

Advanced Variants of Feature Level Fusion for Finger Vein Recognition

Christof Kauba,¹ Emanuela Piciuccio,² Emanuele Maiorana,² Patrizio Campisi² and Andreas Uhl¹

Abstract: Authentication based on vein patterns is a very promising biometric technique. The most important step is the accurate extraction of the vein pattern from sometimes low quality input images. A single feature extraction technique may fail to correctly extract the vein pattern, entailing bad recognition performance. One of the solutions that can be used to improve recognition results is biometric fusion. A possible fusion strategy is feature level fusion, that is the fusion of several feature extractors' outputs. In our work, we exploited the feature level fusion to improve the quality of the extracted vein patterns and thus the feature extraction accuracy. An experimental study involving different feature extraction techniques (maximum curvature, repeated line tracking, wide line detector, ...) and different fusion techniques (majority voting, weighted average, STAPLE, ...) is conducted on the UTFVP finger-vein data set. The results show that feature level fusion is able to improve the recognition accuracy in terms of the EER over the single feature extraction techniques.

Keywords: Finger Vein Recognition, Feature Level Fusion, Binary Vein Images, STAPLE

1 Introduction

Biometric recognition devices are widely used nowadays for different authentication and identification purposes. Vein based recognition systems are one of the more recent biometric techniques, relying on the pattern of the blood vessels inside the human body. Finger vein recognition utilizes the vein pattern inside the fingers of a human and has several advantages compared to the recognition based on traditional biometric traits, such as fingerprint, face or iris. The veins are underneath the skin and only visible in infrared light, thus the vein pattern is more resistant to forgery. The vein pattern is neither sensitive to finger surface conditions nor to abrasion. In addition, liveness detection is easily possible. The major disadvantage of finger vein recognition are the low contrast and low quality vein images. Therefore, suitable preprocessing and feature extraction techniques are vital to achieve good recognition results.

A single feature extraction technique may fail in correctly extracting the vein pattern which leads to bad matching results and thus bad recognition performance. One approach to improve the performance is biometric fusion. Several fusion techniques have been investigated in literature. Fusion of several fingers is one possibility [YYW12, Qi13]. Score level

¹ Department of Computer Sciences, University of Salzburg, 5020, Salzburg, Austria, {ckauba, uhl}@cosy.sbg.ac.at

² Section of Applied Electronics, Department of Engineering, Rome Tre University, 00146 Roma, Italy, {emanuela.piciuccio@stud.uniroma3.it, emanuele.maiorana@uniroma3.it, patrizio.campisi@uniroma3.it}

fusion is another option where multiple features extracted from a single image are then fused after score calculation [ZK10] which showed promising results. Liu et al. [Li13] fused the scores of pixel and super-pixel level features using weighted sum and were able to improve the recognition rates. In [KRU14] the authors tested several pre-processing methods in combination with score level fusion of different feature extraction schemes, improving the matching performance. Yang et al. [YZ12] combined features extracted from fingervein and fingerprint images utilizing a Gabor filter approach into one single feature vector. Yang and Zhang [YZ10] did a feature level fusion, where the extracted vein pattern images are fused before score calculation using global and local vein features combining them using CCA and a weighted fusion scheme. In contrast to their work we investigate advanced feature level fusion techniques for global features using a set of different feature extractors all outputting binary vein images and several different fusion strategies. These are evaluated and compared against the performance results of the single feature extraction techniques in terms of EER (equal error rate). We also perform a runtime analysis in order to find the best tradeoff between improvement of the recognition accuracy vs. increased complexity.

The rest of this paper is organised as follows: Section 2 describes finger vein recognition, including preprocessing, feature extraction and matching techniques. In section 3 a brief introduction to biometric fusion and the used feature level fusion techniques is given. Section 4 explains the setup of our experimental study and provides the results with respect to the different feature extraction and fusion schemes. Section 5 concludes this paper and gives an outlook on future research.

2 Finger Vein Recognition

In this section the finger-vein preprocessing, feature extraction and matching approaches evaluated during our experiments are described.

2.1 Preprocessing

The first preprocessing step is to mask out the background region, setting background pixels to 0. We selected the region of the finger according to the method proposed by Lee et al. [LLP09]. A normalization step, i.e. a rotation compensation, is later performed, following the approach proposed in [Hu10]. Afterwards the image contrast is improved by using CLAHE [Zu94], an adaptive histogram equalization technique.

2.2 Feature Extraction

To be able to fuse the different extracted features at feature level all features must be of the same type. Thus we evaluate only feature extraction techniques producing a **binary vein output image** by trying to separate the vein pattern from the background.

