

CONTENTS

	Page
INTRODUCTION.....	1
CHAPTER 1 INTELLIGENT WATERMARKING OF GRAYSCALE IMAGES.....	9
1.1 Watermarking Metrics	9
1.1.1 Watermark Quality	10
1.1.2 Watermark Robustness	11
1.1.3 Watermark Capacity	13
1.2 Embedding and Extracting Watermarks	14
1.3 Embedding Parameters Optimization	16
1.3.1 Single Objective Optimization Problem (SOOP)	18
1.3.2 Multi-Objective Optimization Problem (MOOP)	20
1.3.3 Dynamic Optimization	23
1.4 Large-Scale Global Optimization Problems	26
1.5 Texture Features and Metrics of Grayscale Images	31
1.5.1 Extract Grayscale Texture Features	31
1.5.2 Grayscale Texture Metrics	32
1.6 Bio-watermarking.....	35
1.7 Discussion.....	36
CHAPTER 2 RAPID BLOCKWISE MULTI-RESOLUTION CLUSTERING (BMRC)	39
2.1 System Overview	39
2.2 Training	45
2.2.1 Full Optimization	46
2.2.2 Find Set of Clusterings SC	47
2.2.3 Find Most Frequent Bands for Candidate Clusters	49
2.3 Generalization.....	53
2.3.1 Extract Texture Features.....	54
2.3.2 Find Image Blocks Clusters for Different k Values	54
2.3.3 Calculate Embedding Capacity for Different Blocks C_i	54
2.3.4 Recall $MF B_k$ for Clusters Found and Calculate Fitness	56
2.3.5 Rank Solutions According to Fitness to Find Value of k	57
2.4 Experimental Methodology	58
2.5 Experimental Results	61
2.6 Conclusions and Future Directions	71
2.7 Discussion.....	73
CHAPTER 3 BLOCKWISE COEVOLUTIONARY GENETIC ALGORITHM (BCGA)	75
3.1 System Overview	76
3.2 Perceptual Texture Masks	79
3.3 Blockwise Coevolutionary Genetic Algorithm (BCGA).....	80

3.3.1	Initialization.....	83
3.3.2	Elitism.....	84
3.3.3	Selection for Mating, Crossover and Mutation.....	85
3.4	Experimental Methodology	88
3.5	Results and Discussion	90
3.6	Conclusions and Future Work.....	96
3.7	Discussion.....	99
CONCLUSION.....		101
ANNEX I	PERCEPTUAL TEXTURE MASKS EVALUATION.....	103
ANNEX II	BIO-WATERMARKING SYSTEMS USING OFFLINE SIGNATURE FEAT- TURES.....	113
REFERENCES		127

LIST OF TABLES

		Page
Table 1.1	Survey for different EC-based methods for watermarking grayscale images.	22
Table 1.2	Survey for different change detection mechanisms in dynamic environments proposed in literature.	26
Table 1.3	Perceptual masks literature survey.....	33
Table 2.1	Computational complexity of the proposed system using even and uneven scheme for a stream of 40 face images using training set size $N = 1$, measured in total CPU time for the stream.	61
Table 2.2	Computational complexity reduction of the proposed BMRC system compared to baseline system and MOGA (Deb, 2001) for a stream of 40 face images using training set size $N = 1$, measured in number of fitness evaluations for the stream.	62
Table 2.3	Mean fitness for validation set of face images for MOGA (Deb, 2001) and baseline system compared to proposed system with even and uneven scheme using watermark of length $WL = 48.8k$	63
Table 2.4	Proposed system performance using uneven scheme with different watermark length.	65
Table 2.5	Impact of β on the total memory size of BCM Mem_{BCM} in k-bits, where $Rep_{float} = 32$ -bits as per IEEE 754 and $Rep_{int} = 6$ -bits to represent index of embedding coefficients $a \in [0, 1, \dots, 63]$	67
Table 2.6	Impact of robustness scores threshold α on the proposed system performance for watermarks of different lengths WL for stream of 40 face images.	68
Table 2.7	Impact of using GPU to perform DCT transform, and C for watermark embedding/extraction on fitness evaluation complexity $T_{QF} + T_{RF}$, accuracy for 40 face images, and optimization iteration complexity $T_{Iteration}$ in traditional approaches (Shieh <i>et al.</i> , 2004).....	69
Table 2.8	Proposed system performance with different training set size N and embedding capacity C	70
Table 2.9	Proposed system performance with two different testing sets of size M equals to 40 and 198 face images respectively using training set size N equals to 3 face images.....	71

Table 3.1	Optimization of embedding 1 bit-per-block in reduced size of face image 0001 from PUT database (Kasinski <i>et al.</i> , 2008) using baseline method proposed by Shieh <i>et. al</i> (Shieh <i>et al.</i> , 2004), PBIL and BCGA with $gf = QF + 20(RF_1 + RF_2 + RF_3)$	92
Table 3.2	BCGA quality of solutions compared to traditional GA (Shieh <i>et al.</i> , 2004) using weighted sum aggregation for tuning set of 40 facial images.	93
Table 3.3	Chebyshev aggregation using different quality fitness weights γ for embedding optimization using BCGA with embedding capacity equals to 4 bits-per-block for tuning set of 40 face images.	94
Table 3.4	Impact of the number of elite individuals EL on BCGA performance using Chebyshev aggregation for gf as shown in equation 3.1.	95
Table 3.5	Impact of the probability of crossover $P_{crossover}$ on BCGA performance using Chebyshev aggregation for gf as shown in equation 3.1.	95
Table 3.6	Impact of the probability of mutation $P_{mutation}$ on BCGA performance using Chebyshev aggregation for gf as shown in equation 3.1.	96
Table 3.7	Impact of the perceptual texture mask on BCGA performance using Chebyshev aggregation for gf as shown in equation 3.1.	97

LIST OF FIGURES

		Page
Figure 1.1	Classification for watermark attacks (Voloshynovskiy <i>et al.</i> , 2001).....	12
Figure 1.2	Optimization candidate solutions representation in traditional methods (Shieh <i>et al.</i> , 2004) with embedding capacity equals to 4 bits per block.	17
Figure 1.3	Data flow diagram depicting the single objective formulation for watermark embedding optimization to find optimal embedding bands EB_{X_c} for cover image X_c	19
Figure 1.4	Data flow diagram depicting the multi-objective formulation for watermark embedding optimization to find optimal embedding bands EB_{X_c} for cover image X_c	21
Figure 1.5	Pareto Optimal Front (POF) and non-dominated solutions for multi-population optimization (Li and Engelbrecht, 2007).	22
Figure 1.6	Intelligent watermarking for stream of images as single dynamic optimization problem (Rabil <i>et al.</i> , 2011b),(Vellasques <i>et al.</i> , 2011).	23
Figure 1.7	Different types of Dynamic MultiObjective Optimization Problems (DMOOP) (Farina <i>et al.</i> , 2004).....	25
Figure 1.8	Coevolutionary model of three species shown from the perspective of each in turn (Potter and DeJong, 2000).	29
Figure 2.1	General overview of BMRC architecture and processing steps for training and generalization phases. BCM is organized based on clustering resolution k ranging from K_{min} to K_{max}	42
Figure 2.2	Selecting solution among Pareto front for training set face images TI_d from PUT database (Kasinski <i>et al.</i> , 2008) based on application domain objectives priorities, where embedding capacity is 8 bits per block.....	47
Figure 2.3	Zoning method to select the most significant DCT coefficients to extract texture features tf_i for block b_i (Yu <i>et al.</i> , 2010), $tf_i = \{ac_a\}$, where $a = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 33, 34, 35, 36, 37, 41, 42, 43, 48\}$	48

Figure 2.4 Robustness scores $rs_{(k,j)}$ calculation example for $k = 4$, where $SC_4 = \{p_{(4,1)}, p_{(4,2)}, p_{(4,3)}, p_{(4,1)}\}$, $NB_4 = \{1.6k, 3.4k, 40k, 5k\}$, $RS_4 = \{0.99, 0.96, 0.65, 0.90\}$, and $CL_4 = \{1, 2, 4\}$ 52

Figure 2.5 Embedding blocks belonging to CL_k of clustering resolutions $k=11,16$ for different WL on face image from PUT database (Kasinski *et al.*, 2008). 52

Figure 2.6 Classifying block b_i for different k values in texture feature space. 55

Figure 2.7 Classifying block b_i in texture feature space and recall $mf b_{(k,j)}$ associated with prototypes $p_{(k,j)}$ for $k = 3$, where $SC_3 = \{p_{(3,1)}, p_{(3,2)}, p_{(3,3)}\}$, and $MF B_3 = \{mf b_{(3,1)}, mf b_{(3,2)}, mf b_{(3,3)}\}$ 57

Figure 2.8 BancTec binary logo used as watermark to be embedded in face images. ... 58

Figure 2.9 Fitness comparison of the proposed system with even and uneven scheme compared to baseline system. 62

Figure 2.10 The impact of variations of the number of blocks belonging to similarly textured clusters on the proposed BMRC performance using uneven scheme. 64

Figure 2.11 Extracted watermarks of different length WL using the proposed system with uneven scheme. 66

Figure 2.12 Proposed system performance with uneven scheme for different watermark lengths WL 67

Figure 2.13 Impact of increasing training set size N on the proposed system performance. 71

Figure 3.1 Block diagram for the proposed watermarking system for high-resolution face images. 77

Figure 3.2 Chromosome representation of the proposed BCGA. 78

Figure 3.3 Data flow diagram for the BCGA to find optimal embedding bands EB_{X_c} for image X_c 82

Figure 3.4 Proposed elitism on block level with $EL = 4$ using block watermarking metric bwm_i 85

Figure 3.5 Proposed crossover on block level with $cp = 2$, embedding capacity $C = 4$, and 6-bit representation for embedding bands. 87

Figure 3.6	Sample face images from first pose of the PUT database (Kasinski <i>et al.</i> , 2008), where grayscale images have a resolution 2048×1536 pixels. 88
Figure 3.7	Evolution of baseline system using GA with two crossover methods, and PBIL compared to the proposed BCGA using embedding capacity of 1 bits-per-block for face image 0001. 92
Figure 3.8	Evolution of reduced size face image 0001 using baseline system using GA proposed by Shieh <i>et al.</i> (Shieh <i>et al.</i> , 2004) with embedding capacity 1 bit-per-block using different sized watermarks. 93
Figure 3.9	Impact of user-defined parameters of BCGA on fitness evolution using Chebyshev aggregation for gf as shown in equation 3.1. Log scale is used for gf due to the high similarity of fitness evolution for EL and $P_{crossover}$ 96
Figure 3.10	Impact of $P_{mutation}$ on BCGA evolution using Chebyshev aggregation for gf as shown in equation 3.1. 97
Figure 3.11	Impact of using different grayscale texture metrics as perceptual masks on BCGA evolution using Chebyshev aggregation for gf as shown in equation 3.1. 98

LIST OF ABBREVIATIONS

AR	Accuracy Ratio
AUR	Area Under Curve
BCGA	Blockwise Coevolutionary Genetic Algorithm
BCM	Block Cluster Memory
BCR	Bit Correct Rate
BMRC	Blockwise Multi-Resolution Clustering
BWM	Block Watermarking Metric
CBIR	Content Based Image Retrieval
CCES	Cooperative Coevolution Evolution Strategy
CCGA	Cooperative Coevolution Genetic Algorithm
CCSO	Cooperative Coevolution Swarm Optimization
CGA	Compact Genetic Algorithm
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DMT	Discrete Multiwavelet Transform
DWT	Discrete Wavelet Transform
EC	Evolutionary Computation
ECGA	Extended Compact Genetic Algorithm
EER	Equal Error Rate
ESC	Extended Shadow Code

XX

FAR	False Accept Rate
GA	Genetic Algorithm
GAR	Genuine Accept Rate
GLCM	Gray Level Covariance Matrix
GMM	Gaussian Mixture Model
HVS	Human Vision System
IW	Intelligent Watermarking
IDCT	Inverse Discrete Cosine Transform
JND	Just Noticeable Distortion
JPEG	Joint Photographic Experts Group
LPF	Low Pass Filter
LSGO	Large Scale Global Optimization
MF	Median Filter
MOGA	Multi-Objective Genetic Algorithm
MOOP	Multi-Objective Optimization Problem
MSE	Mean Squared Error
MWI	Maximum Watermark Image
NC	Normalized Correlation
NVF	Noise Visibility Function
PBIL	Population Based Incremental Learning
PSNR	Peak Signal-To-Noise Ratio

PSO	Particle Swarm Optimization
QF	Quality Fitness
RF	Robustness Fitness
ROC	Receiver Operating Characteristics
RS	Robustness Scores
SOOP	Single Objective Optimization Problem
SSIM	Structural SIMilarity
UMDA	Univariate Marginal Distribution Algorithm
UQI	Universal Quality Index
WL	Watermark Length
wPSNR	Weighted Peak Signal-To-Noise Ratio

LIST OF SYMBOLS

W	Original binary watermark logo of resolution $M_W \times N_W$
M_W	Width of binary watermark in pixels
N_W	Height of binary watermark in pixels
W'	Extracted watermark logo after applying watermarking attack
WL	Watermark length in bits
NC	Normalized Correlation representing the correlation between W and W' to measure watermark robustness
X_c	Original cover image of resolution $M_c \times N_c$ used to embed the watermark W
M_c	Width of cover image in pixels
N_c	Height of cover image in pixels
m_c	Index for rows of 8x8 pixel blocks in cover image X_c ranging from 1 to $M_c/8$
n_c	Index for columns of 8x8 pixel blocks in cover image X_c ranging from 1 to $N_c/8$
X_{cw}	Watermarked cover image after embedding the watermark W
$wPSNR$	Weighted Peak Signal-to-Noise Ratio measuring watermark quality
a	Index of AC coefficients of DCT transformed 8x8 pixels blocks ordered in zigzag order, $a = 1..63$
ac_a	DCT AC a th coefficient ordered in zigzag order
b	Cover face image 8x8 pixel block
B	Number of 8x8 pixels blocks in cover image where $B = (M_c/8) \times (N_c/8)$

TB	Number of 8x8 pixels textured blocks of cover image used for embedding the watermark
i	Index of 8x8 pixels blocks in cover image, with $i = 1..B$ and $i = (n_c - 1) \times (M_c/8) + m_c$
$C_{Max}(g)$	Maximum embedding capacity for blocks b_i of face image x_g
C_{b_i}	Embedding capacity representing number of bits allocated for block b_i
dct_i	DCT transform for cover image block b_i in zigzag order
BCR_i	Bit Correct Ratio for cover image block b_i
$PSNR_i$	Peak Signal-To-Noise Ratio for cover image block b_i
bwm_i	Block Watermarking Metrics for block b_i with aggregated fitness for BCR_i , and $PSNR_i$
e	Index of embedding band, with $e = 1..C_{b_i}$
$eb_e(b_i)$	The embedding band e of the face image block b_i , $eb_e(b_i) \in \{1..63\}$
$EB_{b_i}(d)$	Embedding bands for block b_i of the training set face image d , $EB_{b_i}(d) = \{eb_1(b_i), \dots, eb_e(b_i), \dots, eb_{C_{b_i}}(b_i)\}$
EB_{S_p}	Embedding bands of candidate solution S_p
pop	Population size of candidate solutions in EC methods
$P_{crossover}$	Probability of crossover operator in genetic algorithms
$P_{mutation}$	Probability of mutation operator in genetic algorithms
el	Index of elite candidate solutions
$Elite(el)$	Elite candidate solution of index el
EL	Number of elite candidate solutions

r_l	Reference watermark fitness used in chebyshev aggregation for quality and robustness fitness
λ	Weight for robustness fitness in weighted sum aggregation for Block Water-marking Metrics
γ	Weight for quality fitness in chebyshev aggregation used for $gf(EB_{X_c})$
$gf(EB_{X_c})$	Global aggregated fitness of embedding bands for cover image X_c
$Y_{(m_c, n_c)}(a)$	Block at the m_c th row and n_c th column of the cover image X_c transformed to DCT domain with coefficients $a = 0..63$ ordered in zigzag order
$Y'_{(m_c, n_c)}(a)$	Block at the m_c th row and n_c th column of the cover image X_c transformed to DCT and coefficients $e \in EB_{b_i}$ altered after embedding the watermark bits
$R(a)$	Ratio matrix between AC coefficients and DC used during embedding and extracting the watermark, where $a = 1..63$
$P_{(m_c, n_c)}(a)$	Polarity matrix for the block the m_c th row and n_c th column of the cover image X_c used during embedding and extracting the watermark
$W_{(m_c, n_c)}(e)$	Binary watermark bits allocated for embedding in the block b_i at the m_c th row and n_c th column of the cover image X_c
N	Training set size representing number of face images in training set
d	Index of training set representing the index of training face, with $d = 1..N$
D	Training set of face images texture features along with relevant optimal embedding bands $D = \{(x_1, y_1), \dots, (x_d, y_d), \dots, (x_N, y_N)\}$
M	Size of unseen testing face image stream fed into the system during generalization represented by number of face images
g	Index of unseen testing face image stream during generalization, with $g = 1..M$

G	Unseen testing face image stream during generalization, $G = \{x_1, \dots, x_g, \dots, x_M\}$
x_d	Texture features extracted from all blocks of face image d , where $x_d = \{TF_{b_1}(d), \dots, TF_{b_i}(d), \dots, TF_{b_B}(d)\}$
T	Texture feature vector size for 8x8 pixels blocks
t	Index of texture features in the texture feature vectors, with $t = 1..T$
$tf_t(b_i)$	The texture feature t of the face image block b_i which equals to the different DCT AC coefficients a_{c_a} of 8x8 block b_i
$TF_{b_i}(d)$	Texture features extracted from block b_i of the face image d , $TF_{b_i}(d) = \{tf_1(b_i), \dots, tf_t(b_i), \dots, tf_T(b_i)\}$
y_d	Optimized embedding bands for all blocks of face image d , $y_d = \{EB_{b_1}(d), \dots, EB_{b_i}(d), \dots, EB_{b_B}(d)\}$
k	Clustering resolution representing number of clusters ranging from K_{min} to K_{max} , $k = K_{min}, \dots, k, \dots, K_{max}$
j	Index for partition of blocks based on texture features, with $j = 1..k$
$p^{(k,j)}$	Cluster prototype for j th cluster using resolution k
SC_k	Set of clusterings using resolution k represented by k cluster prototypes, $SC_k = \{p^{(k,1)}, \dots, p^{(k,j)}, \dots, p^{(k,k)}\}$
T_{SC}	Time complexity measured in CPU time in seconds to calculate SC_k for all resolutions
$nb_{(k,j)}$	Number of blocks belonging to the j th cluster using resolution k
NB_k	Number of blocks belonging to different clusters using resolution k , $NB_k = \{nb_{(k,1)}, \dots, nb_{(k,j)}, \dots, nb_{(k,k)}\}$
W_r	Randomly generated binary watermark of length $WL = nb_{(k,j)}$ used for robustness scores calculation

$rs_{(k,j)}$	Robustness score for the j th cluster using resolution k
RS_k	Robustness scores for different clusters using resolution k , $RS_k = \{rs_{(k,1)}, \dots, rs_{(k,j)}, \dots, rs_{(k,k)}\}$
T_{RS}	Time complexity measured in CPU time in seconds to calculate RS_k for all resolutions
α	Robustness score threshold for candidate list of clusters
cl_k	Size of ordered list of candidate clusters for embedding using resolution k
c	Index of the ordered list of candidate embedding clusters with $c = 1..cl_k$
CL_k	Ordered list of candidate embedding cluster indices j_c for resolution k , $CL_k = \{j_1, \dots, j_c, \dots, j_{cl_k}\}$ where $rs_{(k,j_c)} > \alpha$
β	Maximum number of frequent embedding bands stored in BCM
f	Index of frequent embedding bands, with $f = 1..\beta$ where $f = 1$ represents the most frequent band and $f = \beta$ is the least frequent band
$fb_{(k,j)}(f)$	The f th frequent embedding band for j th cluster using resolution k , $fb_{(k,j)}(f) \in \{1..63\}$
$mfb_{(k,j)}$	Most frequent embedding bands for j th cluster using resolution k , $mfb_{(k,j)} = \{fb_{(k,j)}(1), \dots, fb_{(k,j)}(f), \dots, fb_{(k,j)}(\beta)\}$
MFB_k	Most frequent embedding bands for different clusters using resolution k , $MFB_k = \{mfb_{(k,1)}, \dots, mfb_{(k,j)}, \dots, mfb_{(k,k)}\}$
T_{MFB}	Time complexity measured in CPU time in seconds to calculate MFB_k for all resolutions
$QF(k)$	Watermark quality fitness for resolution k measured by wPSNR in dB
$RF(k)$	Watermark robustness fitness for resolution k measured by NC

T_{RF}	CPU time to compute watermark robustness fitness measured by NC for one face image
T_{QF}	CPU time to compute watermark quality fitness measured by wPSNR for one face image
$T_{fitness}$	Time complexity measured in CPU time in seconds to calculate watermark quality and robustness fitness for a stream

INTRODUCTION

Given the advances in face recognition, high-resolution face images have been used extensively in access control applications to verify individuals. For instance, in passport or smart card applications, authorized individuals are granted access to locations and information after verifying them using their facial captures. Large number of these high resolution facial captures must be secured during enrollment, prior to exchange and storage. Digital watermarking has been used to secure high-resolution facial captures by embedding invisible watermarks into these images to ensure integrity and authenticity. It can be used also to combine more biometric traits as invisible watermarks embedded in these captures.

