Fingerprint Indexing via BRIEF Minutia Descriptors

Robert Pollak* and Roland Richter

Department of Knowledge-Based Mathematical Systems,
Johannes Kepler University Linz,
Altenbergerstraße 69, 4020 Linz, Austria
E-mail: robert.pollak@jku.at
E-mail: roland.richter@flll.jku.at
*Corresponding author

Abstract: We use BRIEF binary local image descriptors as minutia descriptors for indexing of biometric fingerprint databases. Tests with varying descriptor size and parametrization are performed on a proprietary database. Compared with the speed of an implementation of conventional minutiae matching, we find that BRIEF descriptors are fast enough for database indexing. The tested descriptors outperform two other image descriptors (LBP, HoG) from recent literature with respect to matching rates and average penetration rates.

Keywords: biometrics; fingerprints; indexing; BRIEF; image descriptors.

1 Introduction

1.1 Fingerprint indexing

Executive state agencies have aggregated huge fingerprint databases nowadays. In order to find a given fingerprint image against such a database, it is not feasible any more to do a brute-force search across all entries. The database needs to be indexed instead. This means that each fingerprint image is represented not only by its match template, but also by a feature vector (also called “descriptor” or “hash”). These vectors do not need to have same lengths; they have to be comparable however. The representation needs to be selected such that images of the same finger yield feature vectors that are as similar as possible, and that vectors of different fingers are as different as possible. The set of feature vectors is then called “index”. Measuring feature vector similarity defines the database search order.

The identification of a query image then works in the following steps:

1. Compute the query finger’s feature vector.
2. For each index entry, compute the similarity of the entry and the query finger’s feature vector.
3. Sort all database entries in descending order of similarity.
4. For each entry in the sorted database, try to match the query finger until a match is found, or the database is fully scanned.

If, in step 3, the best “candidates” are sorted to the front of the database, then in step 4 only a few finger-finger matchings are necessary. Hence, for such an index to be useful, comparing two feature vectors in step 2 needs to be extremely fast compared to ordinary finger-finger matching. Additionally, it is desirable that feature vector computation is fast, and that the vectors are small in terms of computer memory consumption.

Many indexing representations have already been proposed. Surveys are given e.g. by Maltoni et al. (2009), Cappelli, Ferrara, and Maio (2011), and Yang (2011). One can distinguish between approaches that only consider minutia (or minutiae triplet) information, and others which also process image or orientation information:

Representation by minutiae: In fingerprint matching, most of the known methods are based on minutiae. This suggests using them for indexing, too.

An example is the “Minutia Cylinder Code” (Cappelli, Ferrara, and Maltoni, 2010) — a minutia representation that only uses standardized minutiae templates and is time efficient (Cappelli, Ferrara, and Maltoni, 2011).

Xu et al. (2009) present a method based on the Fourier-Mellin transformation. All minutiae are represented as Gauss functions (i.e. smoothed Dirac functions) at their respective position. This creates a descriptor of an image’s whole minutiae set which is invariant to translation, rotation and scaling, and whose size is fixed (but possibly very large).

Representation by image information: Most of the known fingerprint indexing methods integrate information from the initial grayscale image or from the orientation image into the feature vector computation.

For example, Nanni and Lumini (2009) decompose a fingerprint image into a grid of overlapping squares. For each of these squares several characterization numbers are computed. Together with a complex distance function those numbers are then used to estimate the similarity of two images.

Another variant is to center a fingerprint image at its core for generating a feature vector. This happens e.g. in FingerCode (Jain et al., 2000), where a series of centered, Gabor-filtered images are processed.

Many works follow a hybrid approach (called “texture-based local structures” by Maltoni et al., 2009, section 4.4.3): each minutia is characterized by a descriptor; it is computed from the local minutia environment in the fingerprint image, the orientation image or similar. The feature vector corresponding to the fingerprint image then consists of all minutiae’s descriptors. The present work also takes this hybrid approach.