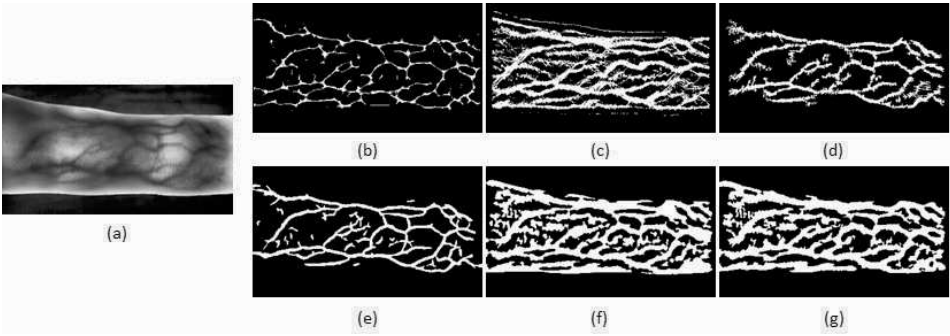


Fig. 1: (a) Original vein image. Features extracted using the (b) MC, (c) RLT, (d) WLD, (e) PC, (f) GF and (g) IUWT methods.

Maximum Curvature (MC [MNM07]) aims to emphasise only the centre lines of the veins and is insensitive to varying vein width. The first step is the extraction of the centre positions of the veins. For this purpose the local maximum curvature in the cross-sectional profiles, based on the first and second derivatives, is determined. Afterwards each profile is classified as being concave or convex where only local maxima in concave profiles indicate valid centre positions of the veins. Then a score according to the width and curvature of the vein region is assigned to each centre position, which is recorded in a matrix called locus space. Due to noise or other distortions some pixels may not have been classified correctly at the first step, thus the centre positions of the veins are connected using a filtering operation. Finally thresholding using the median of the locus space is applied.

Repeated Line Tracking (RLT [MNM04]) tries to track the veins as dark lines inside the image. Veins appear as valleys in the cross-sectional profile of the image. The tracking point is repeatedly initialised at random positions and then moved pixel by pixel along the dark line, where the depth of the valley indicates the movement direction. If no "valley" is detected, a new tracking operation is started. The number of times a pixel is tracked is recorded in a matrix. Pixels that are tracked multiple times as belonging to a line statistically have a high likelihood of belonging to a blood vessel. Thus, binarisation using thresholding is applied to this matrix to get the binary output image.

Wide Line Detector (WLD [Hu10]) is essentially an adaptive thresholding technique using isotropic non-linear filtering, i.e. thresholding inside a local neighbourhood region. The difference of the centre pixel to its neighbours inside a circular neighbourhood and the number of pixels inside this neighbourhood with a difference smaller than a predefined threshold are determined. This number is again thresholded to get the final vein image.

Principal Curvature (PC [Ch09]): At first the gradient field of the image is calculated. Hard thresholding is done to filter out small noise components and then the gradient at each pixel is normalised to 1 to get a normalised gradient field. This is smoothed by applying a Gaussian filter. The next step is the actual principal curvature calculation. It is obtained from the Eigenvalues of the Hessian matrix at each pixel. The two Eigenvectors of the Hessian matrix represent the directions of the maximum and minimum curvature and

the corresponding Eigenvalues are the principal curvatures. Only the bigger one which corresponds to the maximum curvature is used. The last step is again a binarization of the principal curvature values using Otsu's [Ot75] method to get the binary vein output image.

Gabor Filter (GF [KZ12]): A filter bank consisting of several 2D even symmetric Gabor filters with different orientations in $\frac{\pi}{\Omega}$ steps, where Ω is the number of orientations, is created. Several feature images are extracted by filtering the vein image using the different filter kernels of the Gabor filter bank. The final feature image is obtained by fusing all the single images from the previous step. This final vein output image is then post-processed using morphological operations to remove noise.

Isotropic Undecimated Wavelet Transform (IUWT [SFM07]): The IUWT is a redundant wavelet transform that is easy to implement. The scaling coefficients are computed by low-pass filtering and the wavelet coefficients by subtraction. Scaling coefficients preserve the mean while wavelet coefficients encode information corresponding to different spatial scales. Levels 2 and 3 of the transform are exhibiting the best contrast for the blood vessels, thus the feature image is obtained by adding wavelet levels 2 and 3. The final binary vein output image is obtained by thresholding the feature image followed by applying morphological post-processing operations to remove small noise in the image.

