Improving frequency estimation of fingerprint minutiae configurations using automated pre-selection

A. Stuurman Dr. A. de Jongh

July 17, 2020

ABSTRACT: Likelihood ratio based statistical reporting on the comparison of fingerprints is thought to convey more information than traditional discrete reports such as a yes or no answer. Additionally, using a weight for each minutia based on their rarity improves the precision of the comparison process. During this research a tool was designed that assists in estimating the rarity of configurations of one or two minutiae with a core or delta as a reference position. It performs a pre-selection of fingerprints from a database by utilizing an existing minutia extraction and encoding tool and using an orientation and pixel position based matching technique. The found fingerprints are expected to be further examined manually, but since part of the database is automatically discarded the workload is reduced. The minimum requirements set are shown to be marginally reached, after which the shortcomings of the implementation are analyzed, and additional and alternative techniques are introduced that are expected to improve the tool's performance.

KEYWORDS: forensics, fingerprint comparison, Bayesian statistics, automated fingerprint identification system

For a list of terms, see page 10.

1 Introduction

1.1 Statistical reporting on fingermark and fingerprint comparison

In forensic casework, it is often necessary to determine if a found fingermark, the impression left by a finger, originates from a specific person. This is done by comparing the fingermark to a set of reference fingerprints taken from the fingers of that individual in a controlled environment. Recently, interest has increased in the concept of reporting this comparison statistically. Currently this is often done categorically, for example with a clear yes or no answer. This statistical method is already the standard in DNA profiling, as C. Neumann et al. state in their research on this topic[1]. Similar to Neumann's suggestion, the Netherlands Forensic Institute uses a likelihood ratio approach, which measures the odds of one event occurring over another, and is calculated by dividing the probability of the first event occurring over the probability of the second event occurring. The two events here are:

- Observing similar features between the fingermark and the reference fingerprint in the case that the mark was left by the reference donor.
- Observing similar features between the fingermark and the reference fingerprint in the case that the fingermark was left by another person.

1.2 Fingermark rarity estimation

The probability of observing similar features between the fingermark and the reference fingerprint in the case that the mark was left by the reference donor can simply be set to one. Naturally, if not all features between the fingermark and fingerprint correspond, the fingerprint is discarded, and calculation of the statistic is not required. As such only the probability of observing similar features between the fingermark and the reference fingerprint in the case that the fingermark was left by another person needs to be determined. This is estimated at the Netherlands Forensic Institute by determining the frequency of occurrence of the fingermark in a database

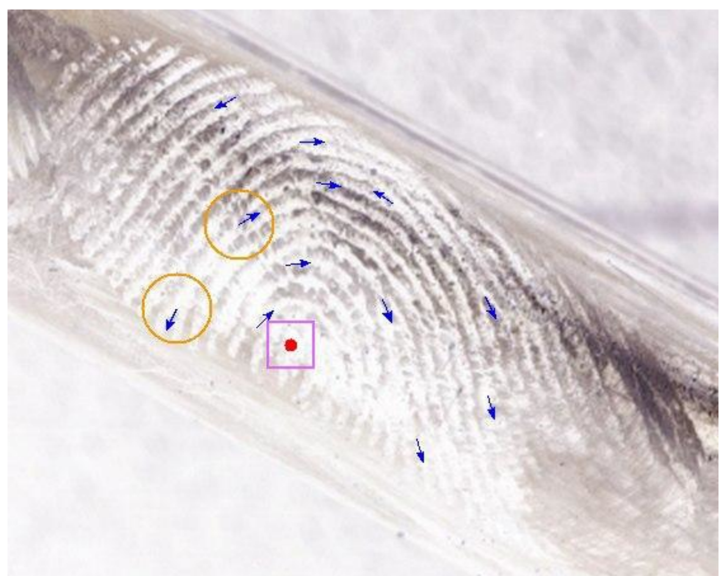

Figure 1: Example of a minutia configuration in a fingermark. The arrows represent minutiae, and the dot the core. Minutiae included in the configuration are marked with circles, and the reference point is noted by the square.

of fingerprints of the Dutch population. For this its pattern type, core-delta ridge distance, and minutiae are considered, which are distinctive features of a fingerprint[2]. The rarity of different pattern types has been researched by A. de Jongh et al.[3], showing that reliable numbers can be achieved. A similar group of researchers is currently working on core-delta ridge distance statistics. However, a complete set of minutiae is usually so rare that reliably estimating its frequency of occurrence would require a very large database. To counteract this, the fingermark is split into configurations of one or two minutiae, each containing a core or delta as reference position. See figure 1 for an example. The rarities of these configurations are then individually determined by looking at fingerprints in the database with the matching pattern type and core-delta ridge distance. Finally, the gained knowledge is used to estimate the rarity of the complete fingermark.