Digital watermarking is characterized by the amount of distortion introduced by embedding watermark bit defined as watermark quality, and the ability to recover the watermark after manipulating the watermarked image defined as watermark robustness. Evolutionary Computation (EC) optimization methods have been widely used to find the watermark embedding parameters that satisfies the trade-off between both watermark quality and robustness in Intelligent Watermarking (IW) literature. Most of EC methods rely on a population of candidate solutions that evolves in the search space till the optimal solution is reached.

In IW literature, different EC techniques have been proposed to find optimal embedding parameters. Authors have proposed using EC optimization techniques like Genetic Algorithms (GA) (Shieh *et al.*, 2004), Particle Swarm Optimization (PSO) (Wang *et al.*, 2007), and combinations of GA and PSO (Lee *et al.*, 2008) to find embedding parameters that maximize the fitness for both quality and robustness (Vellasques *et al.*, 2010). Most of these traditional methods are based on representing all cover image 8×8 pixels blocks in candidate solutions according to their positional order, and iteratively improve the fitness until convergence is reached (Shieh *et al.*, 2004). These methods use single aggregated objective (Shieh *et al.*, 2004; Vellasques *et al.*, 2012). To date, few authors have proposed mutli-objective formulation (Diaz and Romy, 2005; Rabil *et al.*, 2010), where the two objectives are optimized simultaneously and multiple non-dominated solutions are located forming a Pareto front.

Problem Statement

Finding watermark embedding parameters to satisfy the trade-off between watermark quality and robustness is considered an optimization problem. The dimension of this problem is dependent on the resolution of the cover image. For high-resolution face images, the dimension of the optimization problem is high and thus increases its complexity. All blocks of the face image have to be represented in the search space and the candidate solutions to find optimal embedding parameters for all blocks. EC methods suffer from premature convergence for such large search space.

Handling a stream of grayscale high-resolution face images results in a stream of complex optimization problems with large search space. This stream of complex optimization problems is very expensive computationally even with using parallel fitness evaluations using Graphical Processing Units (GPU). Normally access control applications have large streams of high-resolution facial images captures, and thus using IW methods for such application is not feasible unless the computational complexity can be afforded by modest resources.

To tackle this stream of complex optimization problems, the strategy is to divide this into two main challenges. The first challenge is the stream of recurrent optimization problems, and the second challenge is to handle a single optimization problem with high dimension search space.

The first challenge can be addressed by replacing optimizations with memory recalls from an associative memory. This associative memory is populated using previous optimization results for few face images during a training phase. This approach utilizes the previous optimization results, and avoids expensive re-optimizations. The main research questions for this challenge are: What is the granularity of solutions to be stored in the associative memory? How to handle shifting of face pixels inside the face image without re-optimizations? How to calculate embedding capacities such that more textured blocks hold more watermark bits? What is the impact of using GPU for parallel fitness evaluations?

For the first challenge, the quality of solutions produced is dependent on the optimization results during training phase. Thus the second challenge boosts the quality of solutions produced

by the watermarking system. To avoid premature convergence, coevolution can be utilized to improve the exploration capabilities of traditional EC methods to find optimal embedding parameters. The main research questions for this challenge: What is the information exchanged during coevolution? What is the granularity of coevolution between candidate solutions? How to modify EC operators to improve the exploration capabilities of EC methods?

Objective and Contributions

The main objective of this thesis is to reduce the computational complexity of IW for streams of grayscale high-resolution facial images. The baseline system considered in this thesis is proposed by Shieh *et al.* (Shieh *et al.*, 2004), where GA is used to find optimal embedding parameters for grayscale images based on Discrete Cosine Transform (DCT). To date, the authors in IW literature did not pay attention to high-resolution grayscale images nor streams of grayscale images.

The main contribution in this research is a novel framework to handle streams of optimization problems with high dimension search spaces using modest computational resources. The proposed framework was experimented with optimization problems corresponding to watermark embedding parameters optimization for high resolution facial captures. This framework is characterized by a training phase where optimization is performed on few face images. The result of training phase is stored in associative memory. During generalization phase, sub-solutions are recalled from associative memory to be used for unseen stream of face images. The optimization in the training phase is improved using coevolution on block level to improve the exploration capabilities of EC methods.

The first contribution (see Chapter 2) is introducing Blockwise Multi-Resolution Clustering (BMRC) framework. During training phase, few face images are optimized and the sub-solutions for blocks are stored in associative memory. The sub-solutions are stored for all blocks belonging to the same cluster of blocks using their texture features. These sub-solutions are stored for different number of blocks clusters in the associative memory. During generalization phase, the face image blocks are clustered using different number of clusters. The

sub-solutions are recalled for the different number of clusters and watermark fitness is calculated. The optimal number of blocks clusters is decided at the end of the generalization phase using the fitness produced for different number of clusters. This implements multi-hypothesis approach, where sub-solutions corresponding to different number of clusters are calculated, and the decision is postponed till the end of the generalization. A novel grayscale texture measure is proposed for clusters of blocks called Robustness Scores (RS), such that the clusters of blocks of higher RS are utilized for higher embedding capacities.

The second contribution (see Chapter 3) is introducing Blockwise Coevolutionary Genetic Algorithm (BCGA) to address the premature convergence issues of traditional EC methods in high dimension search space. This algorithm is utilizing coevolution on block level using local metrics called Block Watermarking Metrics (BWM). A novel elitism mechanism is proposed to improve the exploration capabilities, where the blocks of higher BWM are assumed to have high global watermarking fitness for the whole face image when concatenated together. The crossover and mutation operators are modified such that they are performed on block level. This contribution improves the quality of solutions produced by BMRC framework by improving the optimization in training phase. Experimental results indicates significant improvement in the watermark fitness produced compared to other traditional EC methods.

An access control application is proposed for the contributions in this thesis (see Annex II). Offline signature features are proposed to be discretized and embedded as invisible watermarks in facial captures used in passports. The individuals crossing borders are verified using their facial captures and offline signatures captured from signed forms. The verification of individuals is significantly improved using both physical and behavioral biometric traits.

Organization of this Thesis

This thesis is organized into three main chapters. The overall framework described in the thesis is shown in Figure 0-1, it shows the contributions in the chapters of this thesis mapped to the overall framework. The overall framework BMRC presented in Chapter II consists of training phase where face images are optimized and then solutions are stored in an associative memory

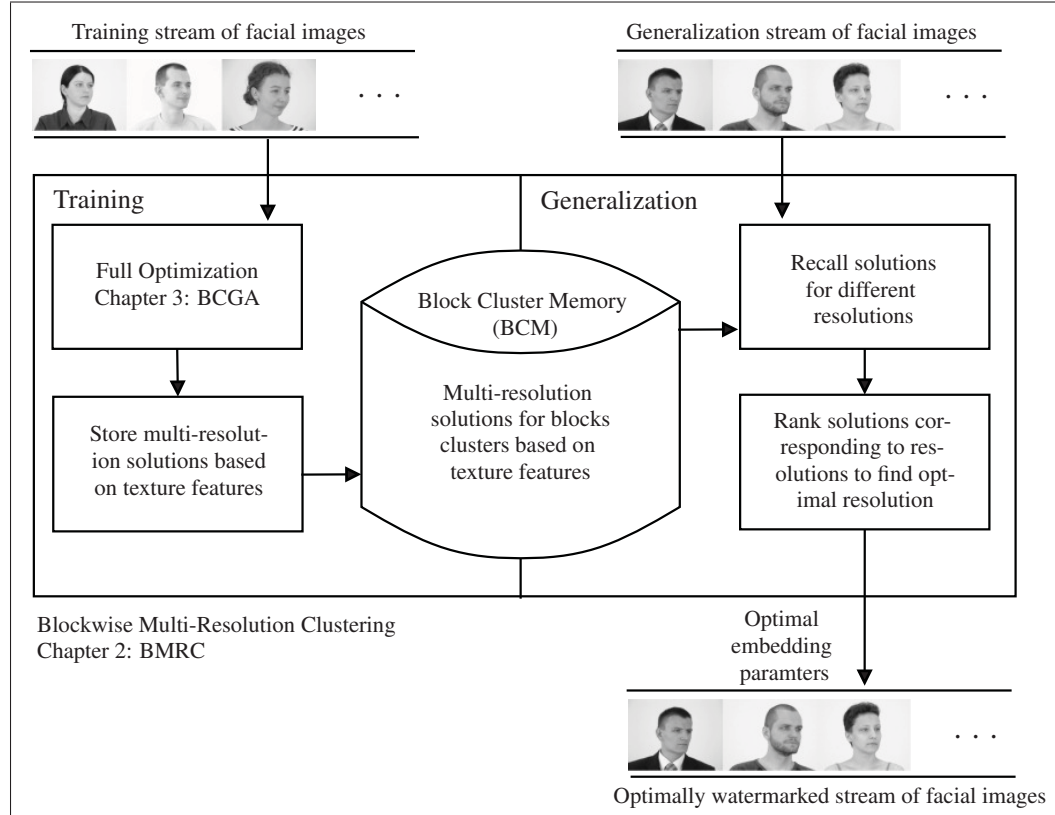


Figure-A 0-1 Proposed BMRC framework including BCGA for full optimization for streams of high-resolution facial images watermarking.

called BCM for different clustering resolutions, and generalization phase using memory recalls then solutions corresponding to clustering resolutions are ranked according to watermarking fitness. The first training step of BMRC involves full optimization of high resolution facial images, this corresponds to high dimensionality optimization problem which is addressed in Chapter III, where BCGA algorithm is proposed to utilize coevolution to resolve premature convergence in such high dimension search space.

Chapter I introduces the main concepts of intelligent watermarking including the watermarking metrics, embedding and extraction algorithm, and data flow for optimization problem corresponding to IW. This chapter also includes literature survey for key aspects on this research and its applications. Chapter II describes BMRC framework, the experimental methodology, and results to evaluate this framework. Finally Chapter III focuses on BCGA, the experimental methodology and results to evaluate this algorithm.

Chapter II focuses more on the BMRC framework to handle streams of grayscale high-resolution face images. This contribution is published in *Machine Vision and Applications* (Rabil *et al.*, 2013a), and WIPRA 2013 (Rabil *et al.*, 2013b). The comparison between different formulations of the optimization problem corresponding to IW is published at IHH-MSP 2010 (Rabil *et al.*, 2010). Formulating the stream of optimizations as single dynamic optimization problem is experimented and published at MDA 2011 (Rabil *et al.*, 2011b). Evaluation of the impact of using GPU in watermark embedding and extraction is published at CIINTI 2012 (García-Cano *et al.*, 2012).

Chapter III describes BCGA to address the optimization problem corresponding to IW of grayscale high-resolution face image with large search space. BCGA performance is compared against traditional GA as a baseline, and PBIL considered in Chapter II. The texture metric Robustness Scores (RS) (Rabil *et al.*, 2013a) is compared with common texture metrics used for selecting the most textured blocks for embedding. This contribution is published in *Expert Systems with Applications* (Rabil *et al.*, 2013c).

Annex I lists complementary experimentation for different texture metrics used for selecting textured blocks for embedding. Different metrics are evaluated using the produced fitness using mid range embedding DCT coefficient. The fitness produced is evaluated for different watermark lengths, and embedding capacities. This helps to determine the suitable metric to be used for different watermark lengths and capacities given the priority of different quality and robustness objectives. The fitness considered in this annex is an initial solution to be fed into BCGA for iterative fitness improvement.

Annex II describes an experimental study on embedding offline signature features instead of embedding logo images in facial images. The impact of watermarking attacks on the signature verification system is evaluated. The impact of different attack intensities, and bits used for quantization of features are considered in the experimentation. This contribution is published in CIBIM 2011 (Rabil *et al.*, 2011a). This annex represents a proposed application for this thesis in border crossing, where offline signature features are proposed to be embedded as invisible watermarks in passport high resolution facial image. On crossing borders, individuals are

verified using the facial image and comparing the extracted signature features with signatures captured from border crossing forms. Using two different biometric traits including a physical trait and a behavioral trait improves individual verification rates.

CHAPTER 1

INTELLIGENT WATERMARKING OF GRAYSCALE IMAGES

Digital watermarking is deployed in many domains to assure integrity and authenticity of the original signal via fragile and robust watermarking respectively (Vellasques *et al.*, 2010). A fragile watermark is a type of watermark to ensure integrity, but it is broken if the watermarked image is manipulated or altered, while the robust watermark ensures authenticity and can be extracted after manipulating the watermarked image. Semi-fragile watermark considered in this thesis is satisfying a trade-off between both the distortion introduced by the watermark and the watermark resistance to manipulations.

Most digital watermarking techniques proposed for grayscale images use different transform domains to embed a watermark that minimizes the visual impact, and to deal with the uncorrelated coefficients in the transform domain. The most commonly used transform domains in watermarking literature are Discrete Cosine Transform (DCT) (Shieh *et al.*, 2004) and Discrete Wavelet Transform (DWT) (Lee *et al.*, 2008). Using DCT transform inheriting robustness against JPEG compression which is based on DCT transform as well, the host image is divided into small blocks of pixels (8×8 pixels), transformed to frequency domain, and watermark bits are distributed among these blocks by changing frequency bands coefficients of these blocks according to the value of the watermark bit to be embedded. Few authors have considered other transforms based on Discrete Fourier Transform (DFT) (Licks and Jordan, 2005) to improve robustness against geometric attacks like image scaling, rotation, and cropping. DFT transforms are more resistant to geometric manipulations to images.

1.1 Watermarking Metrics

Digital watermarking system can be characterized using three main aspects: watermark quality, watermark robustness, and watermark capacity. Watermark quality measures the distortion resulting from watermark embedding, there are limits defined in literature (Voloshynovskiy *et al.*, 1999), where the human vision cannot recognize the distortion resulting from the embedding.

Watermark robustness measures the resistance to different manipulations and processing on the watermarked image, this is measured by the correlation between the extracted watermark after the manipulations and the original watermark. Watermark capacity measures the number of embedded bits per block given thresholds for watermark quality and/or watermark robustness.

1.1.1 Watermark Quality

The visual impact of a watermark can be evaluated by two different approaches - fidelity and quality. Fidelity is a measure of similarity between two signals (from a digital watermarking standpoint, the cover and watermarked signals, or more specifically, images). However, due to some particularities of the human visual system (HVS), the fidelity of a given image does not necessarily relates with the perceived quality by a human viewer.

Peak Signal-To-Noise Ratio (PSNR) is calculated between original image $X_{c(w,h)}$ and watermarked image $X_{cw(w,h)}$ of resolution $M_c \times N_c$ using the Mean Squared Error (MSE), where w , and h represents the index of pixels for width and height respectively:

$$MSE_c = \frac{1}{M_c \cdot N_c} \sum_{w=1}^{M_c} \sum_{h=1}^{N_c} (X_{c(w,h)} - X_{cw(w,h)})^2 \quad (1.1)$$

$$PSNR_c = 10 \log_{10} \left(\frac{255^2}{MSE_c} \right) [dB]$$

Weighted PSNR uses an additional parameter called Noise Visibility Function (NVF) which is a texture masking function defined by Voloshynovskiy *et al.* (Voloshynovskiy *et al.*, 1999). NVF arbitrarily uses a Gaussian model to estimate how much texture exists in any area of an image. For flat and smooth areas, NVF is equal to 1, and thus wPSNR has the same value of PSNR. For any other textured areas, wPSNR is slightly higher than PSNR to reflect the fact that human eye will have less sensitivity to modifications in textured areas than smooth areas. Weighted PSNR shown in Equation 1.2 is proposed in the latest benchmarking for

watermarking systems introduced by Pereira *et al.* (Pereira *et al.*, 2001).

$$wPSNR_c = 10 \log_{10} \left(\frac{255^2}{MSE_c \times NVF} \right) [dB] \quad (1.2)$$

Universal Quality Index (UQI) is another universal measure (Wang and Bovik, 2002) which is independent on the images being tested, the viewing conditions or the individual observers. This metric is composed of components to measure linear correlation, mean luminance, and contrasts of the images. This metric is subset of another more generic metric called Structural SIMilarity (SSIM) (Wang and Bovik, 2004) which takes into consideration luminance, contrast and structure. These quality metrics are more accurate than other metrics, but they are computationally expensive to be used in intelligent watermarking with huge number of fitness evaluations for quality fitness.

1.1.2 Watermark Robustness

Voloshynovskiy *et al.* (Voloshynovskiy *et al.*, 2001) have proposed a classification for different types of watermark attacks into four categories of attacks: removal attacks, geometric attacks, cryptographic attacks, and protocol attacks. Figure 1.1 shows the classification for different types of attacks. Removal attacks aim at the complete removal of the watermark information from the watermarked data without cracking the security of the watermarking algorithm, in other words without the key used for watermark embedding. Geometric attacks do not actually remove the embedded watermark itself, but intend to distort the watermark detector synchronization with the embedded information. The detector could recover the embedded watermark information when perfect synchronization is regained. Cryptographic attacks aim at cracking the security methods in watermarking schemes and thus finding a way to remove the embedded watermark information or to embed misleading watermarks. Protocol attacks aim at attacking the entire concept of the watermarking application. One type of protocol attack is based on the concept of invertible watermarks. The idea behind inversion is that the attacker subtracts his own watermark from the watermarked data and claims to be the owner of the watermarked data.

