

# Evolutionary Optimization of JPEG Quantization Tables for Compressing Iris Polar Images in Iris Recognition Systems

Mario Konrad, Herbert Stögner  
School of Communication Engineering for IT,  
Carinthia Tech Institute, Austria  
mario.konrad@edu.fh-kaernten.ac.at

Andreas Uhl  
Department of Computer Sciences,  
University of Salzburg, Austria  
uhl@cosy.sbg.ac.at

## Abstract

*Recognition performance in iris biometrics strongly depends on the image quality. The appliance of compression algorithms to iris images raises the question whether it is possible to adapt those algorithms for biometrical purposes. In this work, we propose customized JPEG quantization matrices for compressing iris polar images to positively impact the recognition performance. We build on previous research and apply a genetic algorithm to obtain specialized matrices for destined compression ratios. The proposed tables are able to clearly outperform JPEG's standard quantization matrix. Moreover, some matrices also provide superior results in terms of ROC characteristics as compared to the reference scenario using uncompressed images. This leads to clearly lower error rates while also significantly reducing the necessary amount of data storage and transmission.*

## 1. Introduction

The rising adoption and growing popularity of biometric systems creates the need for an efficient data storing process. The compression of these data is getting more and more important due to the fact that different applications may have to process a vast amount of sensor data. Furthermore, a long term storage is often needed which creates the desire for data compression. The most relevant standard for biometrical image compression is the ISO/IEC 19794-6 standard concerning Biometric Data Interchange Formats. The lossy storage of iris images in the JPEG and JPEG2000 formats is described in this standard. In literature, the performance of compression formats is analysed under default settings. Hence, this paper will focus on the effects of customizing the JPEG standard for iris recognition performance enhancement. As lately investigated in [2], Ives presents first results and techniques on image compression and the resulting change of iris recognition performance. Furthermore, in [1] and [5] it is stated that JPEG2000 has a better performance than JPEG when compressing iris images and especially for low bit rates. Otherwise, JPEG requires less computational demand and is competitive to JPEG2000 at higher image qualities. Since low power devices are often part of the processing chain in

biometric systems, adopting JPEG2000 is not a good solution in such circumstances. We have already utilized customized quantization tables for compressing rectangular images in [4] and we have demonstrated the potential for the enhancement of recognition results. In [3] our focus is on heuristically designed tables to show that an improvement of the performance of JPEG's standard matrix is possible where we have utilized polar images. In addition, we have tested a genetic algorithm to examine its ability to improve recognition performance.

In this work, we also apply JPEG compression to polar images but we do not rely on manual table design. By applying a genetic algorithm, we are able to successively create various quantization matrices for different compression ratios. Since we had no previous knowledge about an optimal parameter set for a genetic algorithm for this application, we investigate the impact of various algorithm settings on the resulting performance. In contrast to former research [3], we use randomly created tables as initial population to avoid any preliminary influence on the optimization process. In series, we are capable of reducing the FAR (False Acceptance Rate) as well as the FRR (False Rejection Rate) compared to the error rates produced by images which are compressed with JPEG's standard matrix. Even the performance provided by the matching of uncompressed pictures can be outperformed in some cases. Beside these results, we also present the analysis of various parameters of the genetic algorithm as well as their impact on the optimization performance.

Section 2 describes the function principle of the genetic algorithm and illustrates how it was utilized for our research. Further, in Section 3, we describe the experimental study in detail. First, the development of optimal parameters for the genetic algorithm for the following optimization stage is illustrated. Secondly, several examples of optimized tables are shown. Finally, the achieved results of the optimization process are depicted. Section 4 concludes the paper and gives further perspectives for upcoming research.

## 2. Evolutionary Optimization of JPEG Quantization Tables

With the GA-Toolbox (Genetic Algorithm Toolbox), Matlab® provides a powerful instrument to optimize

certain problems. Its functional principle relies on evolutionary processes. Like in nature, where the strongest, healthiest and smartest individual is likely the reproducing one, the algorithm selects the better individual out of one generation. Consequently, the algorithm needs a criterion for the evaluation of each individual. Therefore, a fitness function is responsible for judging each member of one population and determines the chances for reproduction. One individual consists of a certain amount of genes. If we consider the minimization of a function, the genes would be the variables of the function (e.g.  $X_1, X_2$ ), and one occurrence of these variables (e.g.  $X_1=2, X_2=3$ ) would form one particular individual. In summary, the better the fitness value is, the higher the probability to bequeath genes and produce offspring. After assessing all population members, a new generation is born under predetermined conditions. The best individuals pass their genes on to the next generation. More precisely, the inheritance process has two alternatives. First, the crossover, which is the process of mixing genes of the most successful members of one generation to form a new individual. Second, successful genes are subject of mutation and randomly form a new individual. Furthermore, the fittest individuals are able to produce clones or, in other words, survive as long as no better individual is found. Therefore, after each generation, the mean fitness of the whole population should converge towards the optimal solution of the given problem and produce better performing individuals.