1.2 Image based minutia descriptors

A minutia descriptor contains information that, in addition to position and direction, makes a minutia better identifiable and discriminable from other minutiae. In combination with a similarity measure on descriptors, these are suitable for finding pairs of matching minutiae between the queried fingerprint image and the reference image. This “local matching” (or “correspondence problem”, Feng and Zhou, 2011) serves to measure similarity between query and reference image.

In recent years, many minutia descriptors have been proposed and evaluated. An overview is given by Feng and Zhou (ibid.), where three kinds of minutia descriptors are described:

- Image based: The descriptor is computed directly from the image in the minutia environment (either from the original grayscale image, or after preprocessing like e.g. binarization).

- Texture based: The descriptor is determined by orientation and ridge frequency in the environment of the minutia. Examples are (Feng, 2008) or the SIFT-based descriptor of Park, Pankanti, and Jain (2008).

- Minutiae based: Here one can distinguish between "nearest-neighbor based" and "fixed-radius based" descriptors (Feng and Zhou, 2011).

Subsequently, Feng and Zhou (ibid.) evaluate seven different minutia descriptors in four separate scenarios. In three of those scenarios (good quality, bad quality, large distortion) Minutia Cylinder Code descriptors (Cappelli, Ferrara, and Maltoni, 2010) score best; in one scenario (low overlap) a texture-based descriptor (Feng, 2008) is best.

Nanni and Lumini (2009) propose a method for image based matching with several different texture descriptors, including Gabor filters, invariant Local Binary Patterns (LBP, Ojala, Pietikäinen, and Harwood, 1996; Ojala, Pietikainen, and Maenpaa, 2002), and Histogram of Gradients (HoG, Dalal and Triggs, 2005). From these descriptors, the authors generate 176 features by varying parameter configurations. Sequential forward floating selection (SFFS) is then used to select the most capable configurations.

An extensive overview of image based minutia descriptors is given by Yang (2011). In addition to the above mentioned ones, Yang (ibid.) also describes descriptors that are based on Discrete Wavelet Transformation and Discrete Cosine Transformation, as well as descriptors based on statistical moments.

1.3 Binary local descriptors

Algorithms for the extraction of local image properties like SIFT (Scale-Invariant Feature Transform, Lowe, 2004) or SURF (Speeded Up Robust Features, Bay et al., 2008) were created in recent years for dealing with general image processing tasks like matching, tracking, stitching, and 3D reconstruction. These algorithms usually consist of two steps: a "detector" finds prominent "interest points", and a descriptor computes preferably unique representations of these points.

Time and memory efficiency is in focus here. One approach is to represent descriptors as binary vectors that can be processed and compared faster (Heinly, Dunn, and Frahm, 2012). Examples for such efficient binary descriptors are BRIEF (Binary Robust Independent Elementary Features, Calonder et al., 2010; Calonder et al., 2012) and ORB (Oriented FAST and Rotated BRIEF, Rublee et al., 2011).

In the context of fingerprint recognition, two approaches are possible: Extractors like SIFT or SURF can be used both as a detector of interest points, and as a descriptor to characterize them; this is e.g. done by Park, Pankanti, and Jain, 2008 and He, Zhang, and Hao, 2010. Since different detectors and descriptors can be combined (Heinly, Dunn, and Frahm, 2012), it is also possible to use an existing minutiae extraction as detector, and add another descriptor to enhance minutiae differentiability. This approach has not been followed often; one example is (Laadjel et al., 2010), where palm prints are matched via minutiae.

1.3.1 BRIEF descriptors

Base idea of the BRIEF descriptor (Calonder et al., 2010; Calonder et al., 2012) is that the environment of a point can be characterized by a number of pair-wise gray value comparisons. For this, a set of n random, but fixed point pairs (p_i, q_i) is generated as a test pattern for an $s \times s$ environment. For the environment $U(M)$ of a minutia M, this pattern defines a series of n binary tests as

$$t(U(M); p_i, q_i) = \begin{cases} 1 & \text{if } I(U(M), p_i) < I(U(M), q_i) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

with $I(U(M), p)$ being the (gray) value of the environment $U(M)$ at position p. The result is a binary string of length n that characterizes the environment and therefore its center M. The authors of ibid. further examine several variants: either the test pattern is rotated (oriented, O-BRIEF), scaled (S-BRIEF), or used upright and unscaled (U-BRIEF). Also different point pair distributions are tested there.

