmprint identification system, including the two strategies proposed here, using tree structure and virtual template. We also give the algorithms for constructing and searching in the trees. Section 4 presents the method of handling translation in palmprint matching. Section 5 gives our experiments and the results, and Section 6 offers our conclusion.

2. Coding-based palmprint recognition methods

Coding-based methods are among the most promising palmprint recognition methods because of their small feature size, fast matching speed and high accuracy (Kong et al., 2009). These approaches typically extract features by using a number of filters to convolve the palmprint image and then applying rules to encode the filtering result. Representative methods include competitive code (Kong and Zhang, 2004), ordinal code (Sun et al., 2005), Robust Line Orientation Code (RLOC) (Jia et al., 2008) and Binary Orientation Co-occurrence Vector (BOCV) (Guo et al., 2009). Although the proposed fast identification strategies can be applied with any coding-based method, in this paper, we demonstrate it only on competitive code and ordinal code, which will be briefly described in the following. For more details, please refer to Kong and Zhang (2004) and Sun et al. (2005).

2.1. Competitive code

The competitive code method first convolves the palmprint image using six real Gabor filters of different orientations and then encodes the dominant orientation as features. The distance between two palmprints is measured by the sum of the angular distance of all sampling points. Coding the dominant orientation into three bits allows us to compute the angular distance using a bitwise XOR operator. This distance is also known as the Hamming distance and is naturally a metric.

2.2. Ordinal code

Different from the competitive code method, ordinal code first uses six 2D elliptical Gaussian filters to filter the palmprint image, and then compares each pair of filtering responses orthogonal in orientations to generate one bit feature code. Each sample point thus generates one 3-bit code. The distance between two ordinal codes are also measured using the Hamming distance and thus is a metric.

3. Search strategies for palmprint identification

In this section, we consider the advantages and disadvantages of five search strategies that could be used in palmprint identification. The first strategy is the simplest, brute force searching; the second strategy, reference subset selection or condensing, involves first selecting only some representative templates from all the templates to build a condensed database, and then searching in this database; the third is the method proposed in (Yue et al., 2009), which utilizes the cover tree method (Beygelzimer et al., 2006) to facilitate fast nearest neighbor searching; the fourth uses the templates of each subject to build a tree, and then searches the tree structure; the fifth constructs a virtual template from all the templates of each subject and then builds a tree which, again, is used for searching.

3.1. Brute force searching

This strategy simply considers the templates in the database one by one until all the templates have been compared against the query. Since this strategy is the most time-consuming, it is often used as a benchmark.

3.2. Reference subset selection (condensing)

Reference subset selection or, as it is also known, condensing (Sanchez et al., 1997; Zhang and Sun, 2002) seeks to increase speed by reducing the size of the database. It does this by selecting only a few representative templates and using these to build a condensed database. The central task is to select a small subset that can nonetheless approximate the performance achieved when using the entire database. Although this has been shown to be a useful and promising strategy, a condensed database does not always obtain the same accuracy as using a complete database. Furthermore, its greatest efficiency is achieved on databases having a few classes and many samples per class. Our palmprint identification system, however, has many classes (subjects) and only a few samples (templates) per class. Given this, few templates would be discarded, and the speedup would be correspondingly limited. Consequently, this