In the context of our objective, several quantization matrices, being the individuals, form one population whereas the quantization matrix items are the genes. Accordingly, one individual of such a quantization table population consists of 64 independent genes which determine the fitness or, in other words, the recognition performance.

### 3. Experiments

We apply a 1-D implementation of the Daugman iris recognition algorithm, provided by Libor Masek<sup>1</sup>. The utilized polar images are extracted from the picture set of the CASIA V1.0 database (Chinese Academy of Sciences - Institute of Automation Version 1.0, short: CASIA<sup>2</sup> V1.0). We consider two different cases for image matching. First, uncompressed images are compared with compressed images, considering the case that the captured image is compressed and get matched with an uncompressed image in the database. Second, we tackle the scenario that both images are compressed, which implies that the database image has to be compressed before the matching process. Although this additional conversion requires more computational effort for a biometric system, it is also stated in [2] that comparing two compressed images should positively impact the recognition performance. For each

application of the genetic algorithm we consider 70 polar images from 10 different persons, resulting in 4,410 impostor and 420 legitimates hamming distance values. To finally evaluate the optimized matrices (mean hamming distance and error rate analysis), we have applied 348 polar images from 50 various persons resulting in 118,680 impostor and 2,076 legitimate hamming distance values for the scenario that one image is compressed. Due to the symmetry of the comparison, these values are halved in case of two compressed images. The images of the optimization process are not a subset of the image series for the evaluation.

#### 3.1. Discovery of the optimal settings

We intensively analyse the effect of the various genetic algorithm parameters to improve the convergence speed towards an optimal solution. Among the multitude of genetic algorithm's settings, only the most relevant ones, which have been changed within the test phase, will be listed and described. Furthermore, the plotted diagrams in this section are for illustration and not necessarily the basis for our decisions in selecting appropriate algorithm parameters. Table 1 illustrates these parameters and settings and their used values within the test stage. We keep most of the other settings according to the algorithm's implementation.

Parameter	Values
Population size	10, 20, 30
Elite count	1, 2, 4
Crossover fraction	40%, 60%, 80%
Mutation interval	32, 64, 128

**Table 1: Investigated GA parameters and settings during the test stage**

The used fitness functions for assessing the different individuals focuses on the mean error rates of a certain utilized quantization table.

$$Fitness = mean(FAR) + mean(FRR) \quad (1)$$

The FAR and FRR values are the error rates of the particular picture set, generated by one specific quantization table. To gather these values, all different pictures are compared with each other to obtain the particular hamming distances. This results in two vectors of legitimate and impostor comparison values. A following threshold sweep provides certain FAR and FRR results for each step. The mean values of the resulting FAR and FRR vectors have to be minimized during the upcoming optimization process. The algorithm ends when the mean fitness value of the population remains unchanged for 10 generations (after at least 50 processed generations).

The size of the whole population varies between 10, 20 and 30. This impacts the performance of the optimization as well as the computational demand. Hence, there has to be a trade-off between time consumption and effectiveness of the algorithm. Concretely, 10 individuals after 60 generations and 20 individuals after 30 generations both yield a fitness

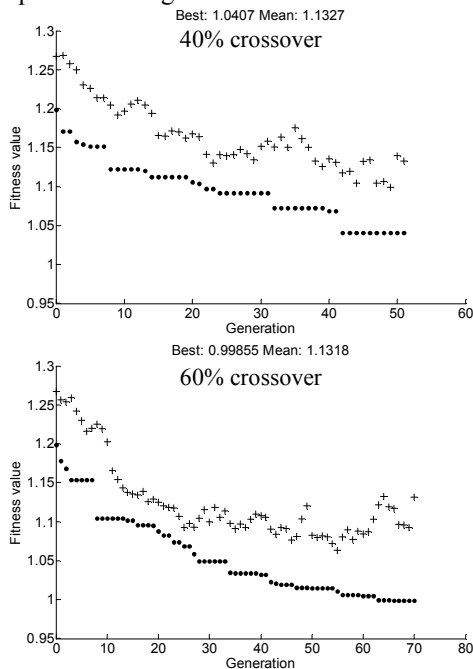
<sup>1</sup> <http://www.csse.uwa.edu.au/~pk/studentprojects/>