2 Method and test setup

2.1 Database

Our industry partner has provided us with a set of 4658 fingerprint images, recorded from a capacitive sweep sensor with 500 dpi resolution. This image set was taken from 96 persons. It contains about 8 images per finger, mostly from the forefingers, middle fingers, and ring fingers. The given data was not selected specially for our work. Moreover, we did not remove any images from the database, although some images show “stitching artifacts” from image acquisition and some images only display a quite small finger contact area.

For our indexing tests, we assigned the first image of each finger to the reference database of size 466. The other 4192 images serve as query fingers, for which the matching finger is searched in that database.

In our tests, fingerprint images were preprocessed by the proprietary fingerprint software of our company partner, written in C++. In particular, minutiae were computed for each fingerprint image. On average, 26.7 minutiae were found per image. For 85 images, or 1.8 % of the database, ten or less minutiae were found, which might be problematic for any minutiae-based matching algorithm.

2.2 Descriptors

The aim of the present work was to find a time and memory efficient minutia descriptor for fast and still powerful “local matching” of fingerprints, allowing to index large fingerprint databases. Since our data was recorded with a sweep sensor, the images are oriented quite upright. In our database, nearly all fingerprint images (99.9 %) are rotated by less than 11 degrees w.r.t. other images of the same finger. Hence, the wanted descriptor only needs minor rotation invariance. Additionally, it does not need to be scale invariant.

As we strived for fast descriptors, a simple descriptor such as BRIEF seemed to be more feasible than more complex descriptors such as SIFT and SURF. We have implemented the following descriptors in C++, embedded in the proprietary fingerprint matching software of our industry partner. In this environment, a list of detected minutiae (position and direction), smoothed gray-level and binarized versions of the original image, as well as an orientation field are already available.

2.2.1 BRIEF descriptors

In order to optimize both for discriminatory power and computation speed (and, to a lesser extent, memory size), we considered some variations of the BRIEF descriptor.

- Descriptor length: The length of a descriptor obviously influences the time it takes to compute and compare it, as well as the amount of information it can hold. Calonder et al. (ibid.) considered descriptor of 16, 32, and 64 bytes in size. In order to determine an optimal BRIEF size, we tested BRIEF descriptors of 8, 16, 24, 32, and 64 bytes.
- Spatial distribution: In the process of creating the random test pattern for BRIEF descriptors, one might impose some restrictions. Calonder et al. (ibid.) give several examples for such variations. We tested "free" patterns generated without any restrictions versus patterns which were "restricted" in the following way: circular instead of quadratic patch; no reuse of test points (to impede fingerprint reconstruction); and a minimum distance of compared points (to integrate knowledge about possible ridge frequencies). Note that enforcing points not to be reused also limits the number of possible point pairs, i.e. the size of the maximum descriptor.
- Binarized vs. gray-level image: As binarized images were available in our environment, we tested whether using binarized images makes computations faster or not. In this case, we used $I(U(M), p_1) \neq I(U(M), p_2)$ (instead of "<") in test $t(U(M); p_1, p_2)$.
- Minutiae orientation: Since directions of all minutiae were already computed, we could easily test the effect of using the test pattern "upright" versus using the test pattern "oriented", i.e. rotated into minutiae direction.

Combining these possibilities gives a number of BRIEF descriptor variants. Our names denote whether the descriptors are free ("F") or restricted ("R"), "binarized image" ("BN") or "gray-level image" ("GL"), upright ("U") or oriented ("O"), and finally give the size of the test pattern in bytes. For instance, RGLU-BRIEF-8 denotes a BRIEF descriptor which uses a restricted test pattern of 8 bytes, used upright on the gray-level image. Figure 1 shows three possible BRIEF patterns.

