Irreversible Fingerprint Template

using Minutiae Relation Code with Bloom Filter

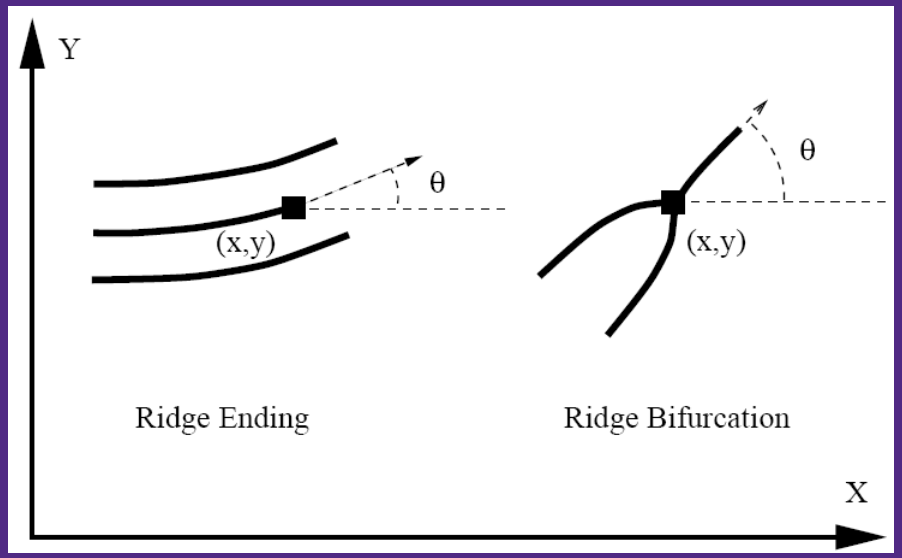
Presenters:

Haikun Liu	hlg483
Ke Wang	kwp862
Weishen Chu	wci004

Northwestern

What is minutiae

- Minutiae:
  - Local ridge characteristics
  - Ridge ending and ridge bifurcation



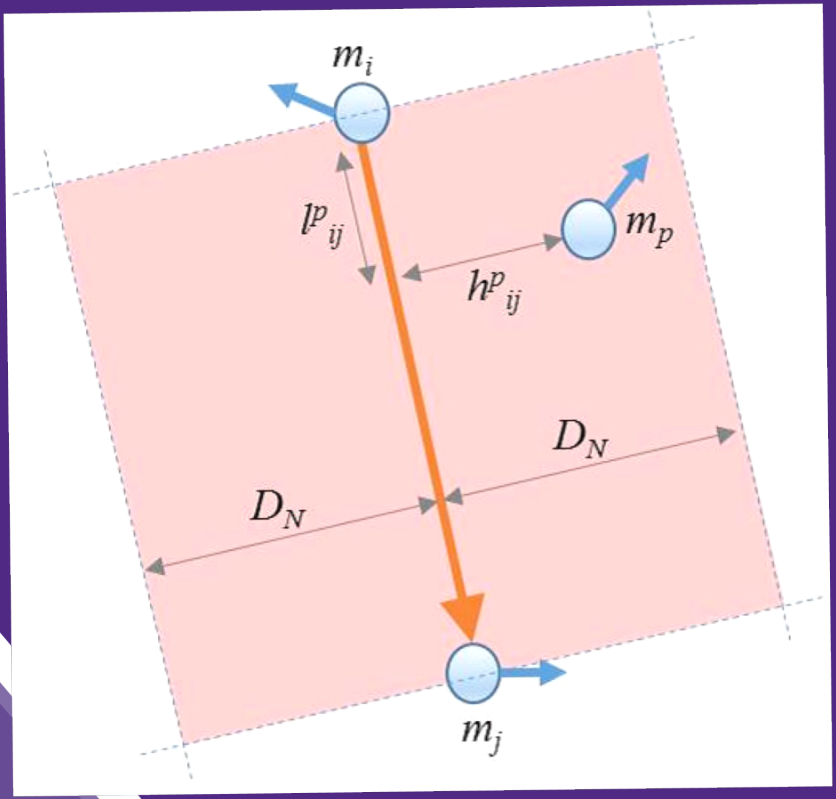
Irreversible template creation technique