<sup>2</sup> <http://www.sinobiometrics.com>

value of 1.05 whereas 30 individuals provide a fitness of 1.08 after 20 generations. Therefore, we have decided to utilize the following optimization stage with a medium population size of 20 individuals.

The number of population members called elite is the amount of best-fitness individuals which are able to bequeath their genes and survive the alternation of generations. We discover better performance with a higher number of elite members just in case of a small population size. For a larger number of individuals, the number of elite members does not have to be explicitly high whereas two elite members provide in nearly every test a better performance than just one. Hence, we apply two elite members in each generation.

The crossover fraction specifies the percentage of individuals which genes are created through a crossover mixture of two parent individuals. We identify the best results with a 60% crossover fraction during our tests. A higher number (80%) lessens the individuals available for mutation and the amount of elite members. A small number (40%) of crossover individuals are just well performing in small populations, obviously as a consequence of a high number of elite members.

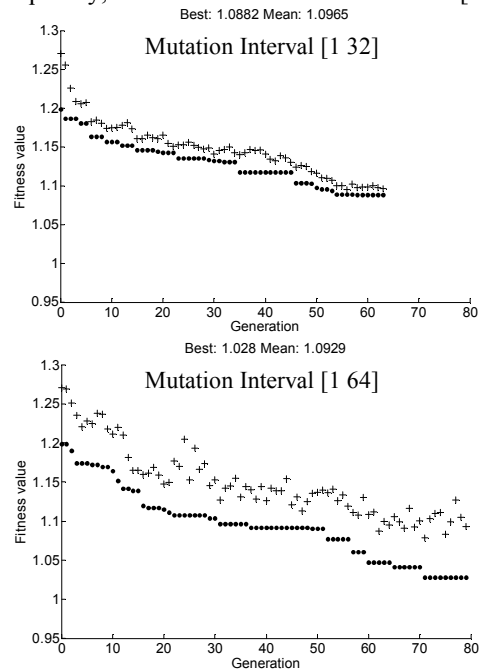


**Figure 1: Comparison between 40% and 60% crossover in a 20-individual population. The dots mark each of the best individuals whereas the pluses depict the mean fitness of the population.**

Figure 1 illustrates the comparison between the well performing 60% crossover and a too low 40% crossover count in a 20-individual population. There are fewer improvement steps for the best individual throughout the generation change, although the population's mean fitness is finally the same. The best performing individual is clearly found in the 60% crossover population. Hence, we utilized a crossover rate of 60% for our analysis.

It is not possible to directly define the mutation rate within a population. The amount of mutated individuals

is determined by the number of elite members and the percentage of crossover-individuals. Concretely, in a population of 10 individuals with 2 elite members and 60% crossover there would be 2 members created by mutation. Whenever an individual is created by mutation, it inherits randomly changed genes from elite members of the previous generation. The amount of allowed gene mutation is defined by the mutation interval. That means that a gene with the value of e.g., 96 and a mutation interval of [1 32] creates a random new gene between 64 and 128. This process is continued for all 64 genes of one individual. During the test stage, we discover that a small mutation interval negatively impacts the performance of small populations. With increasing population sizes the impact of the mutation interval disappears. Figure 2 depicts the difference between a small and a medium sized mutation interval in a population of the size 10, 4 elite members, 40% crossover and 2 mutating individuals. We discover fewer and smaller steps during the optimization, and therefore the process stops earlier and with higher mean and best fitness values. Consequently, we define a mutation interval of [1 64].



**Figure 2: Comparison between small and medium sized mutation intervals. There are fewer and smaller steps during the optimization process.**

The initial population of each following optimization process consists of 20 random tables, which means that each of the 64 table items is randomly chosen between 1 and 255.