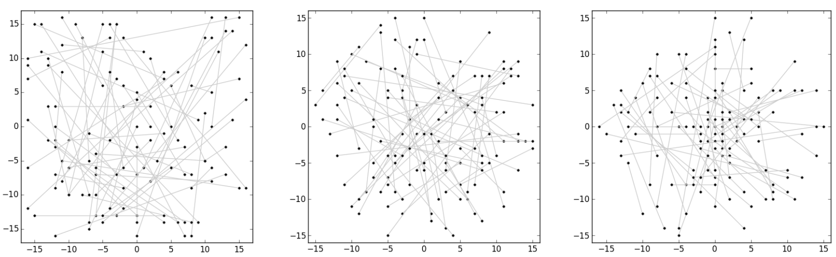

Figure 1 Examples of BRIEF patterns: left: even distribution on a square; middle: even distribution on a disk, with additional constraints; right: even distribution in polar coordinates.

2.2.2 HoG descriptor

Histogram of Gradients was introduced by Dalal and Triggs (2005) to detect pedestrians in arbitrary images. Nanni and Lumini (2009) examined HoG and other methods for their suitability to match fingerprints. In our test environment, a computed orientation field is already available. We calculated the HoG descriptor via a histogram of 12 bins, sampled in an $s \times s$ environment of a minutia M, based on the orientations relative to the minutia orientation. The histogram was then normalized, resulting in a descriptor of 12 bytes in size.

2.2.3 LBP descriptor

Local Binary Patterns was first proposed by Ojala, Pietikäinen, and Harwood (1996) as a powerful descriptor of textured images that is easy to compute. In the original version, the eight neighbors of the point M are compared with the grayscale value of M: smaller values are set to 0, larger ones to 1. By this, one gets eight bits, encoding a value from 0 to 255. Later, Ojala, Pietikainen, and Maenpaa (2002) refined and extended this method to larger environments to get invariant descriptors. There, not only a single value, but a histogram of values in the environment of M is generated.

For our approach, we reduced the number of neighbors to four. Hence, each pixel is encoded as a value between 0 and 15. The frequency of these values is collected in a histogram with 16 bins over an $s \times s$ environment of M, resulting in a descriptor of 16 bytes.

2.3 Fingerprint matching

In order to compare a query finger with the database index, two sets of minutia descriptors need to be compared for each index entry. For this, we iteratively find the best matching descriptor pair, until only dissimilar descriptors are left over (dependent on a threshold). To find these best pairs, we use the Hamming distance of the bit strings for BRIEF descriptors, and the cumulated histogram difference for HoG and LBP descriptors. Throughout our tests, two different fingers of the same person were not allowed to match, even if this false match would be harmless in practice.

As described, directions of matching minutiae do not differ very much in our database. We therefore only need to compare the descriptors of those minutiae whose directions are similar. This reduces computation time as well as the frequency of accidental descriptor matches (which reduce the indexing result).

3 Results

In the following, we report the results of two different test runs. In the first test, we compared BRIEF descriptors of several sizes, namely all ten RGL[U/O]-BRIEF-[8/16/24/32/64] variants. In the second run, we fixed the size of BRIEF descriptors to 16 bytes, and compared all eight [F/R][BN/GL][U/O]-BRIEF-16 variants with HoG-12 and LBP-16 descriptors.

Since a BRIEF descriptor is computed by a random test pattern, we repeated BRIEF tests 100 times, and report the minimum, mean, and maximum results. For HoG and LBP descriptors, it suffices to run them once.

3.1 Indexing performance

A well established way to measure the performance of fingerprint indexing and retrieval methods is the average penetration rate (Maltoni et al., 2009). It is the database search depth necessary to find the first matching entry, averaged over all query fingers. In our case, a value of e.g. 1.00 % means that, on average, only 4.66 entries in our database have to be examined until a match is found.