- Minutiae Relation Code(MRC)
  - Describe the minutiae information efficiently
- Bloom Filter
  - realize the irreversibility feature

Irreversible template creation technique

- MRC consists of a set of vector-represented relation information between arbitrary minutiae, which enables to create a useful fingerprint template by handling boarder minutiae and isolated minutiae efficiently

How to construct MRC



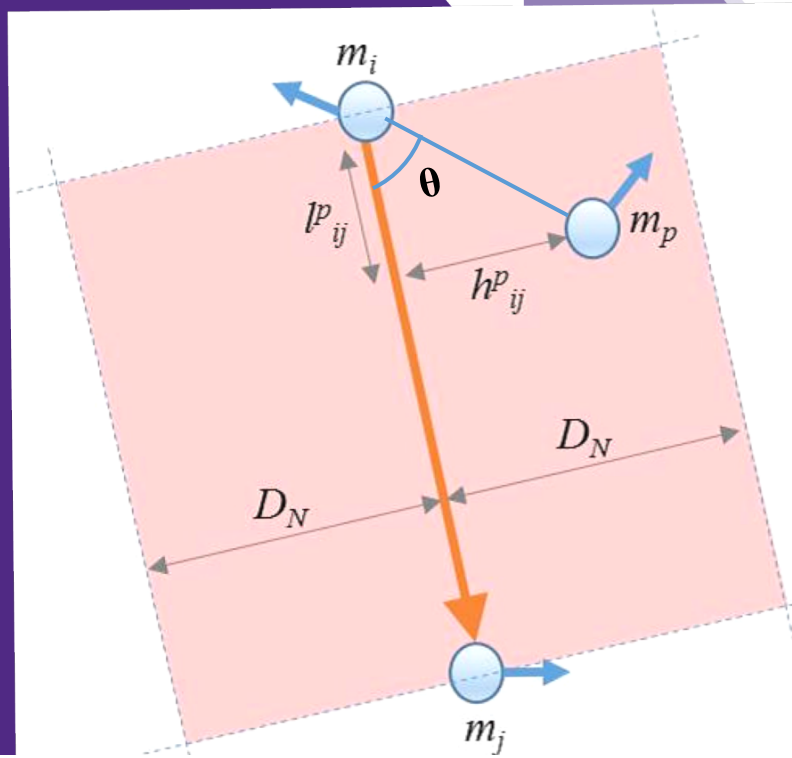
$$m_i = (x_i, y_i, \theta_i)$$

$$h_{ij}^p = \frac{|\boldsymbol{v}_{ip} \times \boldsymbol{v}_{ij}|}{L_{ij}}$$

$$l_{ij}^p = \sqrt{L_{ij}^2 - h_{ij}^{p\,2}}$$

How to construct MRC

- h_{ij}^{p} and l_{ij}^{p}
- N_S – the number of divided space information
- N_D – the number of divided direction information
- S_{nm}^{l}— a set of neighbor minutiae on the left
- S_{nm}^{r} — a set of neighbor minutiae on the right
- MRC represents an $N_S \times N_D$ matrix



$$lMRC_{ij}(s,t) = \sum_{p \in S_{nm}^{l}} f(s, l_{ij}^{p}, \sigma_S) \cdot f(t, h_{ij}^{p}, \sigma_D)$$

$$rMRC_{ij}(s,t) = \sum_{p \in S_{nm}^{r}} f(s, l_{ij}^{p}, \sigma_S) \cdot f(t, h_{ij}^{p}, \sigma_D)$$

$$f(x,c,\sigma) = exp\left(-\frac{(x-c)^2}{2\sigma^2}\right)$$

$$0 \le s < N_S, 0 \le t < N_D$$

How to construct MRC

- C_{energy} – summation value of MRC_{ij}
- N_{norm} – normalization factor
- th – bit-quantization threshold