In intelligent watermarking literature, only removal attacks are considered for watermark robustness. The most common attacks from this category is the JPEG compression due to the popularity of JPEG in exchanging information via Internet, low pass filtering, and median filtering. These attacks are considered in the experimentation of this thesis. The watermark robustness metrics measure the similarity between the original watermark and the extracted watermark from the attacked image. The following metrics have been considered in literature:

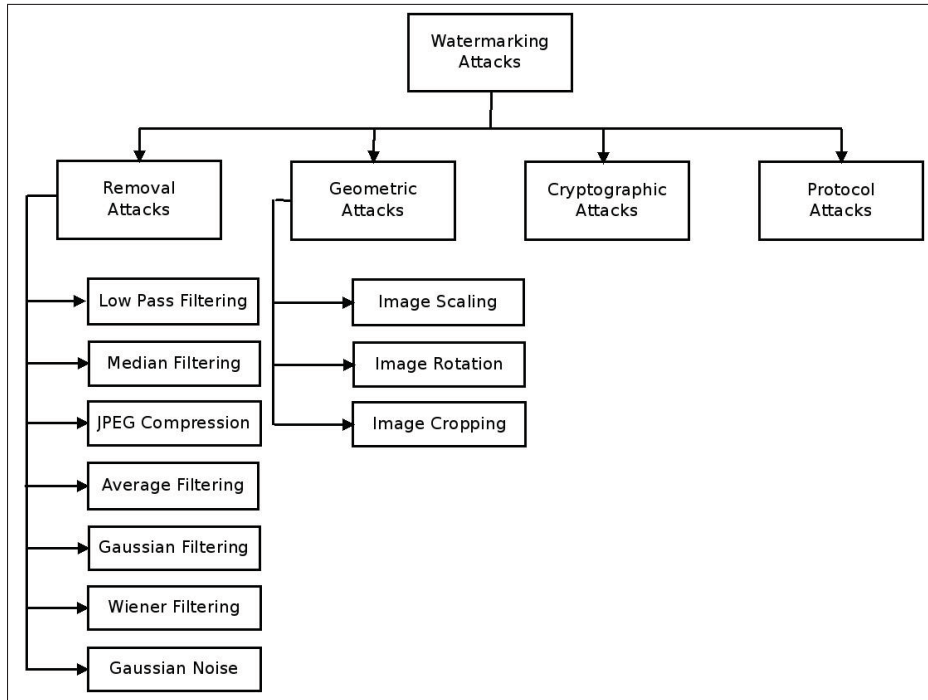


Figure 1.1 Classification for watermark attacks
(Voloshynovskiy *et al.*, 2001).

The Normalized Correlation (NC) is calculated between embedded watermark $W_{(w,h)}$ of resolution $M_W \times N_W$ where w and h represents the index of pixels for width and height respectively and the extracted watermark from the attacked image $W'_{(w,h)}$ using Equation 1.3.

$$NC = \frac{\sum_{w=1}^{M_W} \sum_{h=1}^{N_W} [W_{(w,h)} \times W'_{(w,h)}]}{\sum_{w=1}^{M_W} \sum_{h=1}^{N_W} [W_{(w,h)}]^2} \quad (1.3)$$

Wang *et al* (Wang *et al.*, 2007) have proposed using Accuracy Ratio (AR) to represent the robustness objective, which is the ratio between the number of bits correctly extracted (CB), and the number of bits of the original watermark (NB). Also Areef *et al* (Areef *et al.*, 2005) have proposed using Bit Correct Rate (BCR) as robustness metric as shown in equation 1.4, where \oplus is bitwise addition operator.

$$BCR(W, W') = \left(1 - \frac{\sum_{i=1}^{M_W \times N_W} (W_i \oplus W'_i)}{M_W \times N_W}\right) \times 100\% \quad (1.4)$$

Checkmark watermarking benchmark (Pereira *et al.*, 2001) have all these classes of attacks implemented. The attacks are grouped into different types of application domains, like copyright protection which takes into consideration all categories of attacks, and non-geometric application which takes into consideration removal attacks only.

1.1.3 Watermark Capacity

Watermark capacity represents the amount of information to be embedded as watermark. Many authors have addressed the issue of calculating the amount of information to be embedded as watermark bits. Cox *et al* (Cox *et al.*, 1999) and Moulin *et al* (Moulin *et al.*, 2000) have modeled the watermarking as information channel between transmitter and receiver. The requirement that the fidelity of the media content must not be impaired implies that the magnitude of the watermark signal must be very small in comparison to the content signal, analogous to a stringent power constraint in traditional communications. Barni *et al* (Barni *et al.*, 1999) have proposed a method based on DCT coefficients values to estimate the number of bits to be embedded in DCT domain. The amount of modification each coefficient undergoes is proportional to the magnitude of the coefficient itself. The watermark-channel is modeled such that the watermark is the signal and the image coefficients are the noise introduced by the channel. Voloshynovskiy *et al* (Voloshynovskiy *et al.*, 1999) introduced Noise Visibility Function (NVF) which estimates the allowable invisible distortion in each pixel according to its neighbors values.

Most recently Zhang *et al* (Zhang *et al.*, 2007) have proposed heuristic methods to determine adaptively the watermark capacity. The watermarking system is analyzed based on the channel capacity and error rate of the communication system, and the relation between the detection error rate with the capacity and payload capacity is derived. The error rate rises with the increase of watermarking capacity. Authors have introduced Maximum Watermark Image (MWI) in which the amplitude (value) of each pixel is the maximum allowable distortion calculated by NVF (Voloshynovskiy *et al.*, 1999). Yaghmaee *et al* (Yaghmaee and Jamzad, 2010) have proposed methods based on the image complexity, where authors have studied different image complexity metrics and its relation to watermark capacity.

In this thesis, watermark quality and robustness are measured using wPSNR and NC respectively. These metrics are the most commonly used metrics in IW literature and the most recent watermarking benchmark Checkmark. This helps to compare the experimental results with other contributions in literature. These metrics also have low computational complexity which fits the intelligent watermarking with extensive fitness evaluations for candidate solutions. Watermark capacity is measured using number of bits per 8×8 pixels block.

1.2 Embedding and Extracting Watermarks

The watermark embedding/extracting algorithm considered in this thesis is an algorithm proposed by Shieh *et al* (Shieh *et al.*, 2004), where the original cover image is not required during extraction of the watermark, this reduces the required space needed to store the original cover images. Using this algorithm, the cover image X_c to be watermarked of size $M_c \times N_c$ is divided into 8×8 blocks, and then transformed into DCT domain. The resultant matrix $Y_{(m_c, n_c)}(a)$ for each image block at row m_c and column n_c of cover image blocks has the upper left corner as DC coefficient and the rest of matrix are the AC coefficients, where the DCT coefficients index a ranging from 0 to 63 for 8×8 blocks are placed in zigzag order. The DCT transformed image $Y_{(m_c, n_c)}(a)$ is then used to get the ratio between DC and AC coefficients $R(a)$ for all AC

coefficients a using:

$$R(a) = \sum_{m_c=1}^{M_c/8} \sum_{n_c=1}^{N_c/8} \left(\frac{Y_{m_c, n_c}(0)}{Y_{m_c, n_c}(a)} \right), a \in [1, 2, \dots, 63] \quad (1.5)$$

Then polarities P are calculated using the Equation 1.6.

$$P_{(m_c, n_c)}(a) = \begin{cases} 1 & \text{if } (Y_{(m_c, n_c)}(a) \cdot R(a)) \geq Y_{(m_c, n_c)}(0) \\ & a \in \{eb_i\}, i = (n_c - 1) \times (M_c/8) + m_c \\ 0 & \text{otherwise} \end{cases} \quad (1.6)$$

Next, the watermarked DCT coefficient Y' is obtained using the Equation 1.7. The index of DCT coefficients modified belonging to $\{eb_i\}$ referred to as embedding bands for block b_i with i equals to $m_c \times (M_c/8) + n_c$. The embedding capacity for block b_i is defined as C_i in bits per block, and the watermark bits allocated for block at m_c row and n_c column $W_{(m_c, n_c)}(e)$, where e represents the index of set of embedding bands and finally the watermarked image X_{cw} is obtained using the inverse DCT for Y' .

$$Y'_{(m_c, n_c)}(a) = \begin{cases} Y_{(m_c, n_c)}(a) & \text{if } P_{(m_c, n_c)}(a) = W_{(m_c, n_c)}(e) \\ & a \in \{eb_i\}, i = (n_c - 1) \times (M_c/8) + m_c \\ (Y_{(m_c, n_c)}(0)/R(a)) + 1 & \text{if } P_{(m_c, n_c)}(a) = 0 \\ & W_{(m_c, n_c)}(e) = 1 \\ & a \in \{eb_i\}, i = (n_c - 1) \times (M_c/8) + m_c \\ (Y_{(m_c, n_c)}(0)/R(a)) - 1 & \text{otherwise} \end{cases} \quad (1.7)$$

1.3 Embedding Parameters Optimization

Modifications in certain frequency bands are less perceptible than others, and modifications in other frequency coefficients are more robust against manipulations. Many authors have therefore proposed using different evolutionary optimization techniques to find optimal frequency bands for embedding the watermark bits to maximize the fitness for both watermark quality and robustness objectives. The embedding parameters for frequency domain watermark embedding and extraction algorithms are represented using frequency coefficients altered due to watermark bits embedding which are commonly called embedding bands in literature.

EC methods like GA and PSO have attracted authors attention due to simplicity of these techniques and the ease in adapting them to many different types of watermarking systems. Moreover EC does not assume a distribution of the parameters space represented by selected frequency bands for embedding (Shieh *et al.*, 2004).

EC methods, inspired by biological evolution, are generally characterized by having candidate solutions which evolves iteratively to reach the target of optimization based on the guidance of objectives fitness evaluation. These candidate solutions are referred to as chromosome in GA, and more generally individuals of the population of candidate solutions.

In these traditional methods, all cover image blocks are represented in optimization candidate solutions, and the selected embedding bands are altered along optimization iteratively to maximize both the watermark quality fitness (QF) and robustness fitness (RF) simultaneously. All cover image blocks have to be represented in the optimization candidate solutions as shown in Figure 1.2 to allow distribution of watermark bits among cover image blocks. Each candidate solution consists of the set of embedding bands for all blocks of the cover image X_c . This is defined as $EB_{X_c} = eb_1, eb_2, \dots, eb_B$, where eb_i represents the embedding bands of block i which ranges from 1 to total number of blocks B . The size of the eb_i is dependent on the embedding capacity, for example in Figure 1.2 the size of eb_i equals to 4 which represents the embedding capacity measured in bits per block.

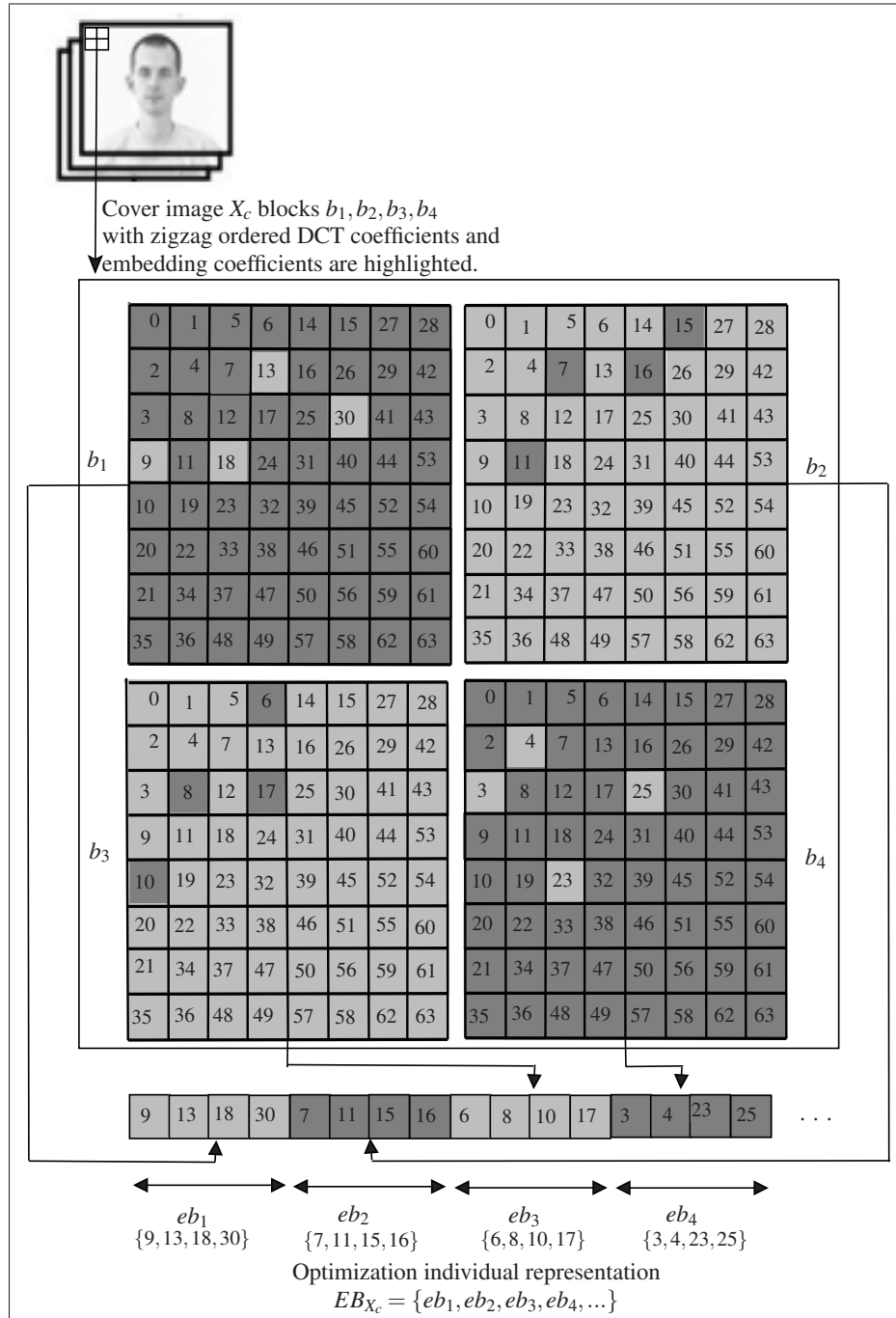


Figure 1.2 Optimization candidate solutions representation in traditional methods (Shieh *et al.*, 2004) with embedding capacity equals to 4 bits per block.

The traditional optimization formulation (Shieh *et al.*, 2004) for IW problem implies that embedding capacity are equal for all blocks of the cover image and at least 1 bit per block is embedded. In watermarking literature, this is referred to as even embedding scheme where

the embedding capacities are equal for all cover image blocks. From watermarking perspective (Wu, 2001), uneven embedding scheme is more suitable for better watermarking fitness where higher textured blocks are utilized for more bits to embed and smooth textured blocks are avoided for embedding.

Many authors have proposed aggregating both quality and robustness fitness into one objective for simplicity utilizing different aggregation weights for the objectives to resolve the issue of different scaling of these different types of objectives, and to favor one objective over the others using these weights. Shieh *et al* (Shieh *et al.*, 2004) have used Genetic Algorithm for optimizing the aggregated fitness for both quality and robustness, while Wang *et al* (Wang *et al.*, 2007) have used Particle Swarm Optimization for optimization. Other authors (Lee *et al.*, 2008) have proposed combining both GA and PSO for optimizing the aggregated fitness for quality and robustness.

Different formulations for watermark embedding optimization have been evaluated and compared in literature (Rabil *et al.*, 2010). Multi-objective formulation corresponds to the trade-off among different quality and robustness objectives. It provides multiple optimal non-dominated solutions (Pareto front) which gives a system operator the ability to choose among multiple solutions to tune the watermarking system (Rabil *et al.*, 2010) resolving the challenge of operating flexibility pointed out in (Haouzia and Noumeir, 2008).

1.3.1 Single Objective Optimization Problem (SOOP)

In traditional methods to optimize watermark embedding (Shieh *et al.*, 2004), all cover image blocks are represented in optimization candidate solutions, and the selected embedding bands are altered iteratively during optimization to maximize the aggregated fitness $gf(EB_{X_c})$. This aggregated fitness includes watermark Quality Fitness (QF) and Robustness Fitness (RF) against different attacks. All cover image blocks have to be represented as optimization candidate solutions to allow distribution of watermark bits among cover image blocks. The optimization problem can be formalized as the following maximization problem:

$$\begin{aligned}
& \max_{EB_{X_c}} \{gf(EB_{X_c})\} \\
& EB_{X_c} = \{eb_1, eb_2, \dots, eb_i, \dots, eb_{NB}\}, \text{ where } NB = (M_c/8) \times (N_c/8) \\
& eb_i = \{a_1, a_2, \dots, a_e, \dots, a_{C_i}\}, \text{ where } a_e \in [0, 1, \dots, 63] \\
& \text{s.t. } a_e \neq 0, \text{ where } 1 < e < C_i, \text{ and } 1 < i < NB \\
& a_{e1} \neq a_{e2}, \text{ where } 1 < e1, e2 < C_i
\end{aligned} \tag{1.8}$$

where b_i represents the 8×8 block in cover image of resolution $M_c \times N_c$, the total number of blocks equals to NB , a_e represents the e th embedding band for block b_i , and the embedding capacity for block b_i is C_i . The first constraint considered ensures avoiding DC coefficient a_e for embedding, and the second constraint considered ensures avoiding using the same embedding bands for the same image block.

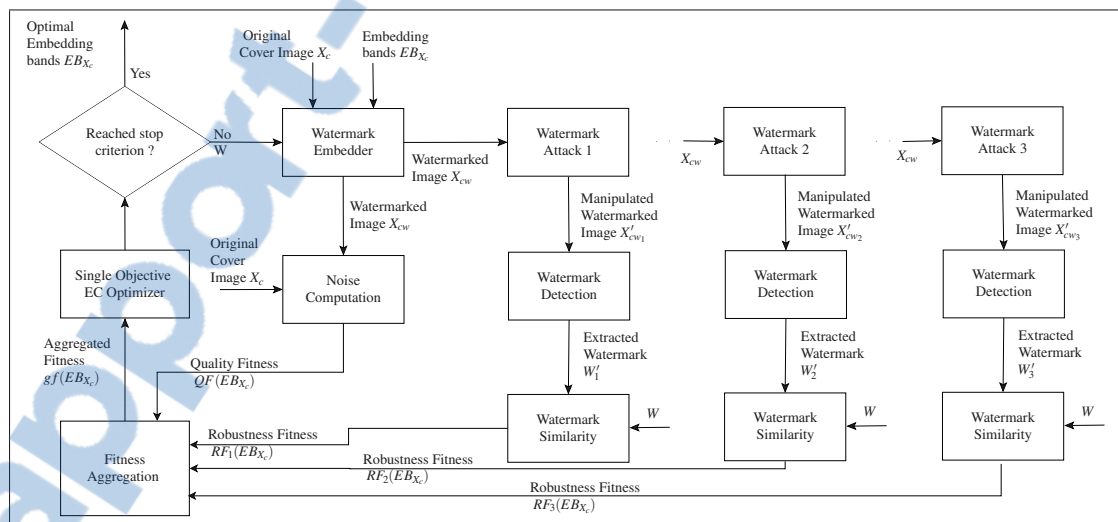


Figure 1.3 Data flow diagram depicting the single objective formulation for watermark embedding optimization to find optimal embedding bands EB_{X_c} for cover image X_c .