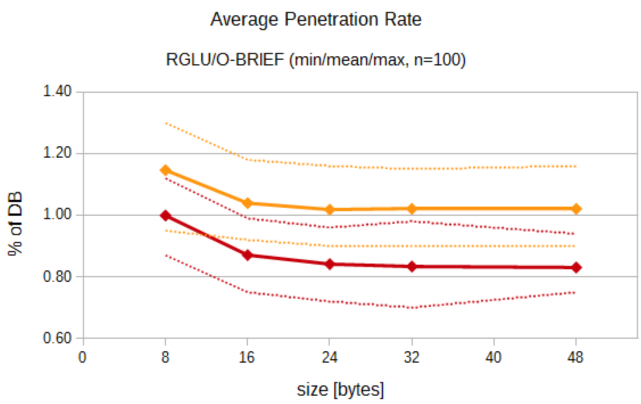

Figure 2 Average penetration rate per BRIEF descriptor size. Maximum (i.e., worst), mean and minimum (i.e., best), in percent of the database.

Figure 2 shows the average penetration rate of our tests using different BRIEF descriptor sizes. As is to be expected, larger descriptor sizes result in lower (i.e. better) average penetration rates. There is a notable performance increase when switching from BRIEF-8 to BRIEF-16 descriptors. However, further increasing BRIEF size beyond 16 bytes yields only marginal increases in performance.

Hence, it was reasonable to fix this size to 16 bytes, and test several BRIEF variants in comparison to HoG-12 and LBP-16. The result of this second test run is shown in figure 3. Roughly speaking, a “restricted” pattern performs better than a “free” one; sampling from gray-level images is better than sampling from binarized images; and rotating the BRIEF pattern in minutia orientation is better than using the BRIEF pattern upright. Also note that most BRIEF variants outperform HoG-12 and LBP-16.

Another way to compare different indexing algorithms is the percentage of unmatched queries dependent on the searched database portion (see ibid., Cappelli, Ferrara, and Maltoni, 2011). This number is called “error rate”, the used percentage of the database is called “penetration rate”, and their relation is called “indexing performance”. Note that these measures assume perfect minutiae matching in order to examine indexing quality independent of minutiae matching quality.

There is a trade-off between error rate and penetration rate: the more of the database is accessed (with corresponding duration), the more entries are found. Zero percent penetration means 100 % error, while useful indexing means that low error rates can already be reached with much less than 100 % penetration.

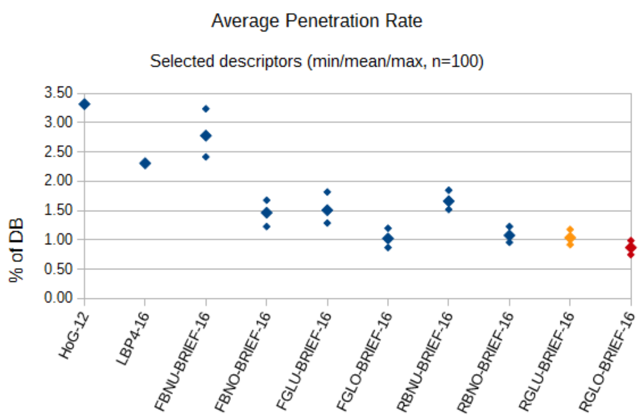

Figure 3 Average penetration rate of descriptor types and variants. Maximum (i.e., worst), mean and minimum (i.e., best), in percent of the database.

Figure 4 shows indexing performances for some selected descriptors. The error rate of HoG and LBP smoothly decreases with increasing penetration rate. There is a notable error rate "plateau" from about 30 % and 60 % penetration rate when using RGLO-BRIEF-16. Figure 4 (right) shows that this behavior can be seen for many BRIEF-16 descriptors. The reason for this behavior remains yet to be determined.

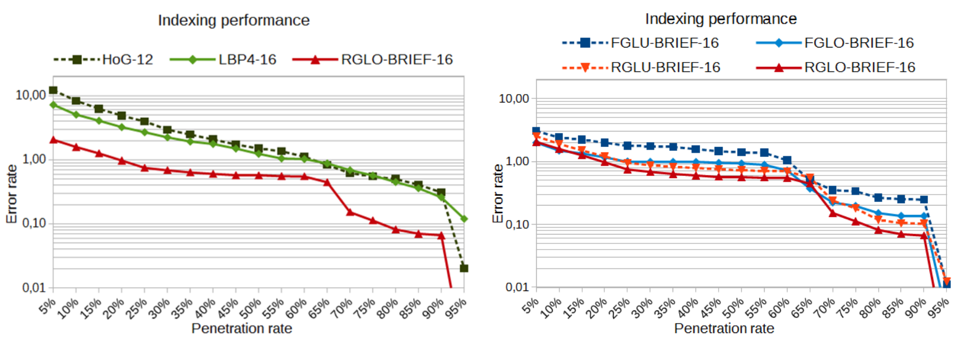

Figure 4 Comparing indexing performance of several tested descriptors ($n = 100$).