### 3.2. Resulting matrices

Applying the genetic algorithm creates several quantization matrices and almost every matrix shows at least the performance of JPEG's standard table. Concretely, 15 tables for compression rates between 2 and 16 are produced or each of the two scenarios. For

two compressed images, 7 tables are able to outperform the uncompressed case and another 6 matrices beat the performance of the standard matrix. For one compressed image, 6 matrices are able to reach the uncompressed case and 8 tables are able to outperform the standard matrix. Overall, it can be stated that the comparison between two compressed images leads to better performance than comparing uncompressed and compressed images. This is not necessarily obvious because in this case the reduction of information yields lower error rates. Interestingly, the performance of some quantization matrices is quite similar while the tables look very different, as illustrated in Figure 3.

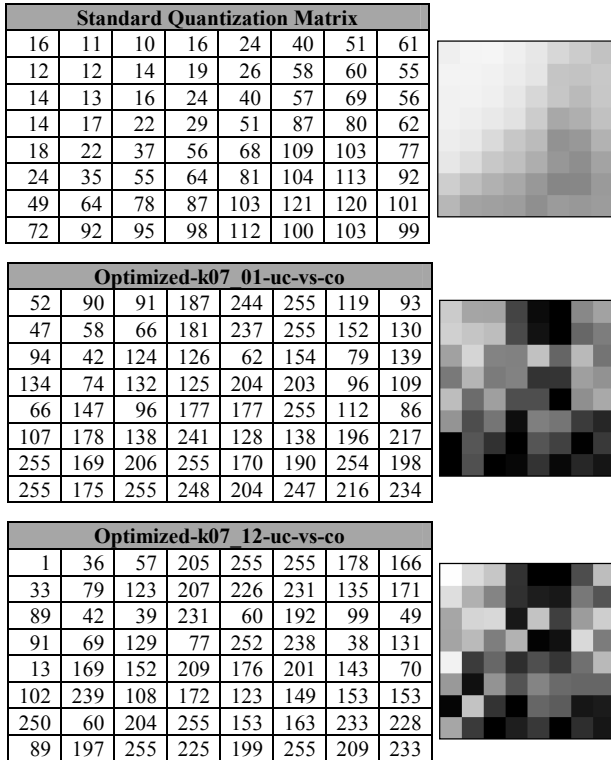


Figure 3: Different quantization matrices which provide nearly the same error rates.

First, the JPEG's standard matrix is shown. The second and third matrices are the result of optimization processes for compression rate 7. The upper matrix is the best performing table of this optimization process whereas the lower matrix is the result of a gene crossover or mutation within the same population. Both are designed for the scenario with one compressed image. The pictures on the right side of each table are for a better readability whereas the matrix items are represented by greyscale areas. Although the values for low frequencies in the upper left corner look rather similar, the whole pattern is very different.

Figure 4 illustrates the error rates of the optimized quantization tables compared to the performance of JPEG's standard table as well as the uncompressed reference case. The diagram shows the two optimized tables of the same optimization process for compression rate 7, short QTK07 (see also Figure 3). The legend description '10p' means that polar images of 10 persons are the basis of the optimization process and 'uc\_vs\_co'

points to the scenario of only one compressed image. Interestingly, the error rates are very similar as shown in Figure 4. The diagram depicts the FAR and the FRR values for a variable decision threshold. We can observe similar error rates between the standard and the two optimized tables although the quantization matrices appear very different.

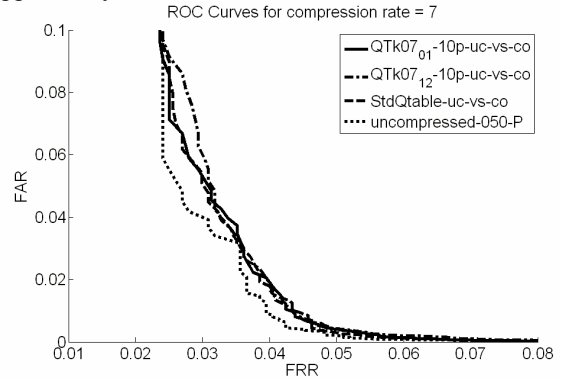


Figure 4: ROC performance of optimized tables compared to JPEG's standard matrix and the uncompressed case.

We have optimized matrices for the usage of polar images and compression ratios reaching from 2 to 16 for both cases, one and two compressed images. Due to the high amount of optimized tables, we are not able to present each matrix in detail. Nevertheless, some examples of optimized matrices are shown in the following section. First of all, in Figure 5, the optimized tables for compression ratios 5 and 12 and the scenario of one compressed image are illustrated.