The flow chart of the IW optimization is shown in Figure 1.3. Assume that watermark bits W are embedded in face image X_c using the embedding bands EB_{X_c} to produce the watermarked image X_{cw} . The quality fitness $QF(EB_{X_c})$ between the original face image X_c , and

the watermarked image X_{cw} is assessed. Then different attacks are applied to X_{cw} , then watermark bits are extracted W'_1 , W'_2 , and W'_3 after applying the watermark attacks. The extracted watermark W'_1 , W'_2 , and W'_3 are compared against original watermark W to calculate robustness fitness $RF_1(EB_{X_c})$, $RF_2(EB_{X_c})$, and $RF_3(EB_{X_c})$ respectively against different attacks. In this formulation quality fitness $QF(EB_{X_c})$ and different robustness fitness $RF_1(EB_{X_c})$, $RF_2(EB_{X_c})$, and $RF_3(EB_{X_c})$ are optimized as a single objective optimization using aggregation. The aggregated fitness is improved iteratively till a stopping criterion is reached, and the sets of optimal embedding bands eb_i for all blocks b_i are concluded EB_{X_c} for face image X_c .

1.3.2 Multi-Objective Optimization Problem (MOOP)

The optimization problem using multi-objective formulation can be formalized as shown in equation 1.9, where b_i represents the 8×8 block in cover image of resolution $M_c \times N_c$, the total number of blocks equals to NB , a_e represents the e th embedding band for block b_i , and the embedding capacity for block b_i is C_i . The same constraints described in 1.3.1 applies to the multi-objective formulation.

$$\begin{aligned}
 & \max_{EB_{X_c}} \{QF(EB_{X_c}), RF(EB_{X_c})\} \\
 & EB_{X_c} = \{eb_1, eb_2, \dots, eb_i, \dots, eb_{NB}\}, \text{ where } NB = (M_c/8) \times (N_c/8) \\
 & \quad eb_i = \{a_1, a_2, \dots, a_e, \dots, a_{C_i}\}, \text{ where } a_e \text{ is 6-bit binary representation} \\
 & \quad \text{for embedding bands index for block } b_i \text{ with } a_e \in [0, 1, \dots, 63] \\
 & \text{s.t. } a_e \neq 0, \text{ where } 1 < e < C_i, \text{ and } 1 < i < NB \\
 & \quad a_{e1} \neq a_{e2}, \text{ where } 1 < e1, e2 < C_i
 \end{aligned} \tag{1.9}$$

The data flow of the multi-objective formulation is shown in Figure 1.4, where this formulation deals efficiently with conflicting objectives like watermark quality and robustness against different attacks. In this formulation quality fitness $QF(EB_{X_c})$ and robustness fitness $RF(EB_{X_c})$ are optimized simultaneously without favoring any objective over the other using aggregation

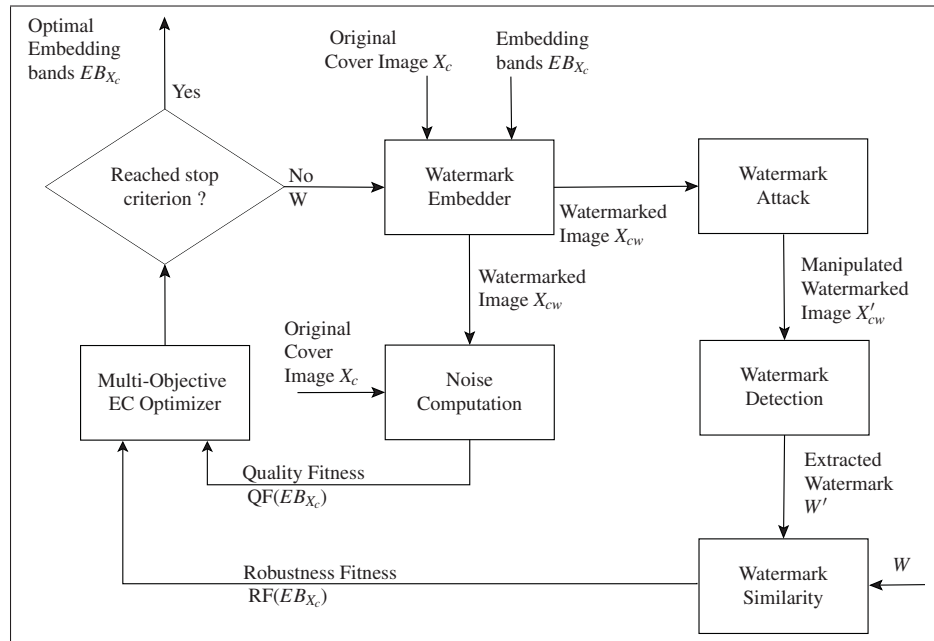


Figure 1.4 Data flow diagram depicting the multi-objective formulation for watermark embedding optimization to find optimal embedding bands EB_{X_c} for cover image X_c .

weights. The fitness for both objectives are improved iteratively till stop criterion is reached, and optimal embedding bands eb_i for all blocks b_i are concluded from a single non-dominated solution belonging to the resulting Pareto front.

In the case of multiobjective optimization, there is no single optimal solution, instead there is a set of non-dominated solutions representing Pareto Optimal Front (POF) as shown in figure 1.5. A solution vector x is said to dominate the other solution vector y if the following two conditions are true: The solution x is no worse than y in all objectives; and the solution x is strictly better than y in at least one objective.

The taxonomy of optimization approaches proposed in watermarking literature is shown in Table 1.1, where these approaches can be categorized into Single Objective Optimization Problem (SOOP), and Multi-Objective Optimization Problem (MOOP). Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Hybrid of GA and PSO have been proposed as single objective optimization techniques where single objective is optimized where it can be only quality objective or aggregated objective with both quality and robustness combined together

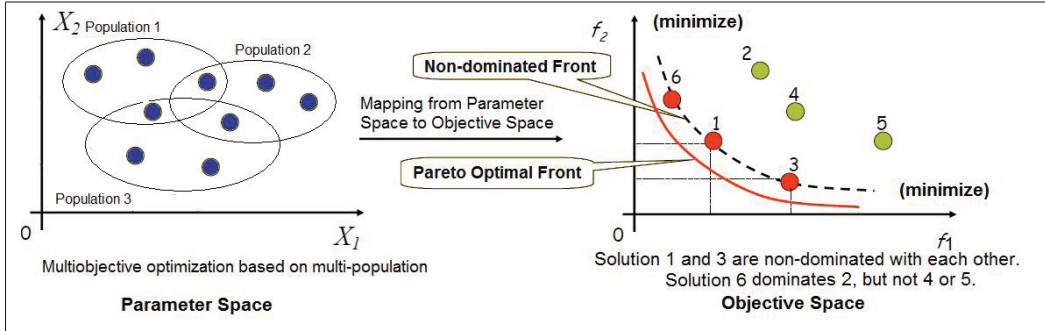


Figure 1.5 Pareto Optimal Front (POF) and non-dominated solutions for multi-population optimization (Li and Engelbrecht, 2007).

Table 1.1 Survey for different EC-based methods for watermarking grayscale images.

Problem	Method	Formulation	Metrics	Vars.	Contribution
SOOP	GA	Single	PSNR	DCT	(Chen and Lin, 2007)
			PSNR	LSB	(Wang <i>et al.</i> , 2001)
			PSNR	LSB	(Ji <i>et al.</i> , 2006)
			PSNR	DCT	(Shih and Wu, 2005)
			NC	DCT	(Wei <i>et al.</i> , 2006)
			PSNR	DCT	(Wu and Shih, 2006)
SOOP	PSO	Aggregated	PSNR+BCR	DWT	(Areef <i>et al.</i> , 2005)
			PSNR+NC	DCT	(Shieh <i>et al.</i> , 2004)
			PSNR+NC	DCT	(Huang <i>et al.</i> , 2007)
			UQI+DIF	DMT	(Kumsawat <i>et al.</i> , 2005)
SOOP	GA+PSO	Aggregated	PSNR	DCT	(Aslantas <i>et al.</i> , 2008)
			PSNR	DCT	(Li and Wang, 2007)
			PSNR+AR	DMT	(Wang <i>et al.</i> , 2007)
SOOP	GA+PSO	Aggregated	PSNR+NC	DCT	(Zhu and Liu, 2009)
			PSNR+NC	DWT	(Lee <i>et al.</i> , 2008)
			PSNR+NC	DCT	(Diaz and Romay, 2005)
MOOP	MOGA	Multi-Objective	PSNR+NC	DCT	(Diaz and Romay, 2005)
			PSNR+NC	DCT	(Sal <i>et al.</i> , 2006)

into one objective. Only GA was proposed for multi-objective optimization technique for digital watermarking domain.

1.3.3 Dynamic Optimization

Intelligent watermarking a stream of bi-tonal documents have been addressed recently (Vellasques *et al.*, 2011). The authors formulated the stream of similar optimization problems as a single dynamic optimization problem, where the new image fed into the system is considered a change in the optimization environment as shown in Figure 1.6. During a change in optimization environment of SOOP, the optimum can suffer a variation either in the parameter (type I), fitness (type II) or both spaces (type III) (Nickabadi *et al.*, 2008). A change is subject to severity in space and time. For minor environment changes, memory recalls is sufficient to find solutions of satisfactory fitness. The dynamic optimization problem is characterized by having a changing maximum value and location. For bi-tonal images considered (Vellasques *et al.*, 2011), the embedding parameters to be optimized are global parameters for the whole document image.

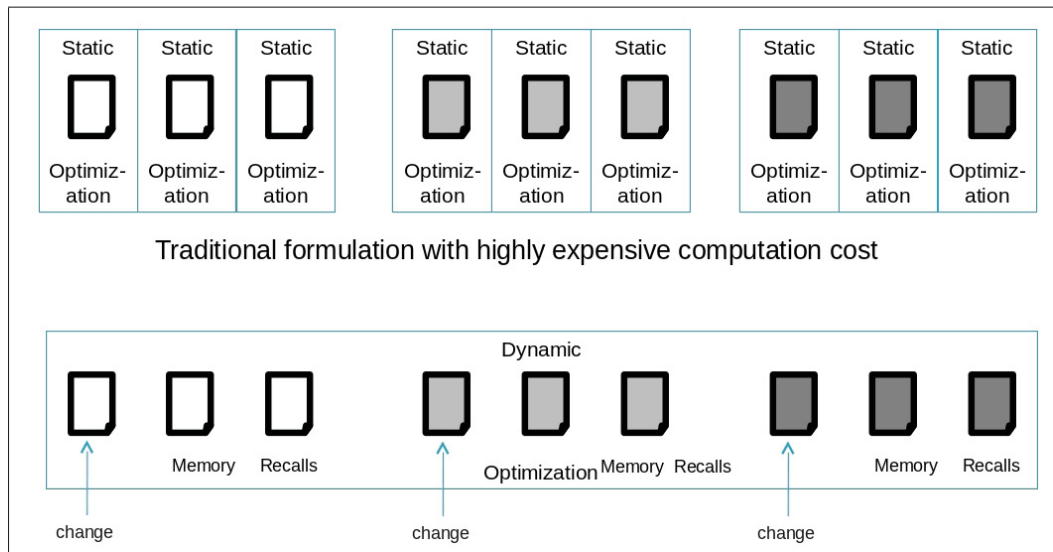


Figure 1.6 Intelligent watermarking for stream of images as single dynamic optimization problem (Rabil *et al.*, 2011b),(Vellasques *et al.*, 2011).

As shown in Figure 1.6, the traditional formulation for a stream of static optimization problems is compared to dynamic optimization formulation. The traditional formulation results in

recurrent static optimization problems corresponding to watermarking the stream of images. For the dynamic optimization formulation, watermarking a stream of images is considered a single optimization problem in a dynamic environment. For homogeneous streams of images, the optimization environment for one image can be utilized for the next image in the stream. Depending on the similarity between images in the stream, the previous optimization can be recalled on solutions level or optimization landscape level. Whenever a different image in the stream, this is considered as a change in optimization problem, and thus full optimization is required for such image. This results in significant complexity reduction for watermarking optimally stream of images.

In an earlier work (Rabil *et al.*, 2011b) by the authors, this approach was applied to stream of high-resolution grayscale facial images as dynamic multi-objective optimization problem. Farina *et al* (Farina *et al.*, 2004) have categorized the multi-objective problems in dynamic environments into four types depending on changes in both parameter space and objective space. The parameter space includes decision variables to be tuned for optimization, and the objective space includes the optimal objective values representing the Pareto front of non-dominated solutions. As shown in Figure 1.7, these four types of dynamic multi-objective optimization problems are described as follows:

- *Type I*: Optimal decision variables change with time, whereas the optimal objective values do not change.
- *Type II*: Both optimal decision variables and optimal objective values are changing.
- *Type III*: Optimal decision variables do not change with time, whereas the optimal objective values change.
- *Type IV*: Neither optimal decision variables nor optimal objective values change.

The multi-objective optimization of digital watermarking of grayscale images can be considered of Type II where both optimal decision variables and optimal objective values change when a new image is fed into the system.

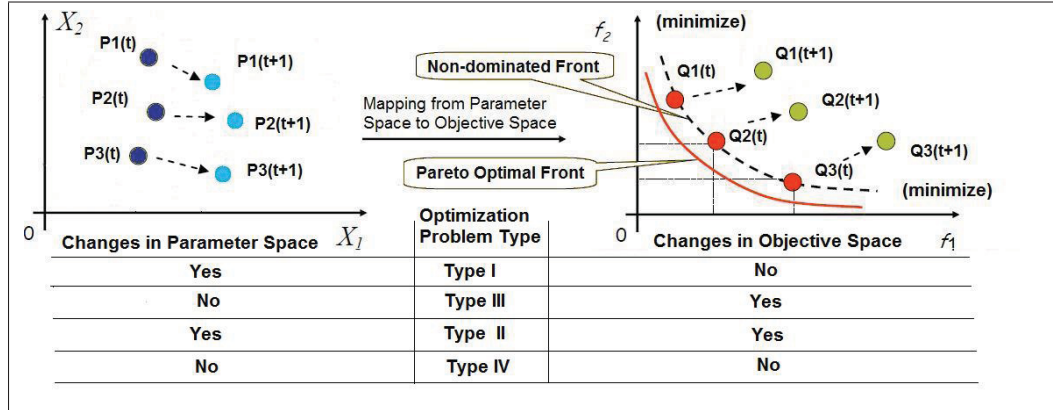


Figure 1.7 Different types of Dynamic MultiObjective Optimization Problems (DMOOP) (Farina *et al.*, 2004).

Using multi-objective optimization with dynamic environment is adding one more challenge to previous challenges for optimization mechanisms in dynamic environment optimization which is the change detection mechanism to detect changes in the environment. Most of literature in dynamic optimization for single objective was based on sentry particles/individuals as change detection mechanism where some particles/individuals are chosen and their fitness is compared to the stored fitness values, and if the fitness is varying, then a change has occurred like the method proposed by Hu and Eberhart (Hu and Eberhart, 2002). In multi-objective optimization in a dynamic environment, sentry particles becomes in-adequate in case of problems where the Pareto front in decision space is changing like problems of Type III and IV, while this can be applied to dynamic multi-objective optimization of Type I and II as proposed by Zheng (Zheng, 2007) and also Greeff and Engelbrecht (Greeff and Engelbrecht, 2010).

For Dynamic Multi-Objective Optimization Problems (DMOOP) of Type III and IV, some authors have considered other change detection with regards to changes in the Pareto set using some proposed metrics for the Pareto set like hypervolume and generational distance like the approaches proposed by Camara *et al* (Camara *et al.*, 2009) and Li *et al* (Li *et al.*, 2007) respectively. Table 1.2 summarizes different proposed mechanisms for change detection in dynamic optimization environment.

Table 1.2 Survey for different change detection mechanisms in dynamic environments proposed in literature.

Optimization Problem	Type of Problem	Method Proposed	Contribution
SOOP	Type I,II,III	Sentry Particles	(Hu and Eberhart, 2002)
MOOP	Type I, II	Sentry Particles	(Zheng, 2007)
	Type III,IV	Generational Distance HyperVolume	(Greeff and Engelbrecht, 2010) (Li <i>et al.</i> , 2007) (Camara <i>et al.</i> , 2009)

For DMOOP two groups of performance metrics exist, namely performance metrics where the true POF is known and performance metrics where the true POF is unknown. The convergence, measured by the generational distance proposed by Van Veldhuizen (Veldhuizen and Lamont, 2000), and spread or distribution of the solutions are often used to measure an algorithm performance when the POF is known (Goh and Tan, 2007; Zeng *et al.*, 2006). The reversed generational distance and the collective mean error were proposed as performance metrics by Branke *et al* (Li *et al.*, 2007). Another metric, the $HVR(t)$ metric, represents the ratio of the hypervolume of the solutions and the hypervolume of the known POF at a specific time (Li *et al.*, 2007; Veldhuizen and Lamont, 2000). Li *et al* (Li *et al.*, 2007) proposed a metric of spacing that can be used when the true POF is unknown. Measures of accuracy, stability and reaction capacity of an algorithm proposed by Camara *et al* (Camara *et al.*, 2007).

This approach reduced the computational complexity for the stream of optimization problem for streams of minimal changes in face pixels from one face in the stream to the other. This approach is sensitive to changes in location of face pixels inside the image due to the positional representation of face image blocks in the optimization candidate solutions

1.4 Large-Scale Global Optimization Problems

Many real-world applications correspond to large scale global optimization (LSGO) problems to find optimal decision variable values in large search space. Authors in many domains

have explored different strategies to handle such challenging optimization problems (Shan and Wang, 2010). Screening strategy involves removing noises and insignificant variables from the large problem by exploring the nature of the optimization problem. Mapping strategy reduces the search space by removing correlated variables and thus reducing the complexity of the large problem. And the most commonly used strategy used by authors is decomposing the large problem into smaller sub-problems, different decomposition and interaction methods have been proposed to be used among these sub-problems. In this thesis, decomposing and screening strategies are considered for handling optimization problem corresponding to IW of high-resolution facial images. These two strategies are more suitable for the nature of IW problem.

EC methods are not efficient for large scale problems, this attracted the attention of authors in EC literature to improve the capabilities of these methods for large dimension of search space. In recent years, there has been a growing interest in methods that learn the structure of a problem on the fly and use this information to ensure an adequate evolution of the different variables of the problem. One approach is based on probabilistic modeling of promising solutions to guide the exploration of the search space instead of using crossover and mutation, as in the case of simple GA (Pelikan *et al.*, 2002).

In the Population Based Incremental Learning algorithm (PBIL) (Baluja, 1994), solutions are represented by binary strings of fixed length. The population of solutions is replaced by a probability vector which is initially set to the same probability 0.5 for all positions. After generating a number of solutions, the best solutions are selected, and the probability vector is shifted to the chosen solutions. The probability vector of PBIL evolves throughout generations using the previously calculated fitness. Also the probability vector is considered a good representation for optimization landscape that can be recalled to reproduce the landscape without the need to go through complex iterations. Bureerat and Sriwornamas (Bureerat and Sriwornamas, 2007) proposed changes to PBIL algorithm to handle multi-objective optimization problems. In this algorithm the probability vector is replaced with probability matrix, where each row in this matrix represents the probability vector to create sub-population individuals.

The Compact Genetic Algorithm (CGA) (Harik *et al.*, 1999) replaces the population with a unique probability vector similar to PBIL. However, unlike the PBIL, it modifies the probability vector so that there is a direct correspondence between the population that is represented by the vector of probabilities and the probability vector itself. Each component of the vector is updated by replacing its value by the contribution of a single individual in the total frequency assuming a particular population size. Using this update rule, the theory of simple genetic algorithms can be used directly to estimate the parameters and behavior of the compact genetic algorithm.