1.3 Process automation

The frequency estimation process is manually highly intensive. For this reason, only around one hundred fingerprints are examined for each small minutia configuration, heavily limiting the reliability of the frequency estimation. Automated tools are already in use in related forensic areas using fingerprints, for example for searching a large database to find the fingerprints that best match a certain fingermark. A similar tool that can assist in frequency estimation would be highly valuable, as it can improve the quality of the measurement and reduce the workload on the examiners.

Besides forensic comparison, such a tool would be of use in scientific work. Research on the relationships between fingerprint features, for instance in the aforementioned work[3], could be extended to include minutia configurations, for which automation would be very useful.

At the Netherlands Forensic Institute a tool has been created that filters fingerprints from a database by their pattern and core and delta ridge distance. However, filtering by small minutia configuration is still performed manually. The database created by A. de Jongh et al.[3] during their previous research was manually examined and extended to include core and delta positions and the ridge distance between them, in addition to the already known pattern types. However, the manual encoding of minutiae is very labor intensive, making their addition to the database not feasible, which is why the existing tool currently does not support searching by minutia configuration. A tool that filters a database by a configuration of one or two minutiae and a reference core or delta needs to be found, one which is able to automatically analyze the image of each fingerprint.

1.4 Development of a new tool

Many tools for the comparison of fingerprints exist, often categorized as being an Automated Fingerprint Identification System(AFIS). However, a tool suitable for minutia configuration frequency estimation is not widely available. Therefore, a new tool needs to be developed specifically for performing rarity estimation. When compared to existing tools, two relevant improvements can be distinguished, as discussed below.

1.4.1 Minimum minutia amount

Existing tools do not support searching by one or two minutiae. The lowest required minimum amount of minutiae found during this research in an existing tool was five, which was in the Motorola Printrak Biometric Identification Solution 9.1. It is logical that such a restriction exists, as the information contained in a configuration drops quickly when removing minutiae, increasingly reducing comparison performance. Figure 1 visually demonstrates the information lost when using a subset of the minutiae.

A new tool that searches by one or two minutiae would thus have a significantly lower performance than existing tools. Because of this, the tool cannot reliably perform the complete frequency estimation process. Instead, it should function as an assistant, filtering fingerprints of which is it certain do not contain the configuration, and leaving the remaining fingerprints for manual examination. If the parameters of the tool are set correctly, it may be possible to achieve a useful filter rate, while finding most of the fingerprints containing the minutia configuration.

1.4.2 Statistical measure

A new tool can take advantage of the the ability to exclude any fingerprint from the database without influencing the final frequency measure, as long as no correlation exists between the discarding of a fingerprint and it containing the required minutia configuration. An example use of this would be the omission of blurry images, which would make image processing more reliable. This is something existing tools cannot do, as they are required to consider every fingerprint in their database.

1.5 Research objective

The goal of this research was twofold. Firstly, to develop a tool that can make a pre-selection of fingerprints from a database of fingerprint images that might contain a certain configuration of either one or two minutiae and a reference core or delta. Secondly, to expand this tool to include a quality measure for fingerprint images, resulting in the inclusion of poor quality fingerprints from frequency estimation.

The performance of the tool was measured using an existing set of fingerprint and minutia configuration comparisons. The viability of the tool was determined by seeing if the tool meets a minimum required performance, as described below. The effects of requiring a minimum image quality were analyzed by excluding these poor quality fingerprints from the ground truth set used to measure the tool's performance, and comparing the improved tool's performance to the original performance.

The scope of the research was limited to the following:

- An existing minutiae extraction and encoding tool called the Motorola Printrak Biometric Identification Solution 9.1, selected because it has been used for previous research at the Netherlands Forensic Institute.
- An existing fingerprint image database containing core and delta positions, created during previous research at the Netherlands Forensic Institute[3].
- The creation of an algorithm that filters this database based on whether or not a configuration of one or two minutiae and a core or a delta is present in a fingerprint. It will be based on the pixel position and orientation of features, and the orientation of the complete configuration.
- The NFIQ2 tool for automatic determination of fingerprint image quality.

In consultation with fingerprint examiners and researchers, an educated estimate was made of the minimum performance required to be reached for the tool to be viable. A false rejection rate of 20% was deemed low enough to reliably adjust the frequency estimation for the ratio of fingerprints missed by the tool. At this false rejection rate, we determined a false acceptance rate of at least 50% has to be reached, which effectively doubles the amount of fingerprints considered in the complete frequency estimation. We expect that a performance above a FAR of 50% at a FRR of 20% significantly improves the frequency estimation, compared to a completely manual situation.

The rest of this paper describes the design of the tool and its performance measurement method in section 2, the measurement results in section 3, recommended improvements in section 4, and concluding remarks in section 5.