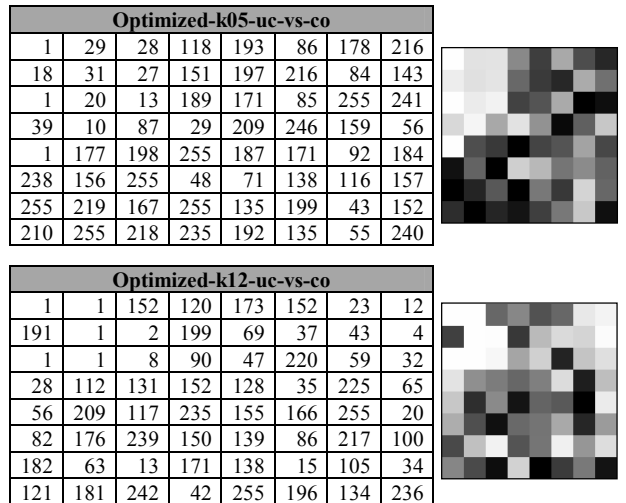


Figure 5: Optimized matrices for compression rate 5 and 12 for the scenario of one compressed image.

Next, in Figure 6, two tables of the scenario of two compressed images for the same compression ratios (5 and 12) are visualized. Although these four matrices are all computed out of tables with random items, a basic trend is clearly detectable. There are always small values for low frequencies whilst some higher frequency components are also preserved. If we compare the matrices for the two different scenarios, we will also notice some similarities in upper left corner whereas the

rest of the table is highly variable, although they are optimized for the same compression rate. Finally, in Figure 7, two of the best performing tables developed in [3] are also shown. Compared to the previously designed tables in [3] it is evident that the new ones partially preserve the higher frequencies and that they have a much more random appearance. Qtable22 was designed heuristically and Qtable\_optk5\_old was also genetically evolved but with heuristically designed tables as initial population and without an optimized parameter set.

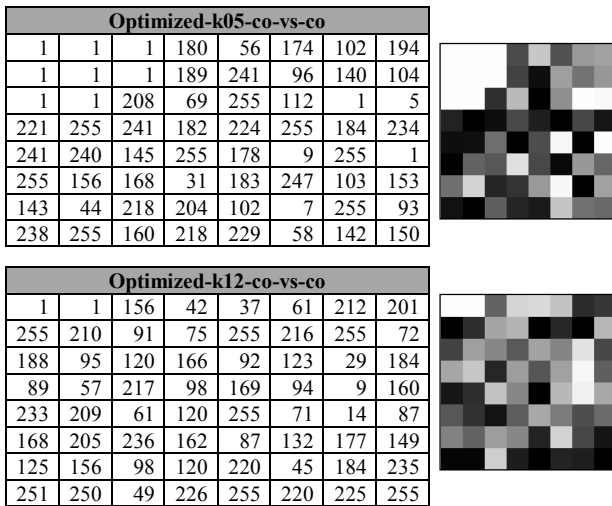


Figure 6: Optimized matrices for compression rate 5 and 12 for the scenario of two compressed images.

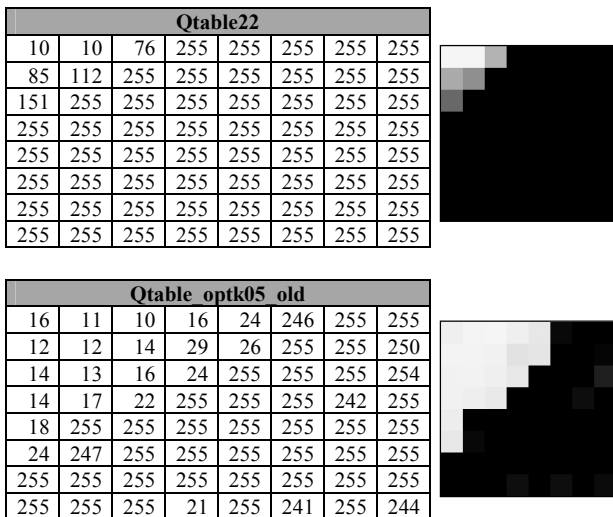


Figure 7: Previously designed tables (Qtable 22 was heuristically designed, Qtable\_optk05\_old was already optimized with the GA) [3].