Previous methods based on learning the structure of the problem are not efficient for problems of high dimension search space like the IW problem for high-resolution face images. Thus such problems would involve other strategy for decomposing this large problem into smaller sub-components interacting together to find optimal overall solutions. Recently authors adopted coevolution strategy for handling such optimization problems.

In biology, coevolution is defined as a change of a biological object triggered by the change of a related object. In bio-inspired EC techniques this concept is adopted either this coevolution is cooperative or competitive. In cooperative coevolution algorithms (Potter and DeJong, 2000), the goal is to find individuals from which solutions can be better. The adaptability of the individual depends on its ability to cooperate with individuals of other species to solve a given problem. On the other side for competitive coevolution (Rosin and Belew, 1997), the adaptability of the individual depends on the competition with other individuals of other species, where each of these species compete with the rest of species.

The basic model for coevolution is shown in Figure 1.8, each species is evolved in its own population and adapts to the environment through the repeated application of an Evolutionary Algorithm (EA). The species interact with one another within a shared domain model and have a cooperative relationship (Potter and DeJong, 2000).

Cooperative Coevolution (CC) adopts natural divide-and-conquer strategy, recently this approach is considered a promising solution for handling high-dimensional optimization prob-

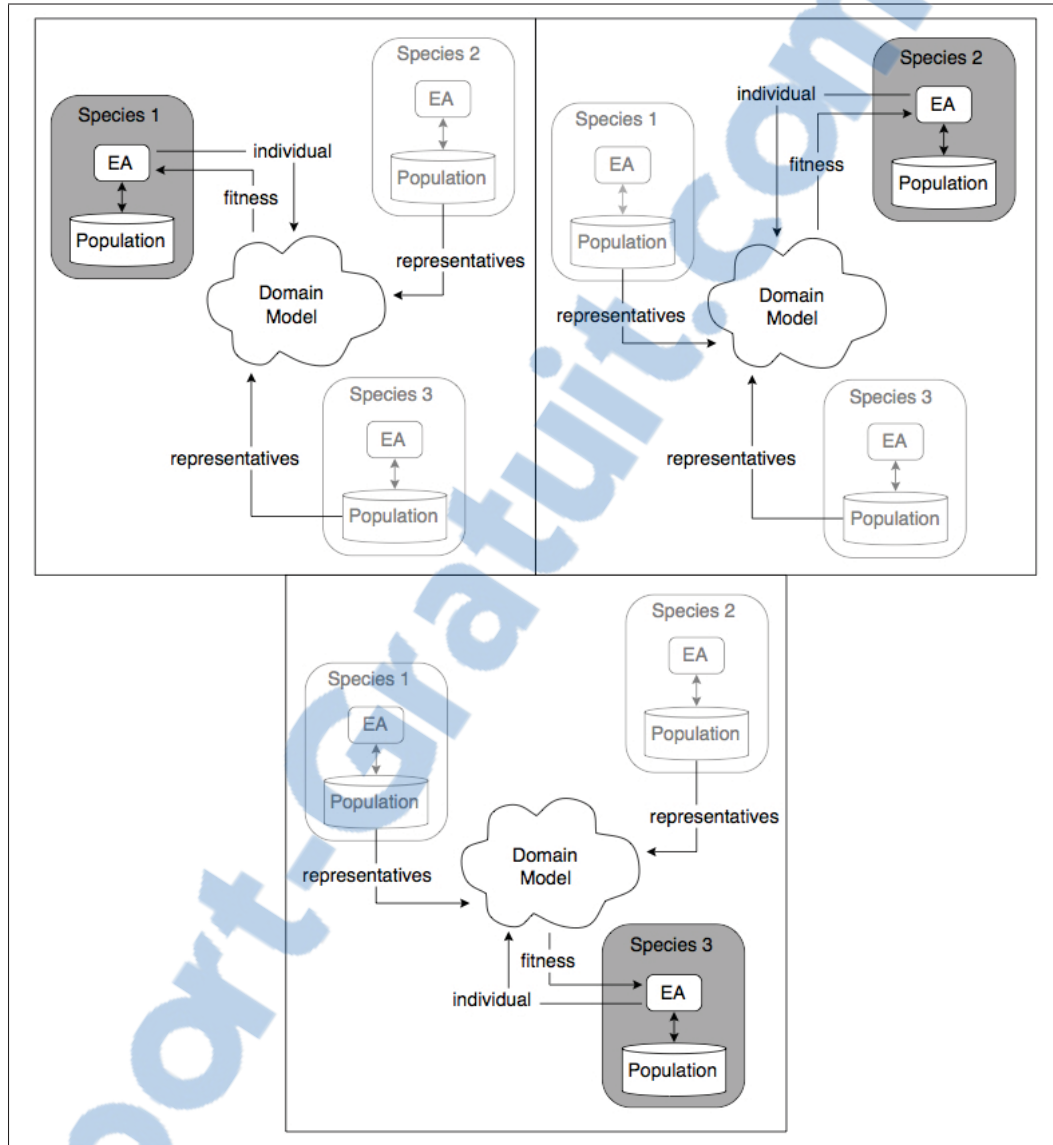


Figure 1.8 Coevolutionary model of three species shown from the perspective of each in turn (Potter and DeJong, 2000).

lems. The main idea of cooperative coevolution based algorithms is to identify which decision variables, i.e., dimensions, of the search space interact. Non-interacting variables can be optimized as separate problems of lower dimensionality. Interacting variables must be grouped together and optimized jointly (Chen *et al.*, 2010). By cooperatively coevolving multiple EA subpopulations (each dealing with a subproblem of a lower dimensionality), we can obtain an

overall solution derived from combinations of subsolutions, which are evolved from individual subpopulations.

Based on this principle, Potter *et al.* (Potter and DeJong, 2000) proposed a Cooperative Coevolutionary Genetic Algorithm (CCGA), which shares the search space by dividing the solution vector into smaller vectors. Potter *et al.* (Potter and DeJong, 2000) proved that CCGA has a significant performance improvement over traditional genetic algorithms for optimization problems of 30 variables. Subsequently, Sofge *et al.* (Sofge *et al.*, 2002) extended the model of Potter *et al.* (Potter and DeJong, 2000) to algorithms of evolution strategy (CCES).

Van den Bergh and Engelbrecht (Bergh and Engelbrecht, 2004) adopted the cooperative co-evolution concept with PSO (CCPSO), this cooperative model was tested on functions of up to 30 variables (Bergh and Engelbrecht, 2004) and 190 variables (Bergh, 2002). All previous algorithms based on cooperative co-evolution adopted two simple strategies for problem decomposition. The first strategy decomposes a high-dimensional vector into single variables, this means an n -dimensional problem would be decomposed into n one-dimensional problems. This strategy is simple but it did not consider interdependencies among variables for non separable problems. The second strategy is splitting-in-half which decomposes a high-dimensional vector into two equal halves and thus reducing an n dimensional problem into two $\frac{n}{2}$ -dimensional problems. This can be extended to recursively decompose into halves for larger dimension problems.

Yang *et al.* (Yang *et al.*, 2008) proposed grouping strategy and adaptive weighting for better capturing of the variable interdependencies for non separable problems. Li and Yao (Li and Yao, 2012) presented a new Cooperative Coevolving PSO (CCPSO) using random variable grouping technique. This proved efficiency with problems with dimensions up to 2000 real valued variables using the benchmarking functions provided in Congress on Evolutionary Computation (CEC'10) (Tang *et al.*, 2009).

For optimization problems corresponding to IW of high-resolution facial images considered in this thesis, the number of variables is equal to 49k variables. These embedding variables

for different blocks affect the overall watermarking fitness for the facial image, however the local watermarking metrics of these blocks can guide the search in such large search space. Such dimension of optimization problems was not considered before in literature, and thus an application specific algorithm is proposed in this thesis to handle such problem using the characteristics of the IW problem.

1.5 Texture Features and Metrics of Grayscale Images

This section describes briefly the common methods to extract texture features from grayscale images. These features are used to cluster blocks based on their texture features. Also it describes different grayscale texture metrics used to select the most textured blocks for embedding, this assures no bits are embedded in smooth textured areas.

1.5.1 Extract Grayscale Texture Features

Texture features are extracted from the grayscale images using 8×8 pixels blocks granularity. The most commonly used texture features can be classified into spatial features like Gray Level Covariance Matrix (GLCM), and other domains features like Discrete Cosine Transform (DCT), and Gabor Wavelet Transform. Taking the computational complexity into consideration, using spatial features would have lower complexity compared to other domains features like DCT domain.

Gray Level Covariance Matrix (GLCM) features use the minimum and maximum intensity discovered in the image as the limits of grayscale intensity to calculate how often a pixel of certain grayscale intensity occurs horizontally adjacent to a pixel with another intensity value. However the watermark embedding and extraction methods based on spatial domain would have lower robustness against different image alterations, and lower watermark embedding capacity. In literature, many authors have considered DCT for extracting texture features of grayscale images. Yu *et al.* (Yu *et al.*, 2010) have proposed zoning method using the most significant 39 coefficients. Sorwar and Abraham (Sorwar and Abraham, 2004) have proposed selecting two different texture features based on lower coefficients, and directional edges coefficients,

where the coefficients of the most upper and left regions represent vertical and horizontal edge information respectively.

In this research, the computational complexity is the main challenge addressed for IW of streams of grayscale high-resolution images. The watermark embedding and extracting algorithm described in Section 1.2 is based on DCT transform which inherits better robustness against JPEG compression, and thus the DCT coefficient are already extracted during embedding or extracting the watermark. Using DCT coefficients for texture features extraction does not add any computational complexity, however using other texture representation would involve adding more complexity to the system even if this representation is based on spatial features with no transforms involved.

In even embedding scheme of watermarking, all cover image blocks have equal embedding capacities. Even embedding results in embedding watermark bits in smooth textured blocks which results in degradation of watermark quality and robustness fitness. This scheme can be useful for images whose blocks are all textured. For facial images considered in this thesis, there is smooth textured blocks which should be avoided to be used for embedding watermark bits.

Uneven embedding scheme is more suitable for facial images considered. It can be used with simple embedding capacity calculation algorithm, where the most textured blocks are used for embedding, and the watermark bits are distributed equally among these textured blocks. Another advanced algorithm for embedding capacity is proposed in this thesis in Chapter 2 to distribute watermark bits among textured blocks, such that the blocks of highest texture holds the maximum embedding capacity and the capacity decreases for lower textured blocks until all watermark bits are embedded. This scheme requires clustering blocks according to their texture to calculate the embedding capacity based on their texture features.

1.5.2 Grayscale Texture Metrics

Adaptive watermarking systems proposed in literature model the human vision system to measure the level of tolerable distortion in an image. Accordingly the insertion of the mark is

adjusted to these levels of distortion in an image to maximize robustness and quality. These systems involve ranking cover image blocks according to texture. The most textured blocks are used for embedding the watermark, thus the embedding parameters and regions are adaptive to the image content. Perceptual masks are proposed to perform the selection of textured blocks for embedding. Table 1.3 shows the different categories of perceptual masks proposed in literature. In this thesis lower complexity masks are considered in the experimentation due to dealing with high-resolution cover images.

Table 1.3 Perceptual masks literature survey.

Spatial	Texturization	Texture Measure	(Kim and Suthaharan, 2004) (Kamble <i>et al.</i> , 2010)
		Texture Model	(Yuan and Zhang, 2006)
	Distortion Estimation	(Yixin <i>et al.</i> , 2009) (Voloshynovskiy <i>et al.</i> , 1999)	
Frequency		(Li <i>et al.</i> , 2008) (Gao <i>et al.</i> , 2006)	

Most of authors proposed spatial perceptual masks due to their modest computational complexity. Kim and Suthaharan (Kim and Suthaharan, 2004) proposed using entropy as a texture measure to identify more textured regions which are more suitable for embedding the watermark. Another texture measure is proposed by Kamble *et al.* (Kamble *et al.*, 2010) which is based on gray level covariance matrix. Yuan and Zhang (Yuan and Zhang, 2006) proposed using Gaussian Mixture Model (GMM) to model the texture and identify highly textured blocks for embedding. Yixin *et al.* (Yixin *et al.*, 2009) employed Just Noticed Distortion (JND) concept to identify the blocks which can be used for embedding without significant distortion compared to other blocks. Voloshynovsky *et al.* (Voloshynovskiy *et al.*, 1999) introduced Noise Visibility Function (NVF) which is based in Gaussian mixture model of texture based on HVS. Few authors to date have proposed using frequency based perceptual masks, where Li *et al.* (Li *et al.*, 2008) and Gao *et al.* (Gao *et al.*, 2006) proposed exploiting a modified Watson's visual model and Discrete Cosine Transform (DCT) for texture masking. The following paragraphs describe the masks considered in this thesis:

Texturization Masks

Entropy is a scalar value of grayscale image X_c of resolution $M_c \times N_c$ representing the energy of this image. Entropy is a statistical measure of randomness that can be used to characterize the texture of the whole image or blocks. For 8x8 blocks, the image X_c is divided into $\frac{M_c}{8} \times \frac{N_c}{8}$ blocks whose row and column indices defined as m_c , and n_c respectively. The entropy of a block b_i is defined as E_{b_i} :

$$E_{b_i} = \sum_{w,h \in b_i} SP_{(w,h)} \cdot \log_2 SP_{(w,h)} \quad (1.10)$$

where $SP_{(w,h)}$ is the normalized spectrum of 8×8 pixels blocks with w and h representing the index of pixels for width and height respectively. Blocks of higher entropy value E_{b_i} represents more textured blocks, smooth textured blocks have lower entropy E_{b_i} value. Entropy considers only one aspect of the human visual system, and does not exploit other features such as contrast and structure elements of the image.

Distortion Estimation (HVS) Masks

Noise Visibility Function (NVF) uses a general concept inspired by denoising. Voloshynovskiy *et al.* (Voloshynovskiy *et al.*, 1999) have proposed different functions for the definition of NVF based on the function of the image denoising and the statistical properties of the image. In the field of image processing, the variance is used to gain information about the local activity in the image. Its small values indicate smooth regions, and its large values indicate the presence of edges or highly textured regions. Using the local variance of pixels in block σ_x^2 , NVF can be defined as:

$$NVF = \frac{1}{1 + \sigma_x^2} \quad (1.11)$$

Visual models derived in compression techniques are perfectly adapted to the problem of watermarking. A common paradigm for perceptual masking is derived from the calculation of an image mask which depends on JND used in compression applications. Such a model can be

directly extended to digital watermarking applications by supplying information on the ability of inserting a block of the image with respect to another, this ability is related to the degree of noise that can undergo this block that ensures transparency while providing a robust watermarking. The combination of sensitivity, luminance, edges, and corners information are used to estimate a mask that defines the weight assigned to each block of the cover image. This weight provides a classification of blocks of the cover image to identify the more receptive blocks for embedding in terms of robustness and imperceptibility.

In this research the masks with lower complexity are considered in experimentation. The entropy mask is considered as an example of texturization masks, also NVF and JND masks are considered as an example of distortion estimation masks. The frequency based masks are not considered in this research to avoid computationally complex transforms for these masks.

In an earlier work by the authors (Rabil *et al.*, 2013a), Robustness Scores (RS) was proposed as an adaptive texture metric for grayscale images. RS is well suited for full uneven embedding scheme, where it groups image blocks according to their texture features and use different embedding capacity for each group of blocks. RS is calculated for different number of blocks groups using robustness fitness against JPEG. It is calculated for embedding 1 bit-per-block in all blocks belonging to the same group measured using NC defined in Section 1.1.

1.6 Bio-watermarking

Biometrics are the means to recognize individuals using intrinsic physical or behavioral characteristics. Many countries have become strong advocates of biometrics with the increase in fear of terrorism since September 11, 2001. Recently biometrics is synergistically merged into the digital watermarking technology to secure biometric captures against tampering, manipulation and attacks to ensure their authenticity and integrity, also to hide biometric traits invisibly inside other biometric images.

Offline signature based bio-watermarking systems have been studied by few authors in literature. Low *et al* (Low *et al.*, 2009) have used offline signature extracted features discretized as binary watermark to study the impact of watermarking attacks on watermark quality and ro-

bustness fitness only, while Bhattacharyya *et al* (Bhattacharyya *et al.*, 2007) have used offline signature images as watermarks to improve robustness against watermark removal attacks. The impact of watermarking attacks on the biometric verification system have been considered by Dong and Tan (Dong and Tan, 2008) for iris template images, and Huang *et al* (Huang *et al.*, 2008) for 3D face model authentication images. Also Dong and Tan (Dong and Tan, 2008) have considered the impact of watermarking attacks when embedding iris templates as watermarks. None of these bio-watermarking systems was based on computational intelligence techniques to optimize watermark embedding to maximize fitness of watermark quality and robustness.

In face recognition techniques, the recognition is performed using extracted features from facial captures, however the facial captures are stored for later feature extractions and improve classifiers to improve recognition rates. In some applications these captures are stored for the same individual at different ages to perform individual recognition across ages. These stored captures are vulnerable to manipulations, and thus it is essential to secure these facial captures to ensure their integrity and authenticity.

The main bio-watermarking application considered in this thesis is securing facial captures stored during enrollment using binary logo watermarks to ensure their authenticity and integrity. However there is another access control bio-watermarking application introduced in Annex II to embed another biometric trait inside the high-resolution facial captures in the passports to verify individuals crossing boards using two different biometric traits. Preferably combine a physical trait of facial captures with another behavioral biometric trait like offline signature. Offline signature can be captured from different customs forms during border crossing. This would improve significantly the individual verification for people crossing borders.

1.7 Discussion

The previous sections described the main aspects of intelligent watermarking for grayscale high-resolution images, and a literature survey on contributions proposed. Most of contributions proposed in literature rely on representing all cover image blocks in optimization candidate solutions in different EC methods. Watermark quality and robustness fitness are either

aggregated to be optimized as SOOP, or optimized simultaneously as MOOP. The embedding parameters to be optimized are dependent on the watermark embedding and extracting algorithm. In this thesis DCT-based algorithm (Shieh *et al.*, 2004) is used, and thus the embedding parameters maps to the selection of DCT coefficients to be modified to embed the watermark.

Authors in intelligent watermarking literature did not pay attention to using high-resolution grayscale images, and instead they tested their methods against low resolution test images (resolutions 512×512 and 256×256). The dimension of the search space of EC methods corresponding to optimizing embedding parameters of such low resolution images is limited. For high-resolution facial captures of resolution 2048×1536 pixels, the dimension of search space is equal to 49k variables represented by 300k bits using binary representation. Most of EC methods suffer from premature convergence in such dimension. This challenge is addressed in Chapter 3, where a specialized algorithm based on cooperative coevolution is proposed.

Also in intelligent watermarking literature, authors verified their methods using few test images, and did not consider handling streams of homogeneous grayscale images. For such homogeneous streams, the optimization problems corresponding to different images in this stream are similar. The optimization results of one image or sub-image can be reused for another image or sub-image in the stream of similar content. This challenge is addressed in Chapter 2, where optimization problems are proposed to be replaced by associative memory recalls. This associative memory is populated with optimization results for clusters of blocks according to their texture features.