2 Data and methods

2.1 Minutia extraction and encoding

The first step the tool takes is to automatically extract and encode the minutiae in the fingerprints that require filtering. For this research an existing tool was used: Motorola Printrak Biometric Identification Solution 9.1. This is a closed source tool that can retrieve an approximation of minutiae positions and orientations. It does not, however, recognize the difference between minutia types, like line endings and bifurcations. Figure 2 shows a sample of the encoding made by this tool.

The sample image also reveals that the algorithm can make mistakes. Two minutiae at the core are found, one of which does not exist while the other is incorrectly orientated. Additionally, the center stick is not recognized, while in other images it is successfully marked. Different algorithms will have different flaws, and these will influence the performance of the final tool, or require extra techniques to circumvent.

Figure 2: An example fingerprint encoded with the Motorola Printrak Biometric Identification Solution 9.1 tool. The encoding is not perfect, as can be read from the two falsely encoded minutiae near the core, and the missing of the center stick minutia.

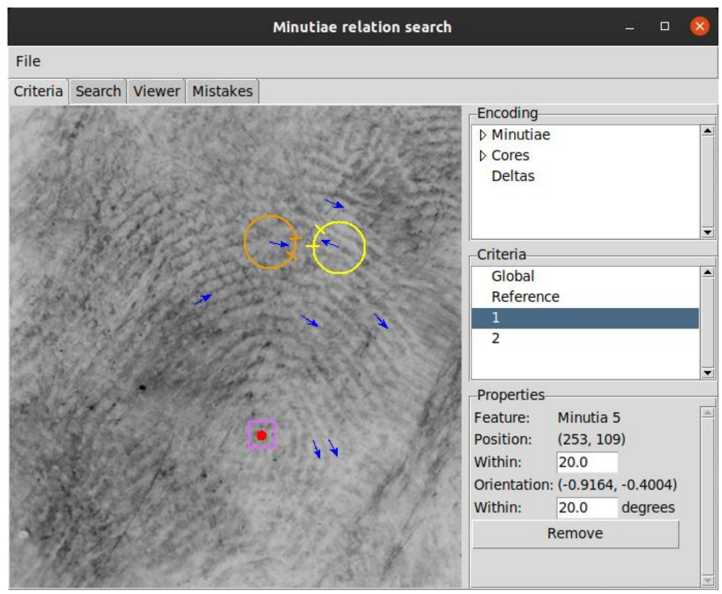

Figure 3: The tool created during this research. The user loads a fingerprint image and a, possibly manually created, encoding of minutiae, cores, and deltas. Minutiae are displayed by arrows, and the core by a red dot. In this case there is no delta. The user chooses a reference point, here displayed by the square surrounding the core. Next, minutiae criteria are configured. Each of the required minutiae have a circle around it, representing the chosen margin for its position. Finally, each circle has two stripes, indicating the margin for the minutia's orientation.

2.2 Filter algorithm

The filter algorithm checks for each fingerprint in the database if the required minutia configuration is present. In this research the number of minutiae in a configuration was limited to one or two, but in principle the algorithm works the same with a greater number of minutiae. The fingermark from which the minutia configuration was derived and all images in the database are required to

- be orientated in the same manner, and
- be 500 pixels per inch.

The algorithm was designed to mimic human examiners by using a core or delta position as reference, and checking if required minutiae are present within a margin at a specified pixel offset. Additionally, the required orientation of each individual minutia can be set, again within margins. Finally, a margin can be set for the rotation of the complete fingermark to compensate for uncertainty in its orientation. Figure 3 shows our developed tool as a visual example. See algorithm 1 for pseudocode. An important distinction between the algorithm and a human examiner is that a person looks at ridge line distance, whereas the algorithm examines vertical and horizontal pixel offset.

2.3 Performance measurement

2.3.1 Reference data

The performance of the tool was measured using multiple minutia configurations. For each configuration a set of approximately one hundred fingerprints to be filtered was compiled. Next, it was manually determined whether or not each of these fingerprints contained their corresponding configuration. The fingerprints are part of the database created during the research performed by A. de Jongh et al.[3], which was extended to contain core and delta positions. The latter was done by two interns who were both trained in a similar manner as the first intern that helped to create the database. In previous casework, two examiners have examined subsets of this database to determine the frequency of occurrence of minutia configurations present in the case's fingermarks. They took into account the position and ridge distance of up to two minutiae relative to a core or delta reference point, as well as the type of each minutia. Because the amount of data was limited, the one minutia and two minutiae cases were considered the same during this research. The subsets were created by filtering the database by the pattern type and core-delta ridge distance of the fingermarks. In total this amounted to approximately 4200 comparisons between minutia configurations and fingerprints, where the fingerprints were previously filtered by the pattern type and core-delta ridge distance of the fingermark from which the configurations originated.