The several different kinds of feature extraction methods considered have one thing in common: they all output a binary vein image. An example of the output binary images of the different algorithms considered is shown in Fig. 1. For MC, PC, RLT and WLD we utilized the MATLAB implementation of B.T. Ton³. For IUWT we used the implementation from ARIA⁴. For GF we used a custom implementation similar to the one used in [Pi16].⁵

2.3 Matching

The approach of Miura et al. [MNM04] is adopted for matching the binary vein images. As the input images are not registered to each other and only coarsely aligned (rotation is compensated), the correlation between the input image and in x- and y-direction shifted versions of the reference image is calculated. The maximum of these correlation values is normalised and then used as final matching score. Due to the calculation of the correlation the matching time is proportional to the number of 1 values in the images. A MATLAB implementation of the matching method is provided by B.T. Ton⁶.

3 Feature Level Fusion

Biometric fusion [RNJ06] is a means of improving biometric system performance by the use of multiple biometric inputs or methods. Biometric fusion is able to improve appli-

³ Publicly available on MATLAB Central: <http://www.mathworks.nl/matlabcentral/fileexchange/authors/57311>

⁴ MATLAB Code available: <https://sourceforge.net/projects/aria-vessels/>

⁵ Can be downloaded from our website: <http://www.wavelab.at/sources/Kauba16e/>

⁶ Publicly available on MATLAB Central: <http://www.mathworks.com/matlabcentral/fileexchange/35716-miura-et-al-vein-extraction-methods>

cability, robustness and system accuracy. According to the different stages of a biometric recognition system, several types of fusion can be distinguished: sensor level, feature level, score level and decision level fusion. The latter in the biometric processing chain fusion is applied the more complex the fusion is.

In this paper we investigate the advantages of feature level fusion in the context of finger vein recognition. In feature level fusion multiple feature representations of the same biometric input data are combined to a new, single feature by using a certain fusion strategy. Feature level fusion is situated at the second step in the processing toolchain and thus less complex than score or decision level fusion but nevertheless able to greatly improve the recognition performance [Li13, YZ10]. Depending on the fusion strategy, fusion is able to mitigate erroneous feature extraction results from single feature extraction methods. To enable feature level fusion we use different finger vein feature extractors which all output a binary vein image. These binary feature images are fused into a new representation which is subsequently used for the task of recognition.

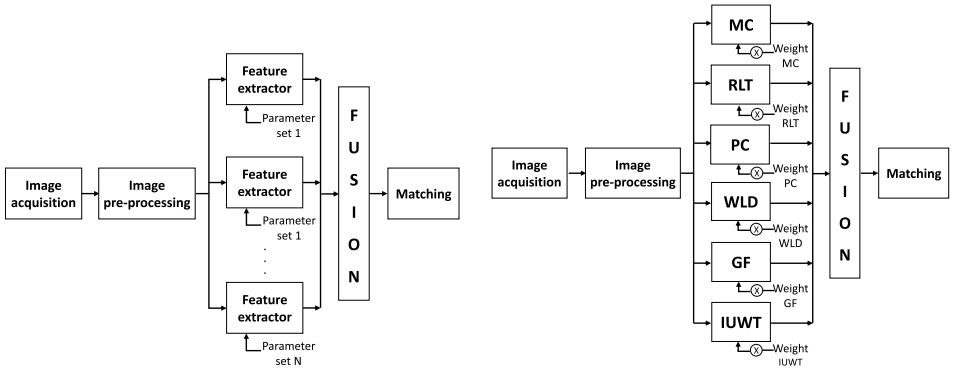


Fig. 2: Feature level fusion, left: considering the combination of features returned by a single algorithm, right: considering features obtained by applying different algorithms.

3.1 Fusion Strategies

We investigate two different strategies of fusion. In the first fusion strategy only a single feature extraction technique is applied several times on the same input image, varying each time the parameters of the algorithm. Tab. 1 shows the parameters that can be adjusted for each of the considered feature extraction techniques and their range of values. The block diagram for this kind of fusion is presented in Fig. 2 left.

The second fusion strategy (see Fig. 2 right) combines features obtained by applying different algorithms in order to obtain a single feature, later used in the matching stage. The fusion strategies exploited in both the proposed scenarios are summarised hereafter.