### 3.3. Achieved performance

The first diagram of Figure 8 shows the error rates of Qtk05\_uc\_vs\_co (see Figure 5) compared to the standard JPEG matrix and the uncompressed reference. We can discover that the optimized matrix shows slightly better results compared to the standard table (between 0.03 and 0.045 FRR). The second diagram illustrates the performance of Qtk12\_uc\_vs\_co (see

Figure 5) and Qtable22 (see Figure 7) compared to the reference cases. The optimized table provides superior results compared to the standard matrix. Although the performance of uncompressed images can not be reached (except around one threshold), Qtable22 is clearly inferior at low FRR values.

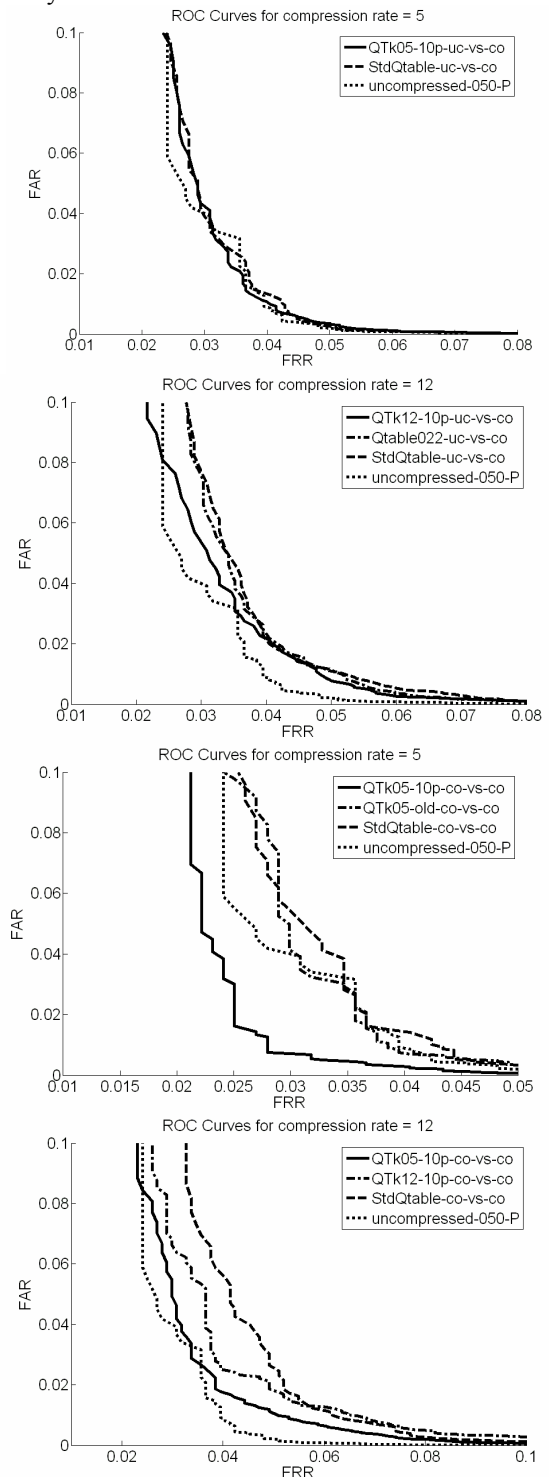
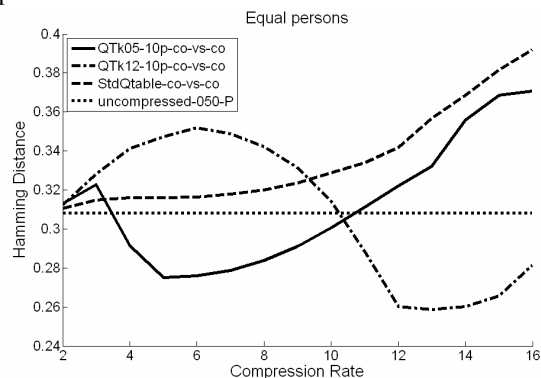


Figure 8: Performance of optimized tables (Qtk05, Qtk12) compared to JPEG's standard, previously designed tables (Qtable22) and the uncompressed reference.