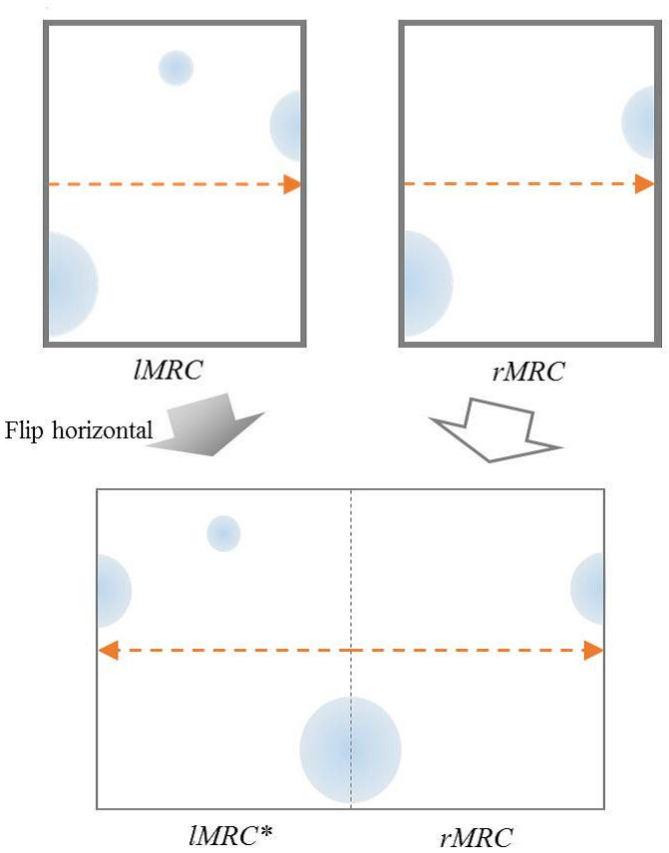
$$bMRC_{ij} = \begin{cases} 1 & MRC_{ij} \cdot N_{norm}/c_{energy} > th \\ 0 & MRC_{ij} \cdot N_{norm}/c_{energy} \leq th \end{cases}$$

- In order to avoid ambiguous relations
- Introduce the limitation on the distance between a minutiae pair

$$A = \{(i,j) | L_{min} < L_{ij} < L_{max}\}$$

- A template T of the image I can be described by the following:

$$T = \{i, j, L_{ij}, MRC_{ij}\}_{i,j \in A}$$



$$MRC_{ij} = [lMRC^{*}_{ij} | rMRC_{ij}].$$

How to calculate similarity between MRCs

- $F_{fd}(MRC_{ij}, MRC_{uv}) = \sum_{s=1}^{N_s} \sum_{D=1}^{N_D} |MRC_{ij}(s,t) - MRC_{uv}(s,t)|$
- *vote_score_table(vst)*
  - Tabulate voting counts
  - $N_S \times N_D$
- *feature_score_table(fst)*
  - Store the minimum MRC distance
  - $N_S \times N_D$

Algorithm 1 Calculate a *vote_score_table* and a *feature_score_table*

$vst \Leftarrow 0$
$fst \Leftarrow max\_value$
for all i, j, u, v such $abs(L_{ij}^1 - L_{uv}^2) < D_R$ **do**
  $[i, j, u, v] \Leftarrow [i, j, u, v | min(F_{fd}(MRC_{ij}^1, MRC_{uv}^2)]$
  $min\_score \Leftarrow F_{fd}(MRC_{ij}^1, MRC_{uv}^2)$
  $vst(i, u) \Leftarrow vst(i, u) + 1$
  $vst(j, v) \Leftarrow vst(j, v) + 1$
  $fst(i, u) \Leftarrow min(fst(i, u), min\_score)$
  $fst(j, v) \Leftarrow min(fst(j, v), min\_score)$
end for