Algorithm 1 The filter algorithm

Require:
database: the set of fingerprints to be filtered
reference_type: the type of the reference feature, which can be either CORE or DELTA.
criteria: the required offsets and orientations of minutiae and their corresponding margins
mark_rotation: the margin for the rotation of the complete fingermark

for each *fingerprint* **in** *database* **do**
    let *minutiae* $\leftarrow$ *minutiae* **in** *fingerprint*
    if *reference_type* is CORE **then**
        let *references* $\leftarrow$ *cores* **in** *fingerprint*
    else
        let *references* $\leftarrow$ *deltas* **in** *fingerprint*
    end if

    if there **exists**
    • a *reference* from *references*,
    • an angle θ_g, $-mark\_rotation \leq \theta_g \leq mark\_rotation$,
    • an *injection* from *criteria* onto *minutiae*
      (An injection here means that every criterium is matched with exactly one minutia, with no more than one criterium per minutia)
    where
        for each *criterium*, *minutia* **in** *injection*
            let $\theta_m \leftarrow$ *orientation* **in** *minutia*
            let $\omega_m \leftarrow$ *position* **in** *minutia*
            let $\omega_r \leftarrow$ *position* **in** *reference*
            let $\theta_c \leftarrow$ *orientation* **in** *criterium*
            let $\omega_c \leftarrow offset$ **in** *criterium*
            let $\delta_\theta \leftarrow$ *orientation_margin* **in** *criterium*
            let $\delta_\omega \leftarrow offset\_margin$ **in** *criterium*

            let $R_g \leftarrow \begin{bmatrix} cos(\theta_g) & sin(\theta_g) \\ sin(\theta_g) & cos(\theta_g) \end{bmatrix}$

            let *orientation_ok* $\leftarrow \theta_c - (\theta_m + \theta_g) \leq \delta_\theta$
            let $offset\_ok \leftarrow |R * (\omega_m - \omega_r) - \omega_c| \leq \delta_\omega$

            orientation_ok **and** $offset\_ok$
    then
        yield *fingerprint*, ACCEPTED
    else
        yield *fingerprint*, REJECTED
    end if
end for

2.3.2 Measurement method

The performance of this tool was measured by its false acceptance rate and its false rejection rate. These are inversely correlated, meaning there will always be a trade-off between the two. A bigger false rejection rate equates to more fingerprints that do contain the minutia configuration being incorrectly filtered as not containing that configuration, and a bigger false acceptance rate means more fingerprints that do not contain the configuration are not filtered. Therefore, the trade-off for this tool is between the amount of work needed to be done by hand, and the portion of fingerprints that contain the configuration that are automatically discarded in error. By varying the input parameters of the tool, its performance in both of these areas changes. In this case the position and orientation margins of the minutiae, as well as the maximum rotation of the fingermark, were varied. For the position margin a single value was used between all minutiae. The same was done for the orientation margin, resulting in three varying parameters. Putting the resulting performance points in a scatter plot and drawing the lower end boundary of this point cloud gives a function similar to a decision error trade-off curve. For a visual example, see figure 4. The same approach has been taken by Q. Tao et al.[4] for a similar problem. By by interpolating between the parameters at the points the curve moves through, optimal parameters are found for each FRR and FAR value.

Following the objectives of this research, the following points of interest on this curve were chosen:

- The false acceptance rate at a false rejection rate of 0.2.
- The false rejection rate at a false acceptance rate of 0.5.
- The equal error rate as a general indication of performance

2.3.3 Determining tool shortcomings

To improve the performance of this and future tools, the mistakes made by the tool were analyzed to determine their cause. Two different errors were considered: false rejects and false accepts. For both cases a random sample of 50 incorrectly classified fingerprints were manually examined, as well as their automatically encoded minutiae. The parameters for the tool were set to reach an equal error rate. First, an overview was created of frequently recurring mistakes, after which the occurrence of each type of mistake was counted. The resulting data shows in which areas the tool does not perform well, ordered by frequency of occurrence in increments 10%. While examining more images would lead to a higher precision, it would be a very labor intensive task. The current precision was deemed sufficient for the set objective.

2.4 Discarding low quality images

The tool was extended to include a quality measure for fingerprint images. Low quality fingerprints are to be excluded during frequency estimation. This is statistically sound as long as no correlation exists between the discarding of a fingerprint and it containing the required minutia configuration.

For determining fingerprint image quality the NIST Fingerprint Image Quality tool 2.0[5] was utilised. This is the second generation of a tool developed by the National Institute of Standards and Technology in cooperation with various international organizations. It comes with an extensive report that explains the used algorithms in depth. The tool can calculate scores based on different factors, such as the clarity of ridges and valleys and the certainty at which line orientation can be determined. An overall image quality score is also generated, optimally combining all the sub-scores.

This last score was used as the image quality score in the minutia filter tool, adding the exclusion from rarity estimation of fingerprints that do not match a quality threshold, while keeping the rest of the algorithm the same. See algorithm 2 for pseudocode of the extended tool. We expected that increasing this threshold would have a positive influence on both the false rejection rate and false acceptance rate of the tool, at the cost of having a smaller database. This research had a limited amount of data available, so to keep performance measurements reliable, the threshold was increased so that approximately only fifty percent of the database would be discarded.

