Selective Bitplane Encryption of Fingerprint Images

Elias Pschernig
Contents

• Introduction
  – selective bitplane encryption, replacement attack

• Paper
  – bitplane encryption, method used in paper

• Matching of encrypted fingerprints
  – Preprocessing
  – FVS
  – Java-Applet
  – Verifinger
General overview

- Sensor reads image, client does feature extraction (minutiae)
- Data transmission:

  Sensor → Client → Matching → DB

  Feature Extraction

  Result
Selective Bitplane Encryption

• Full encryption: too much computation
• Possibility: Encrypting spatial subset
• but “mosaic technique”
• -> confidentiality lost
• Instead: Only one plane is encrypted
• usually 8 bitplanes in FP picture
• but: Replacement attack
Replacement attack

encrypted MSB

encrypted MSB and next plane
Selective Bitplane Encryption

- Paper: New method using LSB and XOR
- Distortion: XOR of LSB with other planes (sort of one time pad, but one plane to pad 7)
- Encryption: only LSB is encrypted
Selective Bitplane Encryption

- XOR with LSB, simply repeated 8 times and shifted to the right as column stripes
Selective Bitplane Encryption

- Other example, with large border
Selective Bitplane Encryption

- Other simple distortions with LSB

Per-pixel  Stretch  Accumulate  Run
Selective Bitplane Encryption

- Possibility: First encrypt LSB
- but: 8 x XOR with first stripe
Method from Paper
Matching of encrypted images

- Preprocessing
- FVS
- Java Applet
- Verifinger
Preprocessing

• Instead of selective encryption, assume simple replacement attack -> plane is simply missing
• preprocessing: histogram normalization
• result fed directly to fingerprint application
• 256 bitplane combinations
Preprocessing

• Histogram normalization
Possible bitplane combinations

- 256 combinations, single plane missing or single plane present marked
Normalized Bitplanes

MSB  plane 6  plane 5  plane 4
plane 3  plane 2  plane 1  LSB
FVS

- by Shivang Patel, 2002
- [http://fvs.sourceforge.net](http://fvs.sourceforge.net), CVS access
- Open Source, written in C++
- Minutiae based
- somewhat unfinished
- matching doesn't seem to work
FVS minutiae

- no filtering of minutiae
FVS – feature extraction steps
FVS – bit planes

- MSB missing: 0%
- only MSB: 13%
Java Applet

- by Peter Söllner and Christian Birzer, 1998
  http://fbim.fh-regensburg.de/~saj39122/meile/demos/bisoe/index.html

- Uses Neural Net

- First step: Minutiae extraction

- Minutiae of multiple fingerprints of different persons are learned with a Kohonen Feature Map

- Unknown minutiae set fed to the net is classified to a specific person
Java Applet

### Eingabebilder:

<table>
<thead>
<tr>
<th>Christian_0</th>
<th>Alles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Christian_1</td>
<td>Nichts</td>
</tr>
<tr>
<td>Christian_2</td>
<td></td>
</tr>
<tr>
<td>Christian_3</td>
<td>Invertieren</td>
</tr>
<tr>
<td>Christian_4</td>
<td></td>
</tr>
<tr>
<td>Christian_5</td>
<td></td>
</tr>
<tr>
<td>Christian_6</td>
<td></td>
</tr>
</tbody>
</table>

### Kantendetektionsverfahren:

- Schwellwert
- Sobel vertikal
- Logarithmisch

### Bildbearbeitung starten

**Bereit!**

### Vektoren:

<table>
<thead>
<tr>
<th>Christian_0W</th>
<th>Alles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Nichts</td>
</tr>
<tr>
<td></td>
<td>Invertieren</td>
</tr>
<tr>
<td></td>
<td>Nächste 10</td>
</tr>
</tbody>
</table>

Alle löschen
Java Applet

- Visualization of Kohonen feature map
- Neural Net is trained with 4 persons, 10 fingerprints of each
- Searching: Returns best match
Java Applet

- Supports identification only: Result seems to always be the closest matching person, never no match

- Bitplanes: Encrypted all 10 prints of each person, and recorded the percentage of correctly recognized prints as result
Java Applet

Java

- L
- P
- A
- C

0
25
50
75
100
125
150
175
200
225
250
275
300
325
350
375
400

bars representing different categories and values.
Verifinger

- demo by Neurotechnologija
- Minutiae based
- no batch processing
- Bitplane encryption:
  - Enroll fingerprint
  - Try to match with missing bitplanes -> matching score
Verifinger

- enrolled same print as with FVS

- matched against other print with score of 642
Verifinger

- MSB missing: 369
- only MSB: 1011
Verifinger

Matching score for encrypted bitplanes

- 1_1.bmp
- 4_1.bmp
- 2_1.bmp

Matching scores for different bitplanes.
Verifinger - preprocessing

- only plane 6: score 388
- blur: score 399
- crop: score 451
Verifinger - preprocessing

- only plane 5: score 133
- cropped: score 187
- + blurred: score 236
Results

• Fingerprints with any plane except MSB missing usually match

• MSB alone matches, depending on images also other planes

• Same with combinations

• All planes contain ridge information – but with no/simple preprocessing no good matches for most
Possible further work

- More advanced preprocessing
- E.g. try to predict missing bitplanes using other bitplanes
- Or use techniques similar to the ones used in matching itself, like better determine regions with ridge information, extract bad-quality ridges (e.g. with Hough transformation), then reconstruct fingerprint out of the information before feeding to matching application