One of the simpler fusion strategies is **majority voting (MV)**: in our context the decision if a pixel is a vein or not is based on the majority vote/decision of all the feature extractors involved. If more than k of the extractors agree that a pixel belongs to a vein, it is

| Feature Extractor | Parameters | Ranges for parameters |
|-------------------|---|--|
| MC | parameter of the filter used for determining derivatives used to compute curvatures (σ) | $\sigma \in [1.9, 2.5]$ |
| RLT | number of random starting points (N), valley width (W), search radius (r) | $N \in [3000, 4000]$ $W \in [15, 21], r \in [1, 5]$ |
| WLD | radius of the circular neighbourhood region (r), thresholds used for binarisation (t), sum of neighbourhood threshold (g) | $r \in [6, 10]$ $t \in [0.5, 1], g \in [45, 160]$ |
| PC | parameter of the Gaussian kernel used to filter the image (σ), threshold used for thresholding the gradient (t) | $\sigma \in [1, 3]$ $t \in [0.3, 1]$ |
| GF | number of rotations (N), wavelength (λ) and bandwidth (bw) of the filter, size of the kernel (sz), aspect ratio of the Gaussian part of the filter (γ) | $N \in [8, 16], bw \in [0.5, 1.5]$ $sz \in [10, 15]$ $\gamma \in [0.3, 0.9]$ |
| IUWT | wavelet levels considered ($levels$) | $levels \in [1, 4]$ |

Tab. 1: Parameters used in the fusion stage.

marked as a vein in the final fused output, else it is marked as background pixel. Additional weighting can be applied to the single votes.

Another basic strategy is **weighted average (WA)**: The individual feature extraction results are weighted according to predefined weights and the fused output is the weighted average (or sum) of all the single outputs. This output is not a binary one at first, but it can be thresholded to get a binary output again.

STAPLE (Simultaneous Truth And Performance Level Estimation [WZW04]) is an algorithm for performance analysis of image segmentation approaches in medical imaging based on expectation-maximisation. It considers a collection of segmentations and computes a probabilistic estimate of the true segmentation and a measure of the performance level represented by each segmentation. The probabilistic estimate is obtained by an optimal combination of the different input segmentations. Each segmentation is weighted depending upon the estimated performance level and by incorporating a prior model for the spatial distribution of the segmented structures as well as spatial homogeneity constraints.

STAPLER (Simultaneous Truth And Performance Level Estimation with Robust extensions [La12]) is an extension of STAPLE which is able to deal with missing and repeated segmentations. The estimation of the ground truth and performance parameters is improved by using training data.

COLLATE (COnsensus Level, Labeler Accuracy and Truth Estimation [AL11]), another extension of STAPLE, follows a different strategy for the performance estimation. Confusion regions based on differences in labelling of aligned images are defined, leading to different performance estimates depending on the degree of consensus in the initial labelling.

Considering the fusion output there are two options: the binary vein images can be regarded as segmentations (veins and background) and are simply used as input for STAPLE/STAPLER/COLLATE to get a fused output image. In addition, the probabilistic esti-

mate can be used as fusion results and during matching. Results obtained using the probability map output are denoted as *Prob* in the results section.

4 Experiments

The experiments were conducted on the University of Twente Finger Vascular Pattern Database (UTFVP) [TV13]. It consists of 1440 images in total, taken from 60 subjects, 6 fingers per subject and 4 images per finger. Each image depicts exactly one finger. The images have a resolution of 672×380 pixels, a density of 126 pixels/cm and the width of the visible blood vessels is 4 – 20 pixels.

To determine the EER we followed the original procedure suggested by Ton et al. [TV13]. They used 10% of the images for training and parameter tuning and the remaining 90% for testing and EER determination, that is one finger of the first 35 users is used as a training set and the remaining 1300 images of the database are used for matching tests. The images of the testing set are matched between each other, avoiding symmetric matching. This procedure leads to a total of 1950 genuine matches and 842400 impostor matches. The genuine scores and impostor scores are used to obtain FMR and FNMR, respectively, and to compute the EER, which is used in this paper as an indicator of the recognition performance.

The training set is used for parameter optimisation of the feature extraction and fusion schemes (finding the best performing combinations) while the actual evaluations are performed on the testing set (842400 matches).

The evaluation scripts including a detailed listing of all the parameter combinations for the best fusion results stated in the following results section can be downloaded from: <http://www.wavelab.at/sources/Kauba16e/>

4.1 Results

Tab. 2 shows the baseline EER results for the single feature extractors, without performing any fusion ($EER_{Baseline}$), and the results obtained by fusing the features returned from a single feature extraction scheme applied several times on the same input image, varying each time its parameters according to Tab. 1. Different fusion techniques have been tested and the best performing parameter combination (in terms of EER using the aforementioned testing procedure) is stated.