The effects of a minimum image quality threshold on the reliability of the frequency measurement were evaluated by excluding poor quality images from the reference database used to performance measurement. An improved performance then equates to a higher quality frequency measurement.

Algorithm 2 Filter algorithm extended with quality score

Require:
database: the set of fingerprints to be checked for sufficient quality
threshold: the minimum quality score required for a fingerprint image to not be excluded
other: parameters as required by the original algorithm

for each $fingerprint$ **in** *database* **do**
  let *quality* $\leftarrow$ NFIQ2($fingerprint$)
  if *quality* $<$ *threshold* **then**
    yield $fingerprint$, DISCARDED
  else
    $<$*Proceed as with the original algorithm*$>$
  end if
end for

	Base tool	With quality
FAR at 0.2 FRR	0.467	0.435
FRR at 0.5 FAR	0.179	0.160
Equal error rate	0.317	0.302

Table 1: Performance of tool iterations in numbers. 'Base tool' is the tool without the addition of the quality threshold, and 'With quality' is the tool after its addition.

3 Results

3.1 Filter without image quality

3.1.1 Performance numbers

Figure 4 shows the performance of the tool before the addition of the image quality threshold, and the values at the points of interest can be seen in table 1. The minimum requirements set for the tool to be useful are reached, but not by a large margin. We recommend slightly improving its performance before using it for practical applications.

The parameters to reach the optimal performance curve are laid out in table 2. The rotational parameters, both for the completely fingermark as only locally for individual minutiae, vary widely between a FAR of 0.2 and a FRR of 0.5, even though these points lie close to each other. On the other hand, the positional parameter appears to have a clear optimal value at those points.

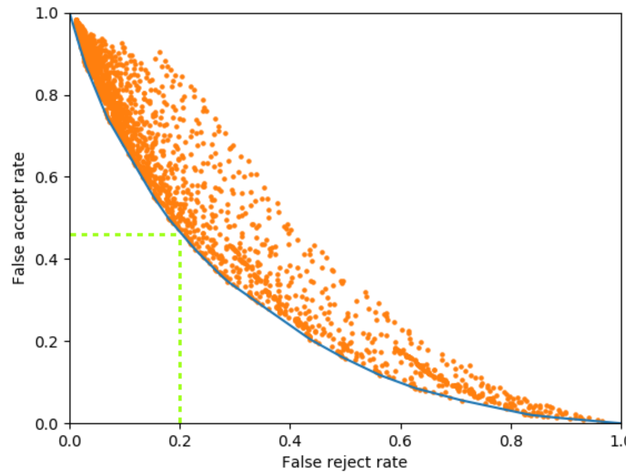

Figure 4: Performance of tool without image quality threshold. The dots form the scatterplot of all found FRR/FAR points and the curve is the lowest boundary of this cloud, representing the DET curve of the tool. The FAR at 0.2 FRR is indicated by the striped square.

Tool variant		Position	Orientation	Mark
No quality	0.2 FRR	0.200	0.626	0.119
	0.5 FAR	0.200	0.839	0.161
	ERR	0.136	0.500	0.150
With quality	0.2 FRR	0.199	0.549	0.101
	0.5 FAR	0.213	0.811	0.143
	ERR	0.135	0.500	0.150

Table 2: Parameters of the tool at the points of interest. The first row shows the tool without quality threshold, and the second row the tool with it added. 'Position' and 'Orientation' are the parameters for their respective margins and 'Mark' is the maximum rotation of the fingermark. 'Position' is measured in pixels, and 'Orientation' and 'Mark' in radians.

3.1.2 Causes of false accepts

Most of the false accepts did not have a single origin of failure, but rather had multiple causes. The percentages given below represent the specific fraction of mistakes that were influenced by each mistake type, and as such do not equate to 100%.

- 40% of false accepts were partially or fully caused by minutiae found by the extraction and encoding tool that do not actually exist. Around these minutiae the image was usually also hard to discern by eye. See figure 5 for an example of this.

- 60% of false accepts were partially or fully caused by minutiae being at the right pixel offset, but not actually being on the correct ridge. For a visual example, see figure 6.

- 50% of false accepts were partially or fully caused by minutiae being of the incorrect type, as is the case in figure 7. The tool does not take into account the type of minutiae, while the examiners that created the reference dataset did, making this a big shortcoming of the algorithm.

3.1.3 Causes of false rejects

As opposed to false accepts almost all false rejects had a single cause of failure. While some overlap exists it was insignificant and as such does not appear here.

- 40% of false rejects were entirely caused by the extraction and encoding tool not finding relevant minutiae. Around these minutiae the image was usually hard to discern by eye as well. See figure 8 for an example.