Furthermore, the ROC characteristics for the scenario of two compressed images are illustrated in the

lower two diagrams of Figure 8. The third diagram shows two tables which are optimized for compression rate 5. The newly evolved table (see Figure 6) clearly outperforms the standard table as well as the uncompressed case. The ‘older’ optimized table [3] (see Figure 7) is able to achieve the results of the uncompressed reference for higher FRR values and reaches the performance of the standard table for lower FRR values. It can be clearly seen that the optimization of the Genetic Algorithm parameter set provides better performing tables. Moreover, the fourth diagram illustrates ROC characteristics for compression ratio 12. Beside the matrix QTk12 (see Figure 6) which clearly beats the standard table in terms of error rates, the previously illustrated table QTk05 (see Figure 6) also shows very good results at higher compression ratios, although the mean hamming distance values do not really indicate that trend (see Figure 9). That is one example for an optimized table which performs very well over a larger range of compression ratios whereas other tables are just perform well around their targeted compression.

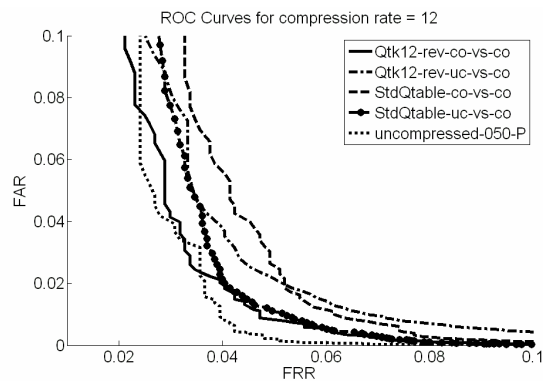
Figure 9 shows the distribution of the mean hamming distances of the optimized tables (QTk05 and QTk12) for the scenario of two compressed images. A clear minimum of the mean hamming distance is detectable at the target compression rate of the optimized tables.



**Figure 9: Hamming distance distribution of optimized tables (QTk05, QTk12) compared to JPEG's standard and the uncompressed reference.**

Another interesting fact is that the scenario of two compressed images regularly yields better matrices than the scenario of only one compressed image. In Figure 10, we use tables which are optimized for the scenario of one compressed image and calculate their performance for the scenario of two compressed images, and vice versa.

Although the table labelled ‘Qtk12\_rev\_co\_vs\_co’ is originally optimized for one compressed image, it performs very well for the opposite scenario and JPEG's standard is clearly beaten. ‘Qtk12\_rev\_uc\_vs\_co’, on the other hand, is not able to reach the performance of the standard matrix (except for low FRR rates). Also in this example the ‘misused’ tables show better performance in the scenario of 2 compressed images.



**Figure 10: Performance of optimized tables in the opposite target scenario compared to JPEG's standard and uncompressed images.**

#### 4. Conclusion

In terms of average hamming distances and the ROC behaviour the identified tables perform significantly better as compared to the default table. Even the performance of uncompressed images can be beaten in some cases. The optimization procedure itself induces high computational costs due to the expensive evaluation of the fitness function for each individual in all generations. However, the optimization process has to be applied only once for a certain image database and would improve recognition results and lower the amount of necessary data. In upcoming research, we will extend our analysis in order to optimize matrices for rectangular iris images out of different databases.

#### 5. References

- [1] J. Daugman, C. Downing, “Effect of severe image compression on iris recognition performance”, *IEEE Transactions on Information Forensics and Security*, Volume 3, March 2008
- [2] R. Ives, R. Broussard, L. Kennell, D. Soldan, “Effects of image compression on iris recognition system performance”, *Journal of Electronic Imaging*, 17:011015, doi:10.1117/1.2891313, 2008
- [3] M. Konrad, H. Stögner, A. Uhl, “Improving Iris Recognition Accuracy by Custom Design of JPEG Quantization Matrices for Compressing Polar Iris”, *Proceedings of the 3rd International IEEE/IAPR Conference on Biometrics 2009 (ICB'09)*, Springer Lecture Notes on Computer Science 5558, pp. 1098 - 1108, 2009.
- [4] G. Kostmayer, H. Stögner, A. Uhl, “Custom JPEG Quantization for Improved Iris Recognition Accuracy”, *Proceedings of the 24th IFIP International Information Security Conference 2009 (IFIP SEC'09)*, D. Gritzalis and J. Lopez, Springer Verlag, IFIP Series 297, pp. 76-86, 2009
- [5] S. Matschitsch, M. Tschinder, A. Uhl, “Comparison of compression algorithms' impact on iris recognition accuracy”, *Proceedings of the 2nd International Conference on Biometrics 2007 (ICB'07)*, Volume 4642, August 2007, pp 232–241