<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Determining Unique Fingerprint Features for Biometric Encoding of Data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Cucu, Catalin; Cucos, Alex; Corcoran, Peter</td>
</tr>
<tr>
<td><strong>Publication Date</strong></td>
<td>2008</td>
</tr>
<tr>
<td><strong>Item record</strong></td>
<td><a href="http://hdl.handle.net/10379/3669">http://hdl.handle.net/10379/3669</a></td>
</tr>
</tbody>
</table>

Some rights reserved. For more information, please see the item record link above.
Determining Unique Fingerprint Features for Biometric Encoding of Data

Catalin Cucu, Alex Cucos and Peter Corcoran, Member, IEEE

Abstract — A novel approach to determining unique fingerprint features with high repeatability is described. The techniques described are particularly suitable for CE applications and devices employing inexpensive capacitive fingerprint sensors. The extraction of robust seeds for generating repeatable encryption keys is discussed and some preliminary approaches are described.

Keywords — Biometrics, Multimedia, Digital Media Encoding, Home Networks.

I. INTRODUCTION

Typical applications for biometric authentication require very accurate sensing technique with high repeatability [1] or a complex analysis technique [2] which can compensate for variations in the sensing process. Either of these approaches can yield a high level of granularity for the comparison step but with the trade-off of requiring an expensive sensing subsystem or sufficient computing power and memory to perform complex processing. Typical applications will also require secure access to a large database for comparing acquired and processed biometric signatures.

Our interests lie in applying authentication techniques to CE devices and services and in the use of biometrics to manage digital content and as a means for actuating the recording or duplication process for digital content [3]. These CE applications have different requirements: the strict level of authentication which is necessary for specialized business applications is simply not necessary; faster response times and greater certainty of correct user authentication are desirable. Using such modified criteria could allow biometric authentication to be built directly into the recording switch on a digital media recorder.

A useful summary of biometric techniques for fingerprint analysis and matching can be found in [4, 5].

II. FINGERPRINT MATCHING TECHNIQUES

In this paper we focus on a novel approach to extracting robust, rotation invariant features from a typical set of fingerprint minutiae. To place our work in context it is useful to quickly summarize some of the prior art in fingerprint analysis.

A. Affine Pattern Matching Techniques

Quite a few different algorithms have evolved, particularly from the field of image registration to determine an optimal match between two 2D point patterns. Amongst the most recent approaches we mention the iterative closest point algorithm (ICP) described in [8]. The basic ICP algorithm has been extended by a number of authors. For example, in [9] it is extended to handle correspondence between a point and a tangent plane to overcome the lack of an exact correspondence between the two point sets; in [11, 13] the algorithm is made more robust to the influence of outliers and features lacking correspondence; in [10] a weighted least mean error estimate is employed and in [12] a metric is employed which trades off distance against feature similarity based on local shape invariance. For real-time applications [14] examines performance improvements to the basic algorithm.

Another recent approach to such problems is that of thin plate spline (TPS) matching algorithms [15, 16] which are robust to false positive-negative candidate spots and unknown non-affine warps between the candidate point sets. This algorithm works by iteratively performing a joint estimation of the correspondence between two sets of points which are to be registered, and the TPS mapping between the images that the points lie in.

Fig 1: User Interface to our Fingerprint Analysis Tools

Although we have tested some of the above algorithms we reached a conclusion that such algorithms are flawed for our purpose as they assume a correspondence between the two point sets to be matched. Thus they are not optimized to indicate quickly when point sets are unlikely to allow a good match.

B. Ridge Field Pattern Matching Approaches

An alternative approach to fingerprint pattern matching is provided by analyzing the ridge-field pattern of an acquired fingerprint. This provides useful global information which is additional to the minutiae points and which is relatively robust to the quality of image acquisition [22]. When combined with local, minutiae based features the use of this orientation field data can provide both accurate and robust matching [21]. However
the algorithms proposed by these authors are too slow for consumer applications which require real-time responses (i.e. < 5 seconds) and are better suited to forensic matching applications rather than for use in embedded CE systems which require realistic response times.