- 60% of false rejects were entirely caused by the relevant minutiae not falling within the set criteria. This occurs when a minutia is on the correct ridge, but only marginally in the right direction from the reference point, not falling within the positional criteria used by the tool. Figure 9 shows an example of this error.

3.1.4 Additional observations

During analysis of the mistakes made by the tool, two discoveries were made that are thought to influence the tool's performance.

Firstly, the used extraction and encoding algorithm appears to often fail to encode center sticks, line endings that lie on a core, as minutiae. It has to be taken into account that different algorithms have different flaws that can skew the performance measurement results.

Secondly, false accepts often occurred when a criterium minutia lay close to the reference point. This is, for example, the case with a center stick, as there the core is also a minutia. We speculate that there might be a positive correlation between the number of minutiae and the distance to a core or delta, which, as far as we know, has not yet been researched. This would cause more minutiae to be within the positional margin in these cases, increasing the chance of a false accept. Similar other correlations could exist that influence the tool, such as more minutiae occurring around deltas.

3.1.5 Conclusion of tool shortcomings

These results led to the conclusion that there are three major ways to improve the tool.

- Increasing the accuracy of the minutia extraction and encoding. This has a major influence on both false rejection rate and false acceptance rate. During this research we attempted to do this, as described in section 2.4.

- Improving the estimation of ridge line position. Currently this is done by using a pixel position, but this is not optimal. This has a large influence on the false rejection rate and also a minor influence on the false acceptance rate.

- Adding a required type to the minutia criteria, making the tool work more similar to a human examiner. This has only minor influence on the false acceptance rate.

3.2 Performance after adding image quality

The performance of the tool including the image quality threshold can be seen in figure 10, and values at the points of interest are again noted in table 1. Clearly, the performance improves only marginally. We expect this is because only 40% of the errors were caused by the problem attempted to be addressed here, as well as the large overlap with other causes in the case of false accepts.

The optimal parameters at the points of interest can be read from table 2. The positional parameter is similar to the tool without image quality threshold, but the orientational margins are required to be significantly smaller, although surprisingly no differences are visible at the equal error rate.

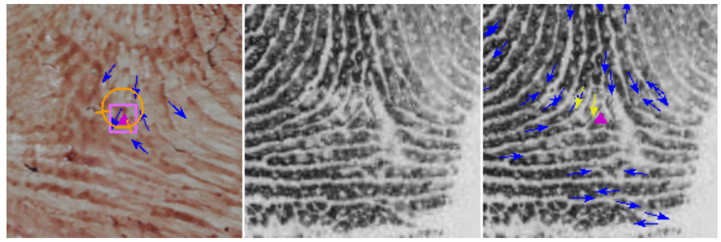

Figure 5: A false accept made by the tool, caused by a minutia found by the extraction and encoding tool that does not actually exist. Left image: The required minutia configuration. Center and right: The compared fingerprint and its encoding made by the tool. The tool found two minutiae that each individually lead to a match, but both of them are mistakes and do not exist.

Figure 6: A false accept made by the tool, caused by a the minutia being on the incorrect ridge. Left image: The required minutia configuration. Center and right: The compared fingerprint and its encoding made by the tool. The criteria were a single bifurcation with a roughly southwards orientation at a ridge distance of 2 north-east from a delta. The tool found a matching minutia, here highlighted in yellow. While by eye it seems to lie just outside the positional criteria, it actually does not because of the rotation of the complete fingermark. The minutia's orientation and type are correct as well, but the human examiner rejected it because the ridge distance from the delta is not correct.

Figure 7: A false accept made by the tool, caused by an incorrect minutia type. Left image: The required minutia configuration. Center and right: The compared fingerprint and its encoding made by the tool. The criteria were a single bifurcation minutia with a roughly southwards orientation at ridge distance zero of a core. The tool found a minutia with the right position and orientation, here highlighted in yellow, and accepted the fingerprint. While it is indeed there, the human examiner rejected it anyway on the grounds that the minutia is a ridge ending instead of a bifurcation.

The shortcomings of the tool after this improvement were again manually analyzed. No significant differences were found, which was initially surprising given the performance improvement, but on second thought is logical given the low precision of the manual analysis, and the fact that the improvement was only marginal. Given this and the DET plot, it can be said that adding this step to the algorithm definitely improves it, but does not prevent all minutia extraction and encoding mistakes. We consider this iteration of the tool to definitely meet the requirements set at the start of this research, but think it is still possible to make large improvements.

4 Future work

The development and testing of this first iteration of the tool raised multiple questions and ideas for improvements. The following section will cover these one by one.

4.1 Good values for FRR and FAR

When using this tool it is important to choose the right FRR and FAR values. Lowering both increases the reliability of the frequency estimation, but because they are inversely correlated, it is always a trade-off between the two. The FRR and FAR values used during this research were an educated estimate, and research is required to find the optimal values for different scenarios.