One cannot expect that the matching procedure described above can be a replacement for established fingerprint matching algorithms. However, the method often sorts the correct entry to the very first position of the database. The frequency of this event can be seen as the "matching rate", and is shown in figure 5. In contrast to the average penetration rate, the matching rate tends to increase with BRIEF descriptor size beyond BRIEF-16.

3.2 Index operation speeds

Whereas the proprietary fingerprint software we work with usually runs on embedded hardware, our tests took place on conventional desktop processors. Hence, it is difficult to

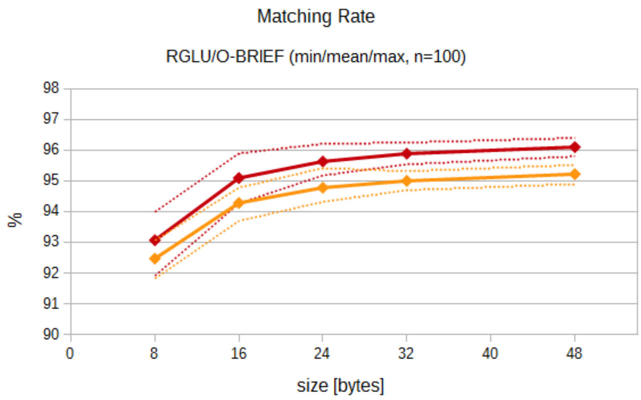

Figure 5 Matching rate by BRIEF descriptor size. Minimum (i.e., worst), mean and maximum (i.e., best).

Computation step	Time
Compute RGLU-BRIEF-16 descr., per image	58.4
Match RGLU-BRIEF-16, per image pair	9.2
Compute RGLO-BRIEF-16 descr., per image	298.5
Match RGLO-BRIEF-16, per image pair	9.4
Match minutiae template, per image pair	156.3

Table 1 Average measured durations of computation steps (in microseconds, $n = 100$).

compare absolute time measurements. It is reasonable, however, to compare the durations of individual computation steps relative to each other.

In our timing experiments, we observed no significant difference between the F/R and BN/GL variants of descriptor computation. The only notable difference was that the computation of oriented descriptors is about five times more expensive than computation of the unoriented counterparts. This does not matter much, however, since the database index only needs to be precomputed once, and the query image's descriptors also just needs to be computed once.

Table 1 shows average computation times for the two descriptors RGLU-BRIEF-16 and RGLO-BRIEF-16, as well as for the conventional extraction and matching algorithms of the proprietary fingerprint software. Note that these times do *not* include time for image preprocessing and the extraction of minutiae.

As stated above, comparing the descriptors of two images must be extremely fast for indexing to be useful. This is indeed the case here, as BRIEF-16 descriptor matching is about 15 times faster than conventional template matching. Even for databases of small size, the additional cost of computing the query finger's BRIEF descriptors is outweighed by the fact that only a small number of conventional template matchings have to be made afterwards.

4 Conclusion

We have implemented a series of image based minutia descriptors into fingerprint matching software. Indexing comparisons with two state-of-the art methods on a fingerprint image database show that oriented BRIEF descriptors give better indexing performance and average penetration rates with at the same time low computational effort and memory consumption. We therefore conclude that BRIEF minutia descriptors are suitable and beneficial for fingerprint indexing.

Additionally, for some descriptors, matching rates of more than 95 % can be achieved. This means that matching based on BRIEF descriptors can be used as a supplement to conventional template matching.