**C. Minutiae Triangle Descriptors**

A number of authors have turned their attention to an alternative approach to the matching of fingerprint patterns [17-20] based on the use of local triangular features formed from nearest-neighbor minutiae groupings. In [17] the author builds a connected graph composed of triangles formed from nearest-neighbor minutiae with shared edges. By allowing for small localized variations in the lengths of the shared edges this matching technique can account for global variations of us to 45% across the entire fingerprint. The authors of [18] use localized matching of minutiae pairs combined with a global matching. They use the best matched local structure pair to determine a global translation and rotation for the remaining minutiae and perform their overall decision based on the resulting match of the global minutiae set. In [19] the authors employ fuzzy similarity measures for determining matches between individual local triangle features and combine these to obtain a global measure of similarity between two fingerprints. Finally, the authors of [20] build a set of triangle descriptor feature vectors for each fingerprint and then proceed to determine if the triangular frameworks for each fingerprint correspond closely enough to declare a match. Their technique is differentiated from that of [18] as it is not necessary to perform a global alignment of the minutiae patterns. Furthermore, where additional local ridge-field data is used the discriminating power of these local triangular features is somewhat improved and these features are essentially rotation-invariant. An example of this modified triangular descriptor is given in Fig 2 below.

![Fig 2: Local Triangle Descriptor formed by 3 Minutiae Points](image_url)

One weakness in the use of such triangular descriptors, and indeed for all of the above techniques is that they are susceptible to erroneous acquisition data. If a single minutiae point is omitted, or is incorrectly detected as two adjacent minutiae then the pattern matching algorithm will often fail. This is not a problem for most authentication scenarios as the user can simply re-apply their fingerprint, paying more attention to the pressure and duration of the acquisition process. However in a consumer application it is desirable that even if the acquisition process is poorly executed there is still a high probability of a successful outcome.

**III. USB Fingerprint Sensor**

For our initial system development the DKF200 software development kit from Fujitsu Inc was used with the MBF200 fingerprint sensor to implement the fingerprint scanning subsystem. The DKF200 kit includes supporting software libraries, however these did not provide access to raw image data of acquired fingerprints so we developed our own device driver and generic image processing algorithms to enable direct access and post-processing of fingerprint images [6, 7].

One disadvantage of a low resolution capacitative sensor such as the MBF200 is that the determination of a repeatable minutiae set is problematic. This can be readily understood if we realize that fingerprint ridges and ridge junctions are of finite thickness and in some scans certain ridges and junctions may be too thin, or be split across adjacent pixels and may not be correctly registered by such a low resolution sensor. As stated above, the high probability that at least some minutiae will missed, or other erroneous minutiae will be registered on subsequent fingerprint scans presents a significant barrier to the use of low-cost fingerprint sensors in CE applications.

**IV. Determination of Unique and Robust Features**

Bearing the above considerations in mind we have developed a novel approach to extracting repeatable fingerprint features from the unreliable minutiae feature sets which occur when low-resolution fingerprint sensors are employed for biometric authentication or encoding applications. Our approach draws inspiration partly from the local triangular features described by the authors of [20]. To overcome the problem of unreliable minutiae we only require that a repeatable feature consists of two corresponding minutiae, and each minutiae point may be analyzed in terms of the set of features it shares with its neighboring minutiae points. This is illustrated in Fig 3 below.

**A. Enrollment Process**

To properly determine a set of such repeatable two-point features we employ a multi-stage enrollment where the user must provide a set of, typically, 4-6 fingerprint scans. After each scan the set of determined features is classified into a set of angle and distance bins. Minutiae are matched across the set of scans – this is easily achieved by a comparison of the extracted feature sets. Then the
most repeatable features from each minutiae are recorded across the set of enrollment scans.

<table>
<thead>
<tr>
<th>Neighbor</th>
<th>Distance</th>
<th>Neighbor Angle</th>
<th>Self Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>284.1°</td>
<td>50.1°</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>209.7°</td>
<td>143.2°</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>296.9°</td>
<td>242.6°</td>
<td></td>
</tr>
</tbody>
</table>

When applications are more security conscious it may be required to match a larger number of features.

**REFERENCES**