4.2 Minutia types

A small portion of false accepts is caused by the fact that the tool does not take into the account the type of the minutiae in the configuration. Adding a new criterium to the tool is not ensured to have a positive influence on its performance, as it may decrease its false rejection rate, but could also negatively influence its false acceptance rate. R. Bansal et al. have written a review on existing minutia extraction and encoding algorithms, showing they can include the detection of minutia type[7]. While these algorithms are primarily focused on bifurcations and ridge endings, C. Champod et al. show these types cover at least half of all minutiae appearing in fingerprints[8]. This leads us to suspect that a minutia type detection algorithm already exists that is good enough to improve the performance of this tool.

4.3 Ridge line distance

A significant part of the errors made by the tool are due to it handling minutia positional requirements differently than a human examiner. Replacing the pixel offset criterium with a required ridge-line distance and direction from the reference point to the minutia would make the algorithm more similar to an examiner, increasing the maximum performance that can

Figure 8: A false reject made by the tool, caused by the extraction and encoding tool failing to find a minutia. Left image: The required minutia configuration. Center and right: The compared fingerprint and its encoding made by the tool. The crucial minutia not found here was the ridge ending next to the core.

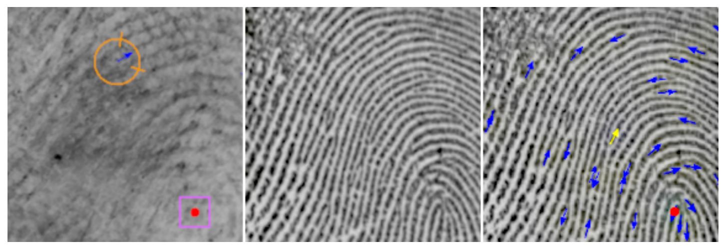

Figure 9: A false reject made by the tool, caused by the positional criteria not catching a minutia that was actually positioned on the correct ridge. Left image: The required minutia configuration. Center and right: The compared fingerprint and its encoding made by the tool. The criteria were a single ridge ending with a north-eastern orientation at a ridge distance of 9 north-west from a core. The human examiner accepted this fingerprint based on the minutia highlighted in yellow, which was not deemed a match by the tool because it fell outside of the positional criteria.

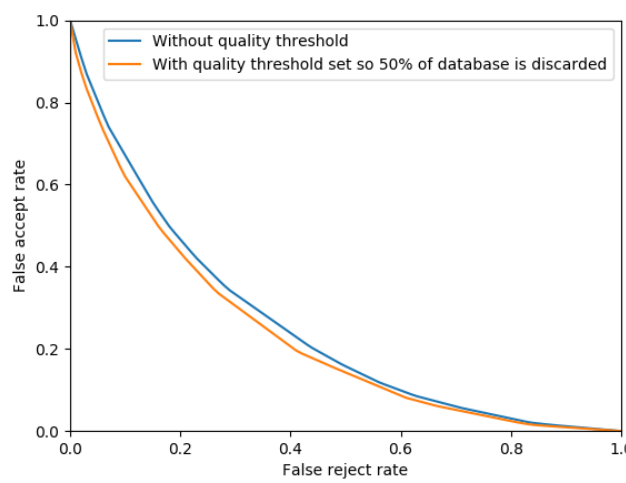

Figure 10: Performance improvement by adding image quality thresholding.

be achieved. As a downside, this introduces ridge distance estimation as an uncertainty factor. However, we think this has great potential, and suspect that even with a crude ridge distance estimation the performance of the tool will improve. It is interesting to note that NFIQ2 calculates a measure that has strong correlation with the ability to count ridges by looking at the frequency of ridges and valleys. This shows that it is also possible to use our existing image quality thresholding technique to improve any existing ridge counting methods.

4.4 Separate parameters depending on the number of minutiae in each configuration

The performance of this tool was measured using configurations of one or two minutiae. These were considered to be the same during this research, even though they could have different effects on the tool's performance. During the process of manually determining the shortcomings of the tool, the suspicion arose that configurations with two minutiae are maybe falsely rejected more often than configurations with a single minutia. This was not further pursued, and additional research could discern if using separate parameters for each of these groups might improve performance.

4.5 Margins specific for each minutia and fingerprint

As stated before, the performance of this tool could be improved by increasing the correlation between the actual ridge position and its approach of this by the use of pixel position. A. V. Maceo. and their referred work show that minutiae positions and orientations are highly influenced by the skin deformation that occurs when the fingertip touches a surface[6]. We wonder if these distortions can be partially compensated for by choosing separate margins for each minutia in every compared fingerprint. For example, a minutia further from its reference point might require a larger margin, or maybe a crude prediction could be made of the positional offset caused by deformity by looking at the thickness of ridge lines.

4.6 More minutiae near cores and deltas

We observed that false accepts often occur when a criterium minutia lies close to the reference point. We speculate that minutiae appear more frequently near a core or delta, which would cause this phenomenon, and further research could show if this is actually the case. If true, a way to solve this problem is to scale the positional offset with the distance between the minutia and its reference.