References

Bay, Herbert et al. (2008). “Speeded-Up Robust Features (SURF)”. In: *Computer Vision and Image Understanding* 110.3. <ce:title>Similarity Matching in Computer Vision and Multimedia</ce:title>, pp. 346–359.

Calonder, M. et al. (2012). “BRIEF: Computing a Local Binary Descriptor Very Fast”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.7, pp. 1281–1298.

Calonder, Michael et al. (2010). “BRIEF: Binary Robust Independent Elementary Features”. In: *European Conference on Computer Vision*. Ed. by Kostas Daniilidis, Petros Maragos, and Nikos Paragios. Vol. 6314. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 778–792.

Cappelli, R., M. Ferrara, and D. Maltoni (2010). “Minutia Cylinder-Code: A New Representation and Matching Technique for Fingerprint Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.12, pp. 2128–2141.

— (2011). “Fingerprint Indexing Based on Minutia Cylinder-Code”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.5, pp. 1051–1057.

Cappelli, Raffaele, Matteo Ferrara, and Dario Maio (2011). “Candidate List Reduction Based on the Analysis of Fingerprint Indexing Scores”. In: *IEEE Transactions on Information Forensics and Security* 6.3, pp. 1160–1164.

Dalal, N. and B. Triggs (2005). “Histograms of oriented gradients for human detection”. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 1, 886–893 vol. 1.

Feng, Jianjiang (2008). “Combining minutiae descriptors for fingerprint matching”. In: *Pattern Recognition* 41.1, pp. 342–352.

Feng, Jianjiang and Jie Zhou (2011). “A Performance Evaluation of Fingerprint Minutia Descriptors”. In: *International Conference on Hand-Based Biometrics*, pp. 1–6.

He, Shihua, Chao Zhang, and Pengwei Hao (2010). “Clustering-Based Descriptors for Fingerprint Indexing and Fast Retrieval”. In: *Asian Conference on Computer Vision*. Vol. 5994. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 354–363.

Heinly, Jared, Enrique Dunn, and Jan-Michael Frahm (2012). “Comparative Evaluation of Binary Features”. In: *European Conference on Computer Vision*. Ed. by Andrew Fitzgibbon et al. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 759–773.

Jain, A.K. et al. (2000). “Filterbank-based fingerprint matching”. In: *IEEE Transactions on Image Processing* 9.5, pp. 846–859.

Laadjel, Moussadek et al. (2010). “Partial Palmprint Matching Using Invariant Local Minutiae Descriptors”. In: *Transactions on Data Hiding and Multimedia Security V*. Vol. 6010. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 1–17.

Lowe, David G. (2004). “Distinctive Image Features from Scale-Invariant Keypoints”. English. In: *International Journal of Computer Vision* 60.2, pp. 91–110.

Maltoni, D. et al. (2009). *Handbook of fingerprint recognition*. Springer-Verlag, New York.

Nanni, Loris and Alessandra Lumini (2009). “Descriptors for image-based fingerprint matchers”. In: *Expert Systems with Applications* 36.10, pp. 12414–12422.

Ojala, Timo, Matti Pietikainen, and Topi Maenpaa (2002). “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7, pp. 971–987.

Ojala, Timo, Matti Pietikäinen, and David Harwood (1996). “A comparative study of texture measures with classification based on featured distributions”. In: *Pattern Recognition* 29.1, pp. 51–59.

Park, Unsang, Sharath Pankanti, and AK Jain (Mar. 2008). “Fingerprint verification using SIFT features”. In: *Proceedings SPIE 6944*. Vol. 6944, 69440K.

Rublee, E. et al. (2011). “ORB: An efficient alternative to SIFT or SURF”. In: *IEEE International Conference on Computer Vision*, pp. 2564–2571.

Xu, Haiyun et al. (2009). “Fingerprint Verification Using Spectral Minutiae Representations”. In: *IEEE Transactions on Information Forensics and Security* 4.3, pp. 397–409.

Yang, Jucheng (June 2011). “Non-minutiae Based Fingerprint Descriptor”. In: *Biometrics*. Ed. by Jucheng Yang. InTech. Chap. 4.