CHAPTER 2

RAPID BLOCKWISE MULTI-RESOLUTION CLUSTERING (BMRC)

Population-based evolutionary computation (EC) is widely used to optimize embedding parameters in intelligent watermarking systems. Candidate solutions generated with these techniques allow finding optimal embedding parameters of all blocks of a cover image. However, using EC techniques for full optimization of a stream of high-resolution grayscale face images is very costly. In this chapter, a blockwise multi-resolution clustering (BMRC) framework is proposed to reduce this cost. During training phase, solutions obtained from multi-objective optimization of reference face images are stored in an associative memory. During generalization operations, embedding parameters of an input image are determined by searching for previously stored solutions of similar sub-problems in memory, thereby eliminating the need for full optimization for the whole face image. Solutions for sub-problems correspond to the most common embedding parameters for a cluster of blocks of similar texture features. BMRC identifies candidate block clusters used for embedding watermark bits using the robustness score metric. It measures the texture complexity of image block clusters and can thereby handle watermarks of different lengths. The proposed framework implements a multi-hypothesis approach by storing the optimization solutions according to different clustering resolutions and selecting the optimal resolution at the end of the watermarking process. Experimental results on the PUT face image database show a significant reduction in complexity up to 95.5% reduction in fitness evaluations compared with reference methods for a stream of 198 face images.

The content of this chapter is published in *Machine Vision and Applications* (Rabil *et al.*, 2013a) and *International Workshop on Intelligent Pattern Recognition and Applications, WIPRA 2013* (Rabil *et al.*, 2013b).

2.1 System Overview

The proposed technique is capable of finding blocks used for embedding and specify their embedding parameters in a computationally efficient manner. It also can handle watermarks of

different lengths using proposed Robustness Score (RS) metric for block clusters. This proposed metric is used also to identify the suitable embedding clusters of blocks. BMRC is based on a multi-objective formulation which satisfies the trade-off between watermark quality and robustness, and thus allows adaptability for different application domains, where the objectives priority vary, without the need for costly re-optimizations.

During the training phase, the multi-objective optimization results, obtained on training images, are stored in an associative Block Cluster Memory (BCM). After the full optimization results are obtained, the optimal solution is selected from the resulting Pareto front based on the application domain priorities. This optimal solution represents optimal embedding parameters for all training image 8×8 pixels blocks, it is used to collect the most frequent embedding parameters for all image blocks having the same texture. This information is stored for multi-resolution clustering of face image blocks based on their texture features, where clustering resolution represents the number of clusters. The order of embedding for these multi-resolution block clusterings is determined using the proposed RS metric. BMRC uses an incremental learning scheme in training phase, such that the multi-resolution clusterings and their corresponding most frequent embedding parameters are calculated for the first training images and get updated for subsequent training images.

During generalization phase, texture features are extracted from 8×8 pixels blocks of the unseen stream of images, then these blocks are categorized using the recalled multi-resolution clustering prototypes from BCM. The order of utilizing blocks categories for embedding is identified using RS, which is also used to calculate the empirical embedding capacity for these categories. The empirical embedding capacity is dependent on the watermark length and RS of block clusters identified in the image such that the block cluster of highest RS has the maximum embedding capacity, and it gets decremented until a threshold α of RS is reached. The watermark fitness is calculated for different resolutions stored in BCM, and then solutions are ranked to choose the optimal clustering resolution for each face in the stream.

Proof of concept simulations are performed using the PUT database (Kasinski *et al.*, 2008) of high-resolution face images, and compared against reference method in terms of complexity

and quality of solutions. Simulation results demonstrate that BMRC results in a significant reduction of the computational cost for IW by replacing costly optimization operations with associative memory recalls. The resulting solutions have nearly the same quality and robustness as those obtained with full optimization of each face image. The performance of BMRC is evaluated for different watermark length, and the robustness objective is considered of higher priority to reach up to 99.9% of bits restored after manipulating the watermarked face image. A sensitivity analysis is performed on BMRC tunable parameters to evaluate the impact of these parameters on both the framework performance and the associative memory size. Parallel implementation using Graphics Processing Units (GPU) is employed for the most complex functionalities of BMRC to evaluate its impact on the overall performance.

BMRC shown in Figure 2.1 finds optimal embedding bands in textured blocks for a stream of high-resolution face images using modest computational complexity. This is accomplished by replacing computational expensive full optimization with memory recalls from an associative memory representing prior optimization knowledge. Face images blocks are clustered according to their texture, and then optimal embedding bands for all of blocks of same texture are selected together using prior knowledge stored in associative Block Cluster Memory (BCM).

Solutions recalled from BCM, representing different number of clusters, are proposed to be ranked using watermark fitness to find number of blocks clusters for each face image. The number of clusters is referred to as clustering resolution in this chapter. This implements multi-hypothesis approach where all alternative solutions are stored and the hard decision to choose among these solutions is postponed.

The training set D consists of training face images defined as $D = \{TI_1, TI_2, \dots, TI_d, \dots, TI_N\}$, where TI_d represents face image of index d from the training set D of resolution $M_c \times N_c$. The number of face images in the training set is equal to N . For each training face image TI_d , the image is divided into 8×8 pixels blocks $B_d = \{b_i\}$, where $i = (n_c - 1) \times (M_c/8) + m_c$ with m_c , and n_c defines the row and column index of 8×8 blocks respectively. The total number of blocks $NB = (M_c/8) \times (N_c/8)$, and thus $i = [1, 2, \dots, NB]$.

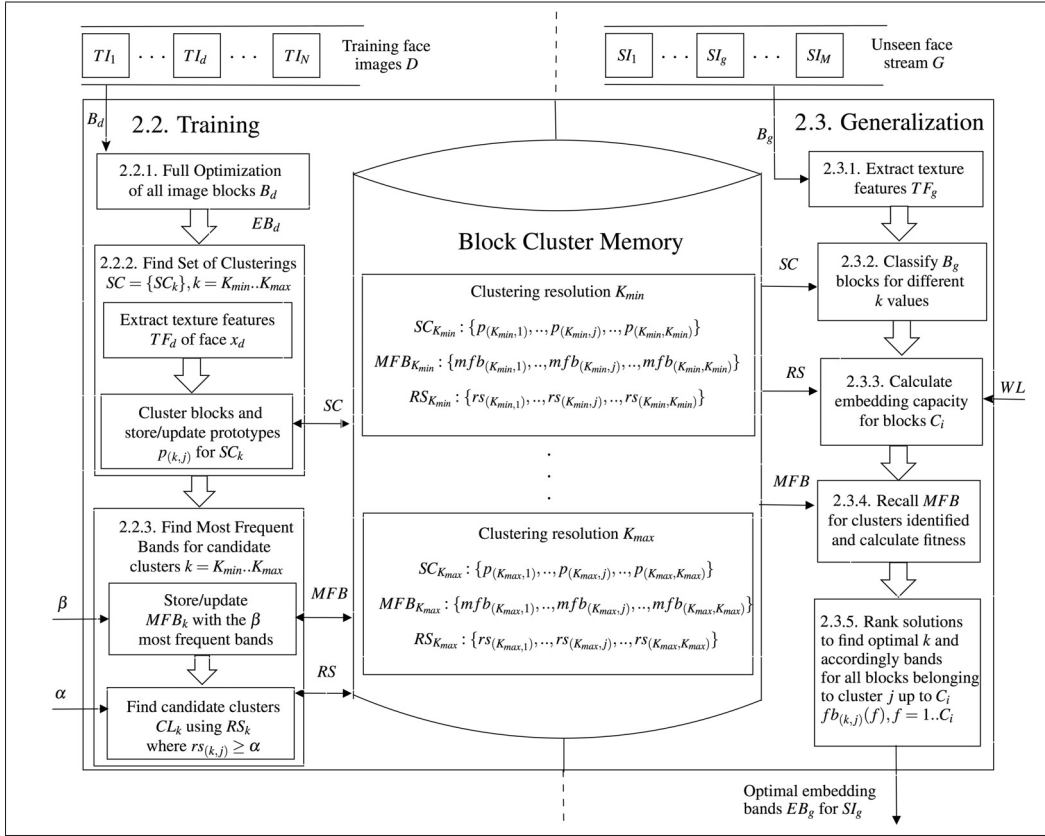


Figure 2.1 General overview of BMRC architecture and processing steps for training and generalization phases. BCM is organized based on clustering resolution k ranging from K_{min} to K_{max} .

Training face image blocks B_d are transformed into DCT domain $DCT_d = \{dct_i\}$, where $dct_i = \{ac_0, ac_1, \dots, ac_a, \dots, ac_{63}\}$ with ac_0 defines the DC coefficient of the 8×8 block b_i , and ac_a defines the a th DCT coefficient of the same block b_i . The texture features TF_d are extracted from DCT_d , where TF_d defines the most significant DCT coefficients from DCT_d for training face image TI_d . The texture feature vectors are defined as $TF_d = \{tf_i\}$, where tf_i defines the texture feature vector of block b_i . This feature vector is defined as $tf_i = \{ac_a\}$, where $a \in [0, 1, \dots, 63]$, and $a = \{a_1, a_2, \dots, a_t, \dots, a_T\}$. The number of coefficients used to extract features is equal to T , and t is the index of texture feature in the feature vector tf_i .

After the full optimization process for face image TI_d , the optimal embedding bands EB_d are concluded for face image TI_d , where $EB_d = \{eb_i\}$ with eb_i representing the optimal embedding bands for block b_i . The embedding bands defines the index of DCT coefficients which are

modified during embedding the watermark. It can be defined as $eb_i = \{a_1, a_2, \dots, a_e, \dots, a_{C_i}\}$ with e is the index of the embedding bands in eb_i , and C_i is the number of embedding bands for block b_i representing the embedding capacity for b_i .

The generalization set of unseen face image G is defined as $G = \{SI_1, SI_2, \dots, SI_g, \dots, SI_M\}$, where the size of generalization set equals to M . The subscript g is used instead of d for the data structures used in generalization phase. Thus B_g , DCT_g , and TF_g defines the 8×8 blocks, the DCT transformed blocks, and the texture features of the face image SI_g respectively.

Algorithm 1 describes the main steps of the proposed framework, where the training phase (lines 1-7) is performed on training set face image TI_d , and the generalization phase (lines 8-14) is triggered using generalization face image SI_g . BCM associative memory is populated (lines 1-7) with prior knowledge during two steps training phase, where the first step finds the multi-resolution clusterings for face image blocks (line 5), and the second step calculates the most frequent embedding bands associated with each and every cluster prototype defined in the first step (line 6). Prior knowledge are recalled for different resolutions k (lines 10-13) and fitness is calculated for these resolutions (line 12), and finally solutions are ranked (line 14) to take the hard decision at the end of the process.

Algorithm 1 Main steps of the BMRC framework.

Input: Either training face image TI_d , or unseen face stream SI_g

- 1: # TI_d is fed into the system and the training phase is called.
 - 2: Full optimization for all image blocks B_d represented positionally to find optimal embedding bands EB_d .
 - 3: Extract texture features TF_d from 8×8 pixels blocks B_d .
 - 4: **for** $k = K_{min} \rightarrow K_{max}$ **do**
 - 5: Store/update multi-resolution clustering SC_k prototypes using TF_d .
 - 6: Store/update most frequent embedding bands MFB_k for candidate clusters whose robustness scores ($rs_{(k,j)}$) equals or larger than α .
 - 7: **end for**
 - 8: # SI_g is fed into the system and the generalization phase is called.
 - 9: Extract texture features TF_g from 8×8 pixels blocks B_g .
 - 10: **for** $k = K_{min} \rightarrow K_{max}$ **do**
 - 11: Find clusters for face SI_g blocks using prototypes $p_{(k,j)}$ in SC_k .
 - 12: Recall most frequent bands MFB_k and calculate fitness.
 - 13: **end for**
 - 14: Rank solutions representing k values and select optimal k value.
-

The prior knowledge is represented by blockwise multi-resolution clustering of face image blocks b_i for different number of clusters k using texture feature vectors tf_i of these blocks b_i . The set of clusterings $SC: \{SC_{K_{min}}, SC_{K_{min}+1}, \dots, SC_k, \dots, SC_{K_{max}}\}$ are stored in associative memory. Each clustering SC_k consists of cluster prototypes $p_{(k,j)}$ with $j = 1, 2, \dots, k$ representing j th cluster for clustering resolution k , where $SC_k: \{p_{(k,1)}, p_{(k,2)}, \dots, p_{(k,j)}, \dots, p_{(k,k)}\}$. This set of clusterings SC are updated along training phase to update prototypes based on face images in the training dataset D .

The Most Frequent embedding Bands (MFB_k) for all blocks belonging to the same blocks cluster are calculated for training set D using previous optimization results. These results are represented by optimal embedding bands eb_i for all blocks b_i of training face image TI_d . For each clustering SC_k there is MFB_k set, where $MFB_k = \{mfb_{(k,1)}, mfb_{(k,2)}, \dots, mfb_{(k,j)}, \dots, mfb_{(k,k)}\}$. $mfb_{(k,j)}$ is associated with cluster prototype $p_{(k,j)}$ representing most frequent embedding bands using clustering resolution k for j th cluster. The set of most frequent bands is defined as $mfb_{(k,j)} = \{fb_{(k,j)}(1), fb_{(k,j)}(2), \dots, fb_{(k,j)}(f), \dots, fb_{(k,j)}(\beta)\}$ and $fb_{(k,j)}(f) \in [1, 2, \dots, 63]$. $mfb_{(k,j)}$ is ordered descendingly with respect to the frequency of occurrence of embedding bands, where $fb_{(k,j)}(1)$ is the index of the most frequent embedding band for j th cluster using resolution k represented by prototype $p_{(k,j)}$, and $fb_{(k,j)}(\beta)$ is the index of the least frequent band. The parameter β is tunable for the proposed system defining the size of $mfb_{(k,j)}$ representing the maximum number of frequent bands stored in BCM.

Robustness Scores is proposed as a texture metric for a cluster of blocks having similar texture features. This metric is calculated for a cluster of blocks using robustness fitness against JPEG compression with quality factor 80% measured using NC when embedding a watermark of length equal to the number of blocks in the cluster $nb_{(k,j)}$. The watermark bits are embedded only in blocks belonging to the cluster with capacity equals to 1 bit-per-block. Higher values of RS indicates higher texture clusters of blocks which are good candidates for embedding watermark bits. Blocks clusters of lower RS represent smooth textured clusters which should be avoided for embedding to improve both watermark quality and robustness. RS is calculated for all clustering SC_k for different clustering resolutions k during the training phase. Whenever

SC_k are updated during the training phase for subsequent training face image, RS is recalculated for the updated prototypes defined in SC_k .

RS are used to identify the order of embedding for different watermark length, and the embedding capacity of different blocks C_i . For each clustering SC_k there is a set RS_k where $RS_k: \{rs_{(k,1)}, rs_{(k,2)}, \dots, rs_{(k,j)}, \dots, rs_{(k,k)}\}$ with $j = 1, 2, \dots, k$ representing the scores for j th cluster using clustering resolution k . The clusters of blocks whose $rs_{(k,j_c)}$ is equal or higher than α threshold are considered candidate embedding clusters CL_k for resolution k , where $CL_k = \{j_1, j_2, \dots, j_c, \dots, j_{cl_k}\}$ and j_c defines the index of the clusters and the number of candidate embedding clusters equals to cl_k .

Even embedding would be a special mode of operation for the proposed system, where ranking clusters based on robustness scores $rs_{(k,j)}$ would not be needed during training phase. During generalization phase, the empirical embedding capacity calculation C_i would not be needed as well. The two phases now are presented in more details in the following sections.

2.2 Training

The processing steps included in training phase are described in the following sections and shown in Algorithm 2. Full optimization of training face image TI_d is performed (line 2), where all face image TI_d blocks b_i are represented positionally as eb_i in optimization candidate solution to find optimal embedding bands EB_d for all blocks of TI_d . After this optimization process, the prior optimization knowledge is concluded and stored in BCM. Texture features are extracted from face image TI_d blocks (line 3), then the prior knowledge is represented by most frequent embedding bands for all blocks having the same texture features (lines 5-7). This knowledge is stored for first training face image and gets updated with following training face images.

BCM holds this prior knowledge for different resolutions k ranging from K_{min} to K_{max} to decide the optimal number of clusters at the end of generalization phase of each face image based on the watermarking fitness. BCM is organized based on the value of clustering resolu-

Algorithm 2 Proposed BMRC training phase.

Input: Training dataset face images D , and empty BCM.

- 1: **for** $d = 1 \rightarrow N$ **do**
- 2: Perform mutli-objective optimization for all face TI_d blocks b_i where all blocks are represented positionally in the optimization candidate solution to find optimal embedding bands EB_d .
- 3: Extract texture features TF_d from face image TI_d
- 4: **for** $k = K_{min} \rightarrow K_{max}$ **do**
- 5: Store/update clustering SC_k including clusters prototypes (centroids) for clustering resolution k in BCM.
- 6: For each cluster $j = 1, 2, \dots, k$ of blocks store/update most frequent embedding bands MFB_k using maximum training capacity β .
- 7: Calculate robustness scores RS_k for the clusters, and identify candidate clusters CL_k for embedding as shown in Algorithm 3.
- 8: **end for**
- 9: **end for**

Output: Clustering SC_k with associated prototypes $p_{(k,j)}$, most frequent embedding bands MFB_k , robustness scores RS_k , and CL_k candidate embedding clusters stored in BCM.

tion k , where for each value of k the associated clusters prototypes $p_{(k,j)}$ are stored along with most frequent embedding bands $mfb_{(k,j)}$, and robustness scores $rs_{(k,j)}$ using prior knowledge of optimization process results.

2.2.1 Full Optimization

Each face image TI_d from the training set D is fed into multi-objective optimizer (Bureerat and Sriworamas, 2007). Multi-objective optimization is performed to find optimal embedding bands EB_d for all individual face image blocks satisfying the trade-off between the conflicting objectives of watermark quality and robustness. All individual face image blocks b_i are encoded positionally in the optimization candidate solutions such that all blocks of face image TI_d has one bit per block to be embedded. This full optimization process is computationally complex because of large dimension of the optimization search space due to representing all face image blocks in the optimization problem.

Multi-objective optimization results in multiple optimal solutions called non-dominated solutions, where improving one objective fitness results in suffering for other objective considered. Choosing the optimal solution among these solutions is based on the priority of objectives in

the application domain. This feature ensures the adaptability of the proposed system in different application domains without computationally expensive re-optimizations.

For example, quality is the most important issue with medical imaging applications where the watermarked image still goes through feature extraction process, on the other hand robustness is the most important issue with biometrics application where the embedded watermark represents biometric traits which are used for recognition after watermark bits extraction. In this research, we employ a fixed trade-off between robustness and quality by fixing quality requirements for optimal solution weighted PSNR at 42 dB (Voloshynovskiy *et al.*, 1999) which is considered acceptable from a Human Vision System (HVS) standpoint as shown in Figure 2.2 for training set face images TI_1 , TI_2 , and TI_3 .

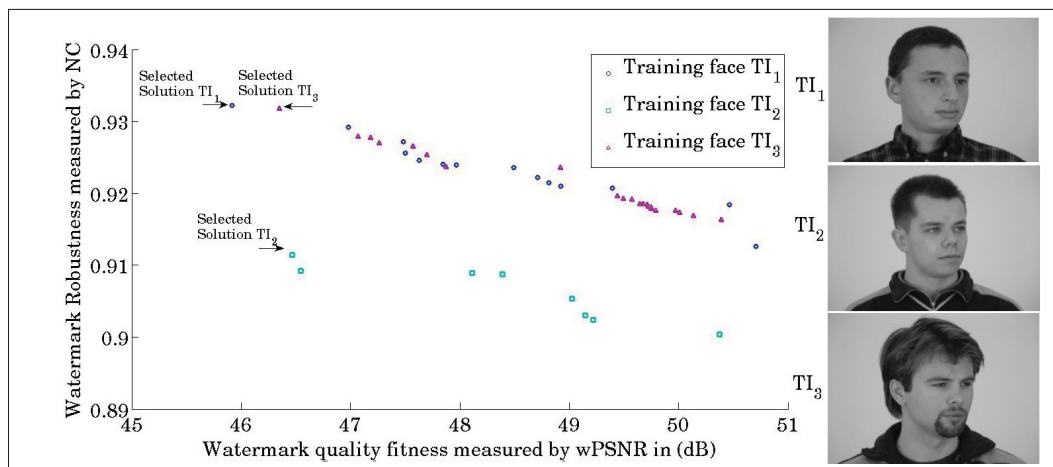


Figure 2.2 Selecting solution among Pareto front for training set face images TI_d from PUT database (Kasinski *et al.*, 2008) based on application domain objectives priorities, where embedding capacity is 8 bits per block.