It can be clearly seen that the MV technique is always able to improve the results for all feature extraction schemes. Most of the fusion schemes are able to improve the results for MC, PC and GF. Neither using STAPLE/STAPLER/COLLATE nor using *Prob* values has a clear advantage over the much simpler MV and WA fusion schemes. The best fusion results for each technique are also shown in the ROC curves of Fig. 3. The best result, i.e. an EER of 0.29%, is obtained for MC using COLLATE fusion. The FMR1000 values for the different combinations can be seen in the figures too.

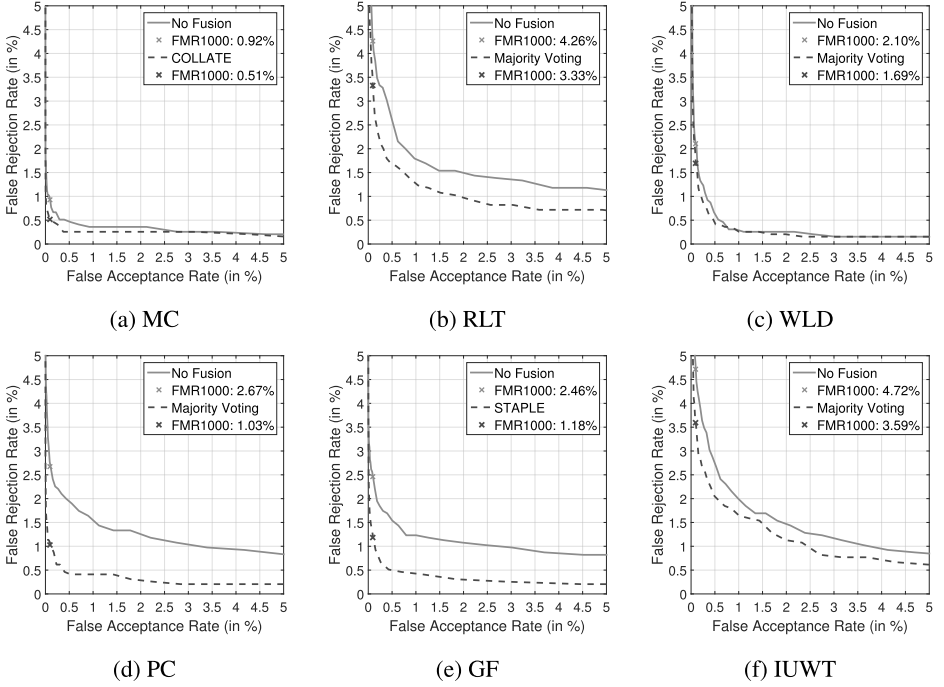


Fig. 3: ROC curves representing the impact on performance of the fusion of features obtained from the same algorithm.

| Feature Extractor | MC | RLT | WLD | PC | GF | IUWT |
|----------------------------|------|------|------|------|------|------|
| $EER_{Baseline}(\%)$ | 0.47 | 1.54 | 0.51 | 1.33 | 1.18 | 1.63 |
| $EER_{MajorityVoting}(\%)$ | 0.36 | 1.23 | 0.46 | 0.46 | 1.12 | 1.47 |
| $EER_{Average}(\%)$ | 0.31 | 1.55 | 0.66 | 0.46 | 1.39 | 1.54 |
| $EER_{STAPLE}(\%)$ | 0.36 | 2.05 | 0.76 | 0.56 | 0.55 | 1.53 |
| $EER_{STAPLE_Prob}(\%)$ | 0.41 | 1.23 | 0.71 | 0.56 | 0.55 | 1.67 |
| $EER_{STAPLER}(\%)$ | 0.35 | 1.71 | 0.71 | 0.56 | 0.60 | 1.48 |
| $EER_{STAPLER_Prob}(\%)$ | 0.36 | 1.53 | 0.71 | 0.56 | 0.64 | 1.69 |
| $EER_{COLLATE}(\%)$ | 0.29 | 1.69 | 0.46 | 0.51 | 0.61 | 1.83 |
| $EER_{COLLATE_Prob}(\%)$ | 0.51 | 2.41 | 1.37 | 1.02 | 1.53 | 3.18 |

Tab. 2: Performance obtained fusing the features returned by a single feature extractor

The results for the second fusion strategy (combination of distinct feature extraction schemes) are listed in Tab. 3. It can be seen that increasing the number of features considered does not always improve the recognition performance. However, with a few exceptions fusion is always able to improve the results compared to the best single feature extractor (MC with an EER of 0.5%). S_Prob and SR_Prob are the best fusion strategies with regard to all tested combinations while the feature combination in the second last row is the best with regard to all fusion schemes. The best performance is achieved when all 6 features