4.7 Continuous criteria

The current algorithm employs hard discrete criteria, where a minutia must lie exactly within the positional boundaries and have exactly the right orientation. However, sometimes a minutia is perfectly at the right location, but its orientation is just outside the margins. One cause for this could be that the extraction and encoding algorithm is sure that something is occurring at a location on the fingerprint, but is not sure what that something exactly is. An examiner would then weigh the found values and make a decision based on that, letting one criterium compensate for the other. Changing the algorithm to work in a similar way might improve its performance.

4.8 Thresholding by local image quality

The current thresholding method considers the quality of the complete fingerprint image. However, if a small area on the image is of poor quality and the rest is good, the total image will be graded as good. This can have negative effects in two cases. Firstly, when the algorithm criteria overlap with such a poor quality area, the image can be accepted anyway, decreasing the performance of the tool. Secondly, when the criteria do not overlap with the area, the image can be discarded anyway, decreasing the size of the considered database. While the former issue can be solved by increasing the threshold, this causes the required database size to increase rapidly.

An image quality algorithm that separately considers the areas on the fingerprint image relevant to the criteria could decrease the required database size. This would then improve the performance of the tool, as a higher threshold can be used. A crude version of this is already available in NFIQ2, which, for some of its measures, separates the images in smaller pieces and calculates scores for them individually, after which it combines these to give the final score[5].

5 Conclusion

The goal of this research was to develop a tool that performs a pre-selection of fingerprints that might contain a minutia configuration of either one or two minutiae, and one that takes advantage of the possibility to ignore fingerprints for other reasons than the selection requirements. The resulting tool is deemed to be sufficient for both forensic and scientific use, even though its approach does not introduce new non-trivial techniques. We used an existing tool for the extraction and encoding of minutiae in fingerprints to be pre-selected, which was enhanced by only considering images that met a quality standard, which was determined by a second existing tool. We then created a straightforward algorithm that compares these minutiae with a chosen minutia configuration, based on their pixel position and orientation. It is expected that significant performance improvement can be achieved in three areas.

Firstly, the finding of and preventing the falsely finding of minutiae. These have a ripple effect as the rest of the algorithm depends on it. Secondly, the estimation of the ridge-line distance between two points. This affects both the false acceptance rate and false rejection rate, and is a major cause of performance loss. A simple estimation by pixel distance quickly reaches a performance ceiling, and we conclude an algorithm that uses ridge distance is required. Thirdly, including the filtering by minutia type. This is currently completely ignored, but largely influences false acceptance rate.

It is important to have reliable metrics in forensics, and automated tools can be used to improve their quality. After its teething issues are resolved, this tool would be a good addition to the list of available forensic tools.

List of terms

- **False reject** - An item that was rejected by the system, while it should have been accepted according to the ground truth.
- **False accept** - An item that was accepted by a system, while it should have been rejected according to the ground truth.
- **False reject rate (FRR)** - The ratio of false rejects to the total number of items that should be accepted according to the ground truth.
- **False accept rate (FAR)** - The ratio of false accepts to the total number of items that should be rejected according to the ground truth.
- **Decision error trade-off (DET) curve** - A curve that shows the function of FAR by FRR.
- **Equal error rate (EER)** - Value where FRR and FAR are equal in the DET curve.

References

[1] C. Neumann, I. W. Evett, J. Skerett. *Quantifying the weight of evidence from a forensic fingerprint comparison: a new paradigm.* J. Royal Stat. Soc, April 2012, volume 175, issue 2

[2] E. H. Holder, Jr., L. O. Robinson, J. H. Laub. *The fingerprint source book.* National Institute of Justice, 2012

[3] A. de Jongh, A. R. Lubach, S. L. Lie Kwie, I. Alberink. *Measuring the Rarity of Fingerprint Patterns in the Dutch Population Using an Extended Classification Set.* Journal of Forensic Sciences, January 2019, volume 64, issue 1, pages 108-119

[4] Q. Tao, R. Veldhuis. *Threshold-optimized decision-level fusion and its application to biometrics.* Pattern Recognition, 2009, volume 42, number 1

[5] *NIST Fingerprint Image Quality tool 2.0.* National Institute of Standards and Technology, 2018. `https://www.nist.gov/services-resources/software/development-nfiq-20`

[6] A. V. Maceo. *Qualitative Assessment of Skin Deformation: A Pilot Study.* JFI, 2009, volume 59, issue 4, pages 390-440

[7] R. Bansal, P. Sehgal, P. Bedi. *Minutiae Extraction from Fingerprint Images - a Review.* IJCSI, 2011, volume 8, issue 5, number 3

[8] C. Champod, C. Lennard, P. Margot, M. Stoilovic. *Fingerprints and Other Ridge Skin Impressions.* CRC Press LLC, 2016