2.2.2 Find Set of Clusterings SC

The first step of training phase involves extracting texture features from face TI_d blocks and cluster the blocks in texture feature space. The clustering technique is multi-resolution clustering approach where the clustering is performed using different number of clusters k ranging from K_{min} to K_{max} . The first training steps results in a set of clusterings SC including clus-

tering SC_k for each resolution k .

Extract texture features: After the full optimization process is over, optimization results obtained for all blocks with similar texture properties will provide the prior knowledge to decrease computational burden. Texture features can be extracted from spatial domain or transform domain for grayscale images. Although extracting texture features from spatial domain is less complex, in the proposed approach the transform domain texture features are already available for embedding and extracting the watermark. Texture feature vectors are extracted from the DCT transformed 8×8 blocks DCT_d of training face TI_d , where the most significant DCT coefficients are employed as texture features vectors tf_i for each block b_i . Using zoning approach proposed by Yu *et al.* (Yu *et al.*, 2010), texture feature vector for block b_i defined as $tf_i = \{ac_a\}$ where $a = \{a_1, a_2, \dots, a_t, \dots, a_T\}$ with $T = 39$ as shown in Figure 2.3.

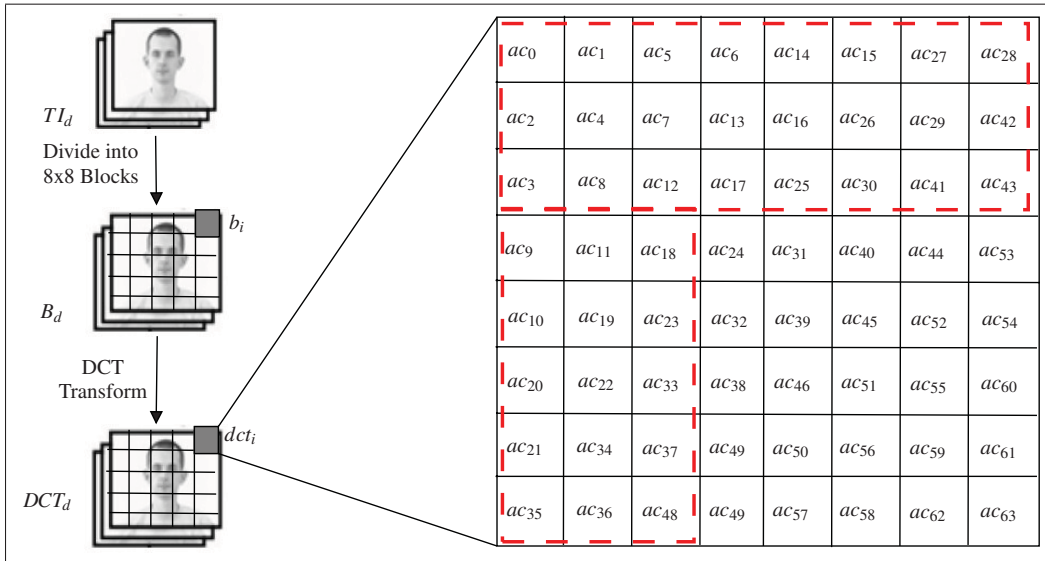


Figure 2.3 Zoning method to select the most significant DCT coefficients to extract texture features tf_i for block b_i (Yu *et al.*, 2010), $tf_i = \{ac_a\}$, where
 $a = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,$
 $16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 33, 34, 35, 36,$
 $37, 41, 42, 43, 48\}$

Store/update multi-resolution clustering SC_k : The blocks b_i of the training face image TI_d are clustered based on their texture feature vectors tf_i . The clustering is performed using multi-

resolution for different values of k . In this approach k different partitions of the texture space, ranging from K_{min} to K_{max} are computed. One of the least complexity category of clustering algorithms is k-means, this attracted lots of attention for authors in the large data clustering literature where it has time complexity of $O(nkm)$, and space complexity of $O(k)$ (Jain *et al.*, 1999), where k is the number of clusters, n is the number of patterns to be clustered, and m is the number of iterations. K-means is widely used in Content Based Image Retrieval (CBIR) (Yang *et al.*, 2010) with large data. K-means results in the prototypes of clusters represented by centroids in texture feature space.

The resulting centroids for the multi-resolution clustering represents the set of clusterings $SC = \{SC_k\}$ which are stored in BCM for the first training face image TI_1 , and the subsequent training images use these centroids as initial clustering and get updated along training for different SC_k . The optimal embedding clusters of blocks are those clusters whose $rs_{(k,j)}$ are higher than the threshold α , however some of these clusters can be ignored during embedding for watermarks of small lengths WL , and thus these are referred to as candidate embedding clusters. The order of embedding is utilizing robustness scores $rs_{(k,j)}$, where the clusters of higher $rs_{(k,j)}$ are used first for embedding until all watermark bits are embedded.

2.2.3 Find Most Frequent Bands for Candidate Clusters

The second step of training phase involves collecting embedding bands statistics for the set of clusterings SC using the full optimization results EB_d for all training face TI_d blocks. These embedding bands represent the AC coefficients in DCT domain 8×8 blocks that are altered to embed the watermark bits in these blocks. The most frequent bands $mf_{(k,j)}$ for each cluster j of blocks represented by the prototype $p_{(k,j)}$ using clustering resolution k are stored in BCM. Then robustness scores $rs_{(k,j)}$ for these clusters are calculated to identify order of embedding, and candidate embedding clusters based on robustness threshold α . The optimal embedding clusters of blocks are those clusters whose $rs_{(k,j)}$ are higher than the threshold α , however some of these clusters can be ignored during embedding for watermarks of small lengths WL , and thus these are referred to as candidate embedding clusters. The order of embedding is utilizing

robustness scores $rs_{(k,j)}$, where the clusters of higher $rs_{(k,j)}$ are used first for embedding until all watermark bits are embedded.

Store/update most frequent embedding bands MFB_k : For each clustering level k , the optimal embedding bands obtained from full optimization EB_d are collected for all blocks belonging to the same blocks' cluster. The most frequent embedding bands MFB_k for each candidate cluster of blocks are stored in BCM for TI_1 face image, and then for subsequent training images TI_d the most frequent embedding bands get updated.

Find candidate embedding clusters: After storing/updating multi-resolution clusterings SC , Robustness Scores $rs_{(k,j)}$ are evaluated using watermark robustness fitness of JPEG compression attack of quality factor 80% when only embedding 1 bit per block for all blocks belonging this cluster j . Robustness scores are calculated for all clustering separately such that $rs_{(k,j)}$ is calculated for j th cluster using clustering resolution k at a time using a random watermark W_r . The length WL of this random watermark equals to the number of blocks $nb_{(k,j)}$ belonging to the cluster represented by the prototype $p_{(k,j)}$ in the training image TI_d .

Algorithm 3 Identifying candidate embedding clusters.

Input: Clustering SC_k with cluster prototypes $p_{(k,j)}$, most frequent embedding bands $mfb_{(k,j)}$, and robustness scores threshold α .

- 1: **for** $k = K_{min} \rightarrow K_{max}$ **do**
- 2: Recall k cluster prototypes from clustering SC_k in BCM.
- 3: **for** $j = 1 \rightarrow k$ **do**
- 4: Count the number of blocks $nb_{(k,j)}$ belonging to cluster represented by prototype $p_{(k,j)}$, where $NB_k = \{nb_{(k,1)}, nb_{(k,2)}, \dots, nb_{(k,j)}, \dots, nb_{(k,k)}\}$.
- 5: Generate random binary watermark W_r with length $WL = nb_{(k,j)}$
- 6: Embed W_r in blocks of face image TI_d belonging to cluster represented by $p_{(k,j)}$ using recalled $fb_{(k,j)}(1)$ from $mfb_{(k,j)}$ yielding to 1 bit-per-block.
- 7: Calculate $rs_{(k,j)}$ using robustness fitness represented by NC.
- 8: Store/Update robustness fitness as $rs_{(k,j)}$ associated with prototype $p_{(k,j)}$ and create CL_k with indices j_c of clusters having $rs_{(k,j_c)} \geq \alpha$ descendingly.
- 9: **end for**
- 10: **end for**

Output: Robustness scores $rs_{(k,j)}$ associated with cluster prototypes $p_{(k,j)}$, and ordered list of the indices of the candidate embedding clusters CL_k

The detailed algorithm to identify candidate embedding clusters is shown in Algorithm 3. Robustness scores evaluation starts with generating random binary watermark W_r of length $nb_{(k,j)}$ for j th cluster using resolution k (lines 4-5). The watermark W_r bits are embedded in the blocks belonging to the cluster identified by prototype $p_{(k,j)}$ (line 6), and watermark robustness fitness against JPEG compression with quality factor 80% is calculated using NC to represent robustness score $rs_{(k,j)}$ for this cluster and stored in BCM (line 7). The indices for those clusters of scores higher than or equal to α in CL_k (line 8). This process is performed on the first training face image TI_1 and gets updated for following training face images TI_d .

Clusters of $rs_{(k,j_c)}$ equals or higher than α are considered candidate clusters CL_k for embedding using resolution k . CL_k includes the indices of these clusters $CL_k = \{j_1, j_2, \dots, j_c, \dots, j_{cl_k}\}$, with cl_k equals to the number of candidate embedding clusters. The order of embedding watermark bits is dependent on the value of $rs_{(k,j_c)}$ for blocks, such that the cluster of highest $rs_{(k,j_c)}$ is used first for embedding and then clusters of lower scores follow descendingly.

Figure 2.4 shows an example using resolution $k = 4$, where 4 random watermarks W_r are generated of lengths $WL = \{nb_{(4,1)}, nb_{(4,2)}, nb_{(4,3)}, nb_{(4,4)}\}$ and embedded one at a time in blocks belonging to cluster j using recalled most frequent band $fb_{(4,j)}(1)$ from $mf_{(4,j)}$. Watermark robustness against JPEG compression is measured using NC for all clusters j to calculate $rs_{(4,j)}$. The scores calculated show that the highest scores are found for edges textured blocks 0.99 and the lowest 0.65 for background smooth textured blocks. Using the robustness score threshold α ensures embedding in textured blocks and avoiding smooth background blocks. Within the textured blocks the clusters of blocks are ranked based on $rs_{(k,j)}$ to use the highest rs first for embedding, and descendingly use the clusters of lower rs . Thus CL_4 would contain indices of clusters 1, 2, 4 ordered for embedding.

Figure 2.5 shows the order of embedding for different message length for resolutions $k = 11$ and 13, where the white pixels represents the embedding blocks belonging to candidate embedding clusters CL_k for watermarks of length WL equals to 1.6k, 5k, 7k, and 10k. The embedding blocks are chosen based on the $rs_{(k,j)}$ of the cluster to which the block belongs to.

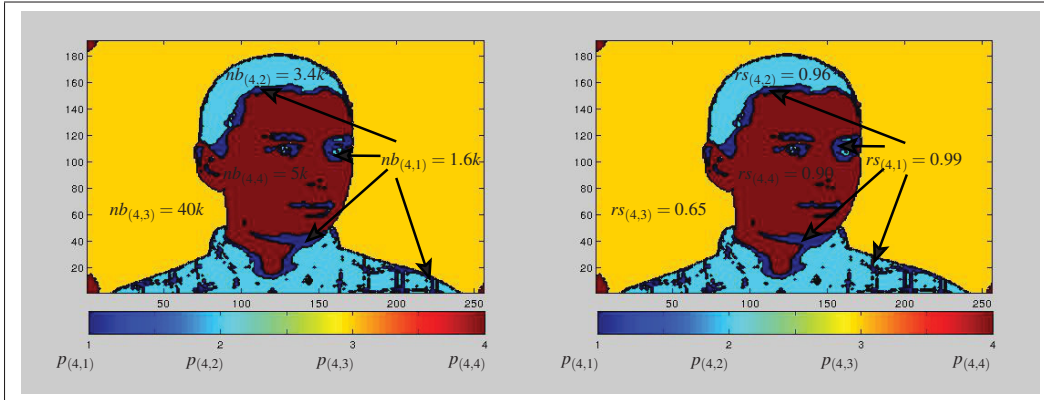


Figure 2.4 Robustness scores $rs_{(k,j)}$ calculation example for $k = 4$, where $SC_4 = \{p_{(4,1)}, p_{(4,2)}, p_{(4,3)}, p_{(4,4)}\}$, $NB_4 = \{1.6k, 3.4k, 40k, 5k\}$, $RS_4 = \{0.99, 0.96, 0.65, 0.90\}$, and $CL_4 = \{1, 2, 4\}$

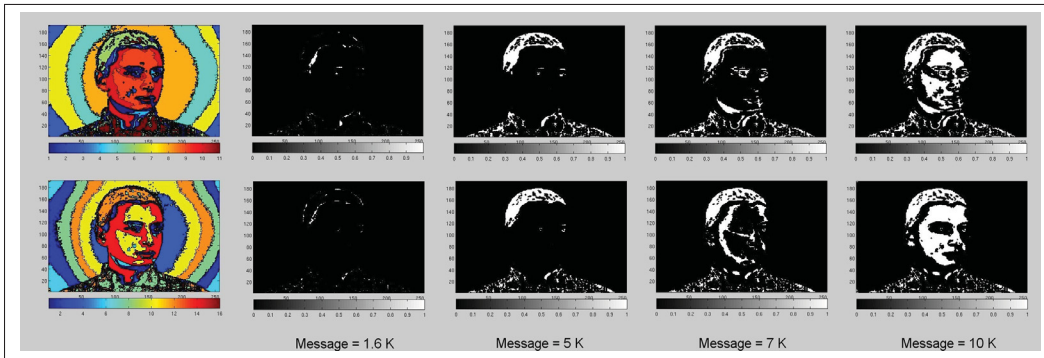


Figure 2.5 Embedding blocks belonging to CL_k of clustering resolutions $k=11,16$ for different WL on face image from PUT database (Kasinski *et al.*, 2008).

For shorter WL only edges clusters of blocks are used for embedding, meanwhile less textured clusters are used for longer WL until all bits are allocated.

By the end of the training phase, the associative memory BCM is populated with prior optimization knowledge. The prior knowledge should cover all clustering resolutions k , because it is infeasible to predict the number of clusters discovered in face images in the stream with different light conditions and different textured clothes, nor the number of blocks belonging to each texture cluster. The proposed associative memory holds the knowledge for different number of clusters k , and the decision of the optimal number of clusters k is postponed to generalization phase. The associative memory is organized based on value of k , where for each

value of k the relevant cluster prototypes $p_{(k,j)}$ are stored along with most frequent embedding bands $mfb_{(k,j)}$ for the candidate embedding clusters whose indices are included CL_k .

Algorithm 4 Proposed BMRC generalization phase.

Input: Clustering SC_k with cluster prototypes $p_{(k,j)}$, most frequent embedding bands $mfb_{(k,j)}$, and ordered candidate embedding clusters CL_k

- 1: **for** $g = 1 \rightarrow M$ **do**
- 2: Extract texture features TF_g from stream face image SI_g .
- 3: **for** $k = K_{min} \rightarrow K_{max}$ **do**
- 4: Find SI_g blocks clusters in texture feature space TF_g using clusterings prototypes SC_k from BCM.
- 5: **if** Uneven embedding is used **then**
- 6: Identify face image SI_g blocks belonging to CL_k .
- 7: Calculate embedding capacity C for SI_g blocks using Algorithm 5.
- 8: **end if**
- 9: Recall $mfb_{(k,j)}$ from BCM for clusters in face image SI_g and calculate watermark quality and robustness fitness for value of k .
- 10: **if** Solutions are good enough with respect to application domain **then**
- 11: Break and find optimal embedding bands using the current value of k .
- 12: **end if**
- 13: **end for**
- 14: Rank solutions representing different k values and select the suitable value for k using threshold based on human vision as shown in Section 2.2.1
- 15: **end for**

Output: Optimal embedding bands EB_g and capacity C for stream images SI_g

2.3 Generalization

The generalization phase processing steps are shown in Algorithm 4. Unseen set of face images $\mathbf{G}=\{SI_1, SI_2, \dots, SI_g, \dots, SI_M\}$ utilize the prior knowledge stored in BCM to find optimal embedding blocks and bands. Blocks b_i of SI_g image are compared against clusterings prototypes based on texture feature vectors tf_i for different resolutions k using recalled clustering SC_k from BCM (line 2-4). The blocks b_i belonging to candidate embedding clusters CL_k are identified and empirical capacity C_i is calculated for these blocks using recalled $rs_{(k,j)}$ and CL_k from BCM (line 6-7), these steps (lines 6-7) are not required in even embedding scheme with equal embedding capacity in all cover image blocks.

Based on the capacity, the most frequent embedding bands $mf_{b(k,j)}$ are recalled from BCM, then used for embedding to calculate watermarking fitness for quality and robustness (line 9). If an acceptable solution is reached using resolution k , there would be no need to calculate fitness for rest of resolutions (line 10-12) to reduce the complexity of fitness evaluations. For example, if robustness of $NC = 1$ is reached in application domains with high priority for robustness, then the fitness evaluations corresponding to other resolutions k are not required. Finally solutions are ranked using watermark fitness and the optimal resolution k for each face is concluded (line 14).

2.3.1 Extract Texture Features

Texture features are extracted from all face image blocks to be able to group these blocks based on their texture, where the face image SI_g is divided into 8×8 pixels blocks B_g , then the blocks are transformed into DCT domain DCT_g and the most significant DCT coefficients of each block b_i are used as feature vectors tf_i for the face image blocks.

2.3.2 Find Image Blocks Clusters for Different k Values

After extracting texture features, face image SI_g blocks are compared against cluster centroids recalled from BCM for different resolutions k . Each block b_i is associated with the nearest centroid for each value of k in texture space. As shown in Figure 2.6, the block is compared to the recalled centroids for multi-resolution clustering k in texture feature space. In this example, the face image block b_i is associated with clusters 1, 3, 2, and 6 for number of clusters k equals to 3, 4, 5, and 6 respectively in texture feature space.