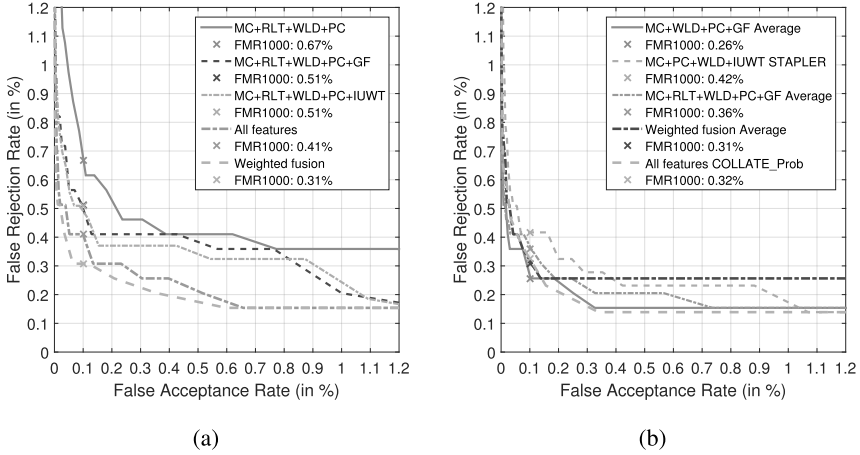


Fig. 4: ROC curves representing the impact on performance of the fusion of features obtained from the different algorithms. (a) Results for the majority voting fusion strategy. (b) Results of the best performing features combination.

| Feature Combination | MV | Av | S | S_Prob | SR | SR_Prob | C | C_Prob | mean F |
|------------------------------------|------|-------------|------|-----------|------|------------|------|-------------|--------|
| MC + RLT + PC + WLD | 0.41 | 0.67 | 0.46 | 0.31 | 0.56 | 0.56 | 1.64 | 1.38 | 0.75 |
| MC + PC + WLD + GF | 0.45 | 0.21 | 0.26 | 0.26 | 0.26 | 0.24 | 0.41 | 1.57 | 0.46 |
| MC + PC + WLD + IUWT | 0.46 | 0.32 | 0.31 | 0.32 | 0.27 | 0.32 | 0.42 | 1.68 | 0.51 |
| MC + RLT + PC + WLD + GF | 0.41 | 0.25 | 0.56 | 0.60 | 0.61 | 0.26 | 0.87 | 0.28 | 0.48 |
| MC + RLT + PC + WLD + IUWT | 0.37 | 0.46 | 0.31 | 0.25 | 0.35 | 0.31 | 0.61 | 0.46 | 0.39 |
| MC + RLT + PC + WLD + IUWT + GF | 0.26 | 0.41 | 0.28 | 0.23 | 0.34 | 0.32 | 0.28 | 0.19 | 0.29 |
| 2* MC + RLT + PC + WLD + IUWT + GF | 0.25 | 0.26 | 0.36 | 0.26 | 0.31 | 0.26 | 0.46 | 1.32 | 0.44 |
| mean value over all combinations | 0.37 | 0.37 | 0.36 | 0.32 | 0.39 | 0.32 | 0.67 | 0.98 | |

Tab. 3: Results obtained when features returned by different feature extractors are considered. The fusion strategies considered are majority voting (MV), weighted average (Av), STAPLE (S), STAPLER (SR) and COLLATE (C).

(MC, PC, WLD, GF, RLT and IUWT) are fused using the COLLATE, resulting in an EER of 0.185%. An ROC curve containing the best results obtained can be seen in Fig. 4.

Kauba et al. [KRU14] did their score level fusion evaluations also on the UTFVP dataset using the evaluation methodology of Ton et al. [TV13]. The best EER they were able to achieve is 0.25% for a combination of LBP, MC and AB. The second best result they achieved is an EER of 0.27% fusing the scores of MC, LBP and SIFT. As LBP and SIFT

do not output binary images, we cannot test these combinations using feature level fusion. But our results indicate that feature level fusion even improves the results over score level fusion on the UTFVP dataset.