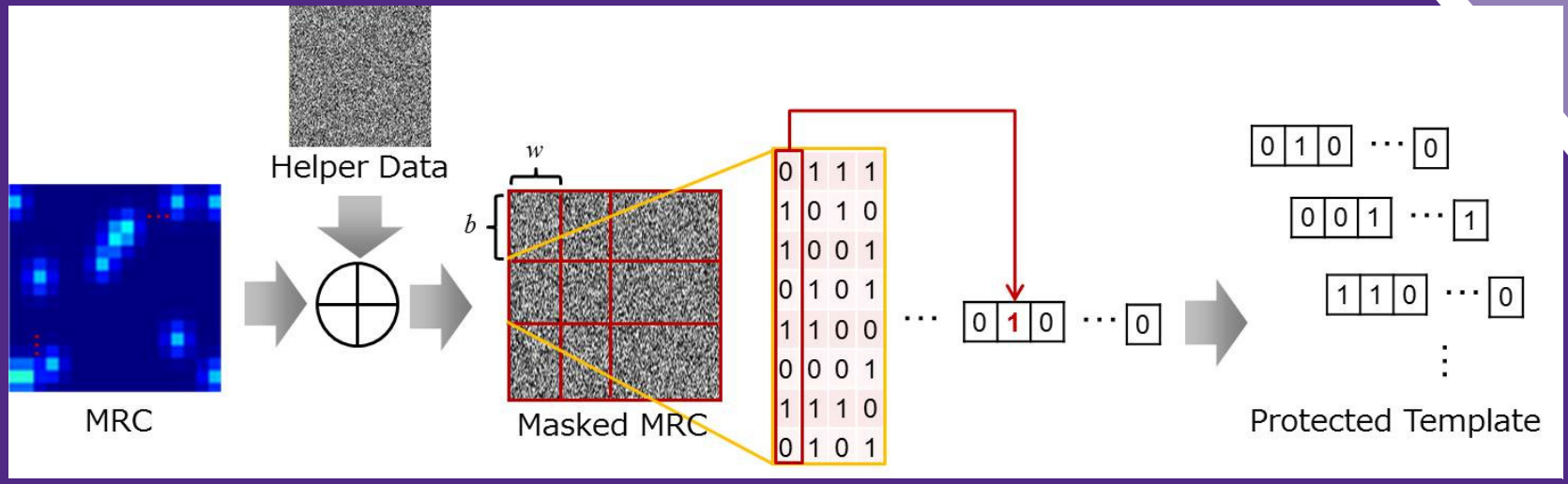
How to calculate similarity between MRCs

- vst_k – sort vst with respect to vote scores
- fst_k – consist of feature scores related to minutiae pairs with the smallest distance
- $$vs = \frac{\sum_{k=1}^{K} vst_k}{n_v K} \qquad fs = \frac{\sum_{k=1}^{K} fst_k}{n_f K}$$
- Where n_v and n_f represent normalization factors
- K – security factor
  - K becomes greater, the number of minutiae pairs between a template and an input data must be large for an appropriate authentication
  - value K is too small, the number of minutiae pairs is not enough for an authentication which may cause a false acceptance
- $$raw\_score = \alpha \cdot vs + (1 - \alpha) \cdot fs$$
- Where α is a weight to adjust the combination of vs and fs, raw score is lower if the two templates are similar with each other

Bloom Filter

- Contributes to the irreversibility of proposed scheme
- Has dimension reduction scheme to project an item to a probabilistic data structure
- Be used as a detector of the existence of the item in the data structure

Bloom Filter



1. Create a $N_D \times 2N_s$ random matrix as helper data
2. Calculate XOR between the helper data and the original MRC as Masked MRC
3. Divide into local blocks whose size is $b \times w$
4. Each column bits(b bits) are converted to a decimal number(0 to 2^b), and set 1 to the corresponding bit in the projection structure
5. The conversion is conducted w times in each block
6. Generate $\frac{N_D}{b} \times \frac{2N_s}{w}$ converted codes
7. Store the converted codes as a protected template

Evaluations and future works

- Evaluate the authentication and security performance of MRC with Bloom Filter
  - Use public fingerprint databases: FVC2002 and FVC2004
  - The proposed method can achieve 1.8% EER in FVC2002 DB2 with 2^{49} attack possibilities
- Future works
  - Keep looking into improvements regarding the authentication accuracy of MRC, and evaluate the other aspects like the unlinkability as a template protection scheme

Thank you

Q and A

Northwestern