Tab. 4 shows the runtimes of the MATLAB implementations of all the feature extraction schemes and Tab. 5 for the different fusion approaches (2nd fusion scheme) on the training set (140 images). The runtimes for the first fusion scheme can be estimated by taking the runtime of the single feature extraction stage multiple times according to the number of different parameters used during feature extraction and then adding the time for the fusion and matching process from Tab. 5. Each process was run 30 times and the values in the tables are the mean values of all runs. Note that matching the *Prob* values takes longer because decimal number are matched instead of binary values. The feature extraction times (*FE*) for the fusion approaches include the times for extraction of the single features as well as the matching times for score level fusion (*SLF*) include matching times of all included features. *4 Feat* means a combination of MC, PC, GB and WLD features. The combination achieving the best EER found so far (*h*) takes about 31 times longer than MC. The simpler (*b*) approach takes more than 8 times longer, which is in the same range as score level fusion, while cutting down the EER to 0.21. For the combinations including all 6 features *RLT* feature extraction takes a considerable amount of the total time. Comparing the runtimes and EERs of (*b*) and (*h*) does not justify the increased processing time for the small EER improvement over (*b*). Thus the simpler and especially faster feature level fusion schemes should be preferred. The advantage of feature level fusion over score level fusion is that the fusion is done only once at feature extraction stage before storing the biometric template, thus the increased processing time does not affect each matching (except if values are not binary any longer) operation which is of particular interest if it comes to identification.

| Method | Runtime (s) | | | | | |
|--------------------|-------------|--------|-------|--------|--------|-------|
| | MC | RLT | WLD | GF | IUWT | PC |
| Feature Extraction | 85.6 | 2639.8 | 65.56 | 143.66 | 132.14 | 7.32 |
| Matching | 97.31 | 413.5 | 193.2 | 1184.3 | 445.5 | 552.7 |

Tab. 4: Runtimes for single feature extraction variants

| Method | Runtime (s) | | | | | EER (%) |
|--------------------------|-------------|--------|----------|------|--------|-------------|
| | FE | FLF | Matching | SLF | Total | |
| (a) MC | 85.6 | - | 97.31 | - | 182.91 | 0.5 |
| (a) MV 4 Feat | 302.13 | 0.69 | 167.6 | - | 470.42 | 0.45 |
| (b) Av 4 Feat | 302.13 | 1.06 | 1217.67 | - | 1520.9 | 0.21 |
| (c) MV 6 Feat | 3064.1 | 0.79 | 240.99 | - | 3305.9 | 0.25 |
| (d) Av 6 Feat | 3064.1 | 1.18 | 1401.4 | - | 4466.7 | 0.26 |
| (e) S 4 Feat | 302.13 | 100.93 | 226.8 | - | 629.82 | 0.26 |
| (f) S 4 Feat <i>Prob</i> | 302.13 | 100.93 | 1193 | - | 1596 | 0.26 |
| (g) C 6 Feat | 3064.1 | 86.07 | 319.29 | - | 3469.4 | 0.28 |
| (h) C 6 Feat <i>Prob</i> | 3064.1 | 86.07 | 2561.3 | - | 5711.4 | 0.19 |
| (i) SLF 1 | 87.88 | - | 1870.9 | 0.26 | 1959 | 0.25 |
| (j) SLF 2 | 148.75 | - | 1268.7 | 0.26 | 1417.7 | 0.27 |

Tab. 5: Runtimes for different fusion variants (FLF... Feature Level Fusion)

5 Conclusion

We showed that feature level fusion is able to improve the recognition performance in terms of the EER. It turned out that even the simple fusion schemes including only one feature extractor improved the recognition performance leading to better EERs compared to the best EER of the single feature extractors. Fusion methods using a combination of different feature extraction schemes were able to improve the performance considerably.

Feature level fusion does neither add time, nor complexity or costs to collect additional biometric data, but the necessity to use different feature extraction methods adds computational complexity to the whole system. However the matching complexity is not inherently increased which is a crucial point if it comes to identification. We showed that the simple fusion schemes are able to improve recognition performance while maintaining a lower runtime compared to the more complex schemes.

Our future work will include tests with different finger vein databases in order to assess the generalizability of our approach.

Acknowledgment

This work has been partially supported by COST Action 1206 "De-identification for privacy protection in multimedia content". This work was partially funded by the Austrian Science Fund (FWF), project nr. P26630. P. Campisi gratefully acknowledges the support of NVIDIA Corporation with the donation of a Titan X used for this research. The UTFVP Database used for our tests was kindly provided by R.N.J. Veldhuis.

References

- [AL11] Asman, Andrew J; Landman, Bennett A: Robust statistical label fusion through consensus level, labeler accuracy, and truth estimation (COLLATE). *Medical Imaging, IEEE Transactions on*, 30(10):1779–1794, 2011.
- [Ch09] Choi, Joon Hwan; Song, Wonseok; Kim, Taejeong; Lee, Seung-Rae; Kim, Hee Chan: Finger vein extraction using gradient normalization and principal curvature. In: *IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics*, pp. 725111–725111, 2009.
- [Hu10] Huang, Beining; Dai, Yanggang; Li, Rongfeng; Tang, Darun; Li, Wenxin: Finger-vein authentication based on wide line detector and pattern normalization. In: *Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE*, pp. 1269–1272, 2010.
- [KRU14] Kauba, Christof; Reissig, Jakob; Uhl, Andreas: Pre-processing cascades and fusion in finger vein recognition. In: *Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG'14). Darmstadt, Germany, September 2014*.
- [KZ12] Kumar, Ajay; Zhou, Yingbo: Human identification using finger images. *Image Processing, IEEE Transactions on*, 21(4):2228–2244, 2012.
- [La12] Landman, Bennett A; Asman, Andrew J; Scoggins, Andrew G; Bogovic, John A; Xing, Fangxu; Prince, Jerry L: Robust statistical fusion of image labels. *Medical Imaging, IEEE Transactions on*, 31(2):512–522, 2012.

- [Li13] Liu, Fei; Yang, Gongping; Yin, Yilong; Xi, Xiaoming: Finger-vein recognition based on fusion of pixel level feature and super-pixel level feature. In: *Biometric Recognition*, pp. 274–281. Springer, 2013.
- [LLP09] Lee, Eui Chul; Lee, Hyeon Chang; Park, Kang Ryoung: Finger vein recognition using minutia-based alignment and local binary pattern-based feature extraction. *International Journal of Imaging Systems and Technology*, 19(3):179–186, 2009.
- [MNM04] Miura, Naoto; Nagasaka, Akio; Miyatake, Takafumi: Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification. *Machine Vision and Applications*, 15(4):194–203, 2004.
- [MNM07] Miura, Naoto; Nagasaka, Akio; Miyatake, Takafumi: Extraction of finger-vein patterns using maximum curvature points in image profiles. *IEICE transactions on information and systems*, 90(8):1185–1194, 2007.
- [Ot75] Otsu, Nobuyuki: A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.
- [Pi16] Piciuccio, E.; Maiorana, E.; Kauba, C.; Uhl, A.; Campisi, P.: Cancelable Biometrics for Finger Vein Recognition. In: *2016 International Workshop on Sensing, Processing and Learning for Intelligent Machines (SPLINE)*. 2016.
- [Qi13] Qin, Huafeng; Qin, Lan; Xue, Lian; He, Xiping; Yu, Chengbo; Liang, Xinyuan: Finger-vein verification based on multi-features fusion. *Sensors*, 13(11):15048–15067, 2013.
- [RNJ06] Ross, Arun A; Nandakumar, Karthik; Jain, Anil: *Handbook of multibiometrics*, volume 6. Springer Science & Business Media, 2006.
- [SFM07] Starck, Jean-Luc; Fadili, Jalal; Murtagh, Fionn: The Undecimated Wavelet Decomposition and its Reconstruction. *IEEE Transactions on Image Processing*, 16(2):297–309, feb 2007.
- [TV13] Ton, B.T.; Veldhuis, R.N.J.: A high quality finger vascular pattern dataset collected using a custom designed capturing device. In: *International Conference on Biometrics, ICB 2013*. IEEE, 2013.
- [WZW04] Warfield, Simon K; Zou, Kelly H; Wells, William M: Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation. *Medical Imaging, IEEE Transactions on*, 23(7):903–921, 2004.
- [YYW12] Yang, Ying; Yang, Gongping; Wang, Shibing: Finger Vein Recognition Based on Multi-instance. *International Journal of Digital Content Technology & its Applications*, 6(11), 2012.
- [YZ10] Yang, Jinfeng; Zhang, Xu: Feature-level fusion of global and local features for finger-vein recognition. In: *Signal Processing (ICSP), 2010 IEEE 10th International Conference on*. IEEE, pp. 1702–1705, 2010.
- [YZ12] Yang, Jinfeng; Zhang, Xu: Feature-level fusion of fingerprint and finger-vein for personal identification. *Pattern Recognition Letters*, 33(5):623–628, 2012.
- [ZK10] Zhou, Yingbo; Kumar, Ajay: Contactless palm vein identification using multiple representations. In: *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*. IEEE, pp. 1–6, 2010.
- [Zu94] Zuiderveld, Karel: Contrast limited adaptive histogram equalization. In: *Graphics gems IV*. Academic Press Professional, Inc., pp. 474–485, 1994.