2.3.3 Calculate Embedding Capacity for Different Blocks C_i

For uneven embedding scheme, candidate embedding clusters based on Robustness Scores RS_k are recalled from BCM. These candidate clusters are ordered such that the cluster with highest $rs_{(k,j)}$ is used first and then less robustness scores clusters are used next for embedding. Classifying face image blocks into foreground and background is not efficient, as some of the foreground blocks have better robustness than others. These can be utilized for larger embedding capacities and other foreground blocks should be avoided during embedding as shown

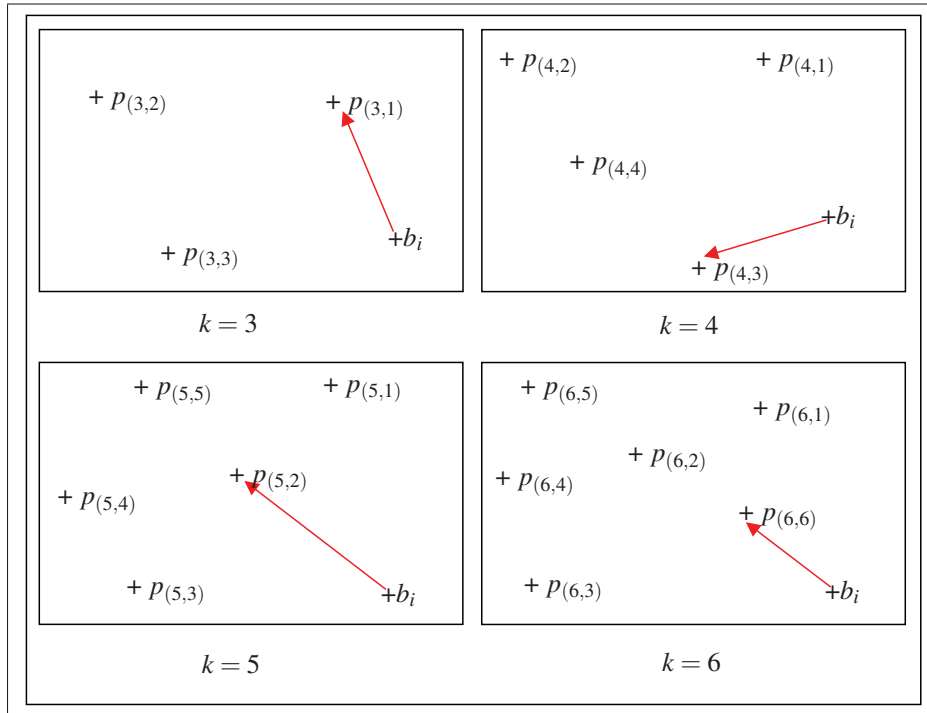


Figure 2.6 Classifying block b_i for different k values in texture feature space.

in Figure 2.5 for smaller watermarks. Ordering candidate embedding clusters also ensures adaptability for different lengths watermarks.

The proposed algorithm to calculate the embedding capacity C for different blocks $C=\{C_1, C_2, \dots, C_i, \dots, C_{NB}\}$ using the ordered list of indices of candidate embedding clusters $CL_k=\{j_1, j_2, \dots, j_c, \dots, j_{cl_k}\}$ is demonstrated in Algorithm 5. In this algorithm the bits are allocated without real embedding, such that the bits allocated to each cluster of blocks are equal representing empirical embedding capacity for all blocks belonging to the same cluster. CL_k is ordered such that the first candidate cluster $c = 1$ is the cluster with highest robustness score $rs_{(k,j)}$, and cl_k is the number of clusters whose $rs_{(k,j)}$ is equal to α or higher using resolution k . Initially maximum embedding capacity $C_{Max}(g)$ for face image SI_g equals to 1 (line 1). The maximum capacity $C_{Max}(g)$ is incremented (line 16) until all watermark bits WL are distributed among blocks b_i belonging to the top candidate clusters (line 2), such that the embedding capacity for the first candidate cluster in list holds the maximum embedding capacity, and this capacity is decremented for the second candidate and the third descendingly (line 7).

Algorithm 5 Calculating watermark capacity for different blocks.

Input: Ordered list of candidate embedding clusters CL_k of size cl_k , watermark length WL , and face image SI_g with NB blocks.

```

1:  $C_{Max}(g) = 1$ 
2: while  $sum(C) < WL$  do
3:   for  $i=1$  to  $B$  do
4:     for  $c=1$  to  $cl_k$  do
5:       if Block  $b_i$  belongs to cluster  $j_c$  of  $CL_k$  then
6:         if  $C_{Max}(g) - c + 1 > 0$  then
7:            $C_i = C_{Max}(g) - c + 1$ 
8:         else
9:            $C_i = 0$ 
10:        end if
11:       else
12:          $C_i = 0$ 
13:       end if
14:     end for
15:   end for
16:    $C_{Max}(g) = C_{Max}(g) + 1$ 
17: end while

```

Output: C identifying embedding capacities for face image SI_g blocks, and $C_{Max}(g)$ representing the maximum embedding capacity for this face image.

2.3.4 Recall MFB_k for Clusters Found and Calculate Fitness

The most frequent embedding bands $mfb_{(k,j)}$ associated with prototypes $p_{(k,j)}$ of clusters found in face image SI_g are recalled from BCM. These most frequent embedding bands are used as optimal embedding bands for all face image blocks belonging to the same cluster of blocks based on their texture features. This recall is performed for different resolutions k to find optimal embedding bands for all blocks, then watermark quality and robustness fitness are calculated using these bands.

As shown in Figure 2.7 for $k = 3$, each block b_i is classified using the distance to the prototypes $p_{(k,j)}$ of defined blocks clusters in texture feature space, then $mfb_{(k,j)}$ associated with $p_{(k,j)}$ are recalled. The embedding capacity C_i of b_i decides how many embedding bands $fb_{(k,j)}(f)$ are selected from $mfb_{(k,j)}$ to be used for embedding the watermark, such that if the capacity is equal to 1 bit-per-block, only the embedding band $fb_{(k,j)}(1)$ is selected for embedding, on the

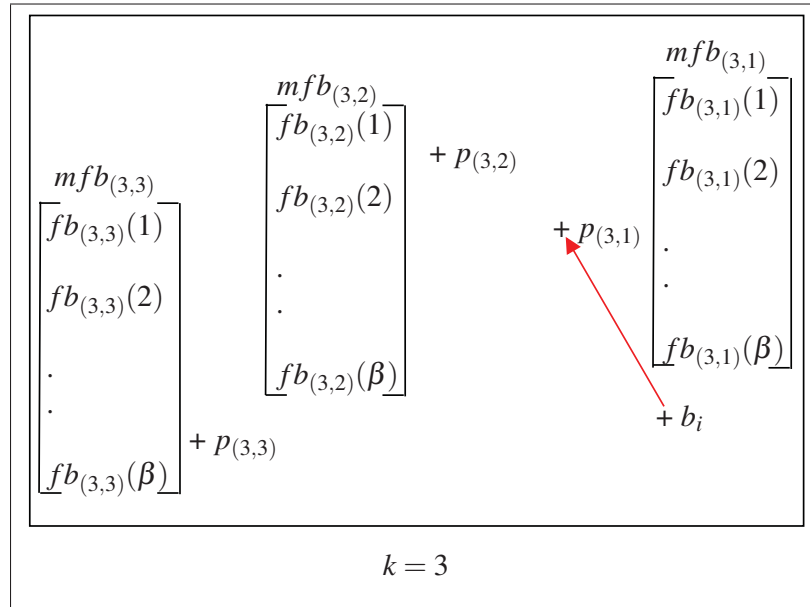


Figure 2.7 Classifying block b_i in texture feature space and recall $mfb_{(k,j)}$ associated with prototypes $p_{(k,j)}$ for $k = 3$, where $SC_3 = \{p_{(3,1)}, p_{(3,2)}, p_{(3,3)}\}$, and $MFB_3 = \{mfb_{(3,1)}, mfb_{(3,2)}, mfb_{(3,3)}\}$

other hand if the capacity is equal to 3 bits-per-block, then the bands $fb_{(k,j)}(1)$, $fb_{(k,j)}(2)$, and $fb_{(k,j)}(3)$ are selected.

2.3.5 Rank Solutions According to Fitness to Find Value of k

Using the watermark fitness calculated for different values of k , solutions are proposed to be ranked to find the optimal number of cluster k to accomplish maximum watermark fitness. Ranking solutions based on two conflicting objectives like watermark quality and robustness would involve decision based on the priorities of the application domain. In this research we employ the same criteria of selecting solutions from full optimization module in training phase at Section 2.2.1. Thus, the number of clusters is determined using application specific metrics rather than traditional clustering metrics like cluster accuracy (ACC) (Nie *et al.*, 2011).

2.4 Experimental Methodology

The database for face images used in experiments is proposed to be PUT (Kasinski *et al.*, 2008) face database which consists of 100 individuals with 100 poses for each individual. Color face images of resolution 2048×1536 are converted to grayscale level. Using the first pose of each individual (face images of name pattern IIII1001.JPG where IIII is the individual number in 4 digits), the first 40 face images are used for verification, the next 10 individuals face images for training with full optimization to populate associative memory, and finally the last 40 face images is used as small testing set. Another larger testing set which consists of 100 for the last poses is used for extended testing for the proposed system (face images of name pattern IIII4001.JPG and IIII4002.JPG where IIII is the individual number in 4 digits). This large testing set consists of 198 face images using the poses 4001 and 4002.

The training set and verification set is used for system design and parameters tuning, and both testing sets are used to test the proposed system. It is not feasible computationally to run the baseline system on the larger testing set due to the huge complexity of the baseline system with full optimization for all positional blocks. Assuming 40 optimization iterations, the baseline system would take years to handle stream of 100 face images using higher embedding capacities.



Figure 2.8 BancTec binary logo used as watermark to be embedded in face images.

The watermark to be embedded is BancTec binary logo with different resolutions shown in Figure 2.8. The watermark embedding/extracting algorithm used in experiments is an algo-

rithm proposed by Shieh *et al* (Shieh *et al.*, 2004) as illustrated in Section 1.2. The metrics used in experimentation for measuring watermark quality and robustness are wPSNR and NC respectively as defined in Section 1.1. Only robustness against JPEG compression of quality factor 80% is considered in experimentation, the impact of attack intensity has been addressed before (Rabil *et al.*, 2011a).

The complexity reduction achieved using the proposed BMRC is measured using the number of fitness evaluations, where the fitness evaluation represents the most complex process for EC based optimization. This complexity measure is more accurate than using traditional CPU time to avoid effects resulting from server load and memory usage. The complexity reduction is measured for a stream of high-resolution facial images to assess the overall reduction for the whole IW process of this stream

The first experiment compares the proposed system with the baseline system representing traditional methods with full optimization for all face image blocks. The optimization for baseline system is based on multi-objective PBIL proposed by Bureerat and Sriwaramas (Bureerat and Sriwaramas, 2007), with maximum iterations set to 40 iterations, and population size equals to 24. The size of external archive is set to 20. The performance of the baseline system using multi-objective PBIL is compared to traditional Multi-Objective Genetic Algorithm (Deb, 2001) (MOGA) using the same settings for both except for initialization. In multi-objective PBIL probability vectors are all initialized to 0.5 for all bits, while for MOGA initial bands are set to random embedding bands $eb_i \in [1, 63]$. Out of the resultant Pareto front, one solution is selected based on HVS criteria. Using this criteria, the solution of highest robustness with quality fitness equals to 42 dB measured using wPSNR metric is selected as described in Section 2.2.1. In this experiment, the minimum number of clusters K_{min} is set to 3 and maximum number K_{max} is set to 40. The maximum training capacity β is set to 20 bits-per-block, and the threshold for RS α is set to 89%.

The second experiment evaluates the performance of the proposed system on validation set with uneven and even embedding as special mode of operation for the proposed system. In even embedding, embedding capacity calculation is not needed, nor ranking clusters of blocks

using robustness scores. The quality of solutions produced using even and uneven embedding are compared against baseline system, and the time complexity is compared in both cases to evaluate the computational complexity reduction using the proposed system. The watermark length is 48.8 k-bits yielding to embedding capacity equals to 1 bit per block for even embedding scheme. The performance of BMRC with uneven scheme is analyzed on sample face images with different fitness levels.

In the third experiment, the performance of the proposed system for watermarks of different lengths is evaluated starting of the smallest logo whose resolution is 30×30 yielding to around 0.9 k-bits up to largest logo of resolution 221×221 yielding to 48.8 k-bits which is compared against the baseline system using Shieh method (Shieh *et al.*, 2004) using even embedding with equal embedding capacity equals to 1 bit per block. The training set for this experiment consists of one face image $N = 1$. From this experiment the optimal watermark length for using biometric traits to be embedded is concluded. Sensitivity analysis is performed for tunable parameters α and β for the proposed system. The functionalities of highest complexity are migrated into parallel implementation using GPU to evaluate its impact on the resulting complexity reduction.

All previous experiments were using training set of size $N = 1$. The fourth experiment focuses on the impact of training set size to populate the associative memory BCM on the quality of solutions produced and the training time for the associative memory. Training set of sizes 1, 2, 3 and 7 face images are used to populate the BCM associative memory for different embedding capacities 1, 2, 4, and 8 bits per block for the training set. One face image represents training set of 49,152 blocks of different textures and their relevant optimal bands.

All previous groups of experiments are performed on the verification set for system development, the fifth experiment uses two different testing sets to measure the performance of the proposed system on unseen face images. The first testing set is smaller set which consists of 40 face images, and the second larger testing set consists of 198 face images to measure the performance of the proposed system on larger dataset.

The experiments are executed on gentoo linux based server of 24 GB memory size and 8 cores Intel Xeon CPU E5520 of speed 2.27GHz. And for GPU implementation, experiments are executed using 8 NVIDIA Tesla C2050/C2070 GPU cards installed on gentoo linux based server of 24 GB memory size as well.

2.5 Experimental Results

Table 2.1 Computational complexity of the proposed system using even and uneven scheme for a stream of 40 face images using training set size $N = 1$, measured in total CPU time for the stream.

CPU Time	Training in k sec.			Generalization in k sec.		
	T_{optim}	T_{SC+} T_{MFB}	T_{RS}	$T_{classify}$	$T_{capacity}$	$T_{fitness}$
Even Scheme	1160	4	N/A	10	N/A	570
Uneven Scheme	1160	4	1	10	1	570

Table 2.1 shows the complexity of the proposed system with even and uneven embedding scheme. Time complexity is measured using CPU k-sec, where in the training phase T_{optim} represents the full optimization for one training face image $N = 1$, T_{SC} represents total time to find SC_k for k ranging from K_{min} to K_{max} , T_{MFB} represents total time to find most frequent bands MFB_k , and T_{RS} represents the total time to calculate robustness scores RS_k . In the generalization phase, time complexity is measured for a stream of 40 face images, where $T_{classify}$ represents total time to classify face images blocks into k clusters using recalled SC_k from BCM. $T_{capacity}$ represents the total time to calculate the embedding capacity C based on RS_k and watermark length, and $T_{fitness}$ represents the time to calculate both quality and robustness fitness for range of k values. The time complexity of watermarking one face image is 1160 k sec for the baseline with full optimization, compared to an average of 14.5 k sec for the proposed BMRC during generalization for a stream of 40 face images. Also results show minimal impact of using uneven embedding scheme on the proposed system complexity.

Table 2.2 shows the complexity reduction measured by number of fitness evaluation, where results show significant reduction of 93.5% when considering the training fitness evaluations

Table 2.2 Computational complexity reduction of the proposed BMRC system compared to baseline system and MOGA (Deb, 2001) for a stream of 40 face images using training set size $N = 1$, measured in number of fitness evaluations for the stream.

Fitness Evaluation	Baseline & MOGA (Deb, 2001)	Proposed BMRC
Training	N/A	960
Generalization	38400	1520
Complexity Reduction		
Generalization only	96.0%	
Overall Reduction	93.5%	

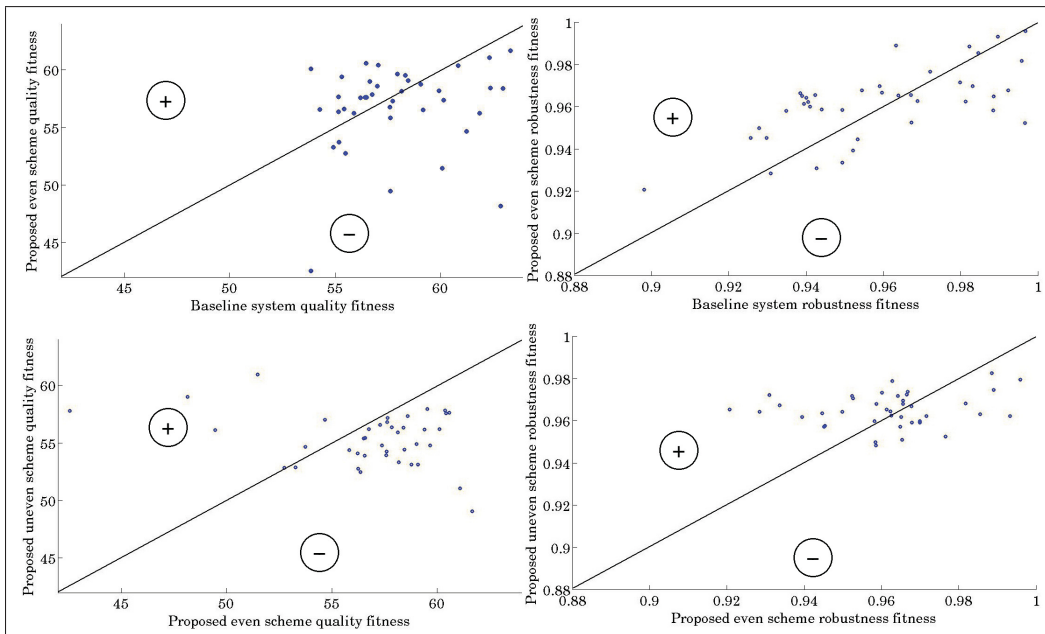


Figure 2.9 Fitness comparison of the proposed system with even and uneven scheme compared to baseline system.

with generalization for a stream of 40 face images. This complexity reduction is increased up to 95.5% for larger streams of 198 face images. The fitness evaluations of the baseline and MOGA represent the fitness evaluations during the full optimization. The fitness evaluations for the proposed BMRC represent the fitness evaluations of the optimization during the training phase, and fitness evaluations for different number of clusters k to select the optimal value of k during the generalization phase. The fitness comparison of the development set of face images is shown in Figure 2.9. The fitness for both watermark quality and robustness of the

Table 2.3 Mean fitness for validation set of face images for MOGA (Deb, 2001) and baseline system compared to proposed system with even and uneven scheme using watermark of length $WL = 48.8k$.

Mean Fitness	Quality	Robustness
MOGA(Deb, 2001)	58.41 ± 3.35	0.9393 ± 0.0061
Baseline	57.93 ± 2.71	0.9589 ± 0.0240
Even scheme	56.79 ± 3.76	0.9617 ± 0.0171
Uneven scheme	55.31 ± 2.24	0.9651 ± 0.0087

proposed BMRC are comparable to the baseline system with full optimization for the development stream of face images. Then the proposed BMRC with even embedding scheme is compared against uneven embedding scheme for the same development set of face images.

Table 2.3 shows mean fitness for validation set where there is improvement in robustness fitness and slight degradation in quality fitness within the acceptable quality according to HVS using highly reduced computational resources. The baseline system based on PBIL has better robustness fitness than using MOGA, because of the intrinsic property of PBIL of having probability vectors. This helps to improve the convergence properties compared to traditional GA. The complexity of MOGA is equal to the complexity of multi-objective PBIL as shown in Table 2.2, where the fitness evaluations of both methods are equal for the same population size and number of generations.

The proposed BMRC is based on guided search technique, compared to global search in traditional methods with large search space. The guided search in BMRC finds blocks of similar texture and find their optimal embedding bands for them together. The main complexity reduction for the proposed BMRC is resulting from the similarity between training face images and the generalization face images. This similarity is a local similarity on block level rather than global similarity between face images. Two face images are considered similar if they include blocks of the same texture, and thus the optimal embedding bands can be reused for these similar images. For homogeneous streams of face images considered in this chapter, the clusters of blocks are expected to be of similar texture. Choosing the optimal number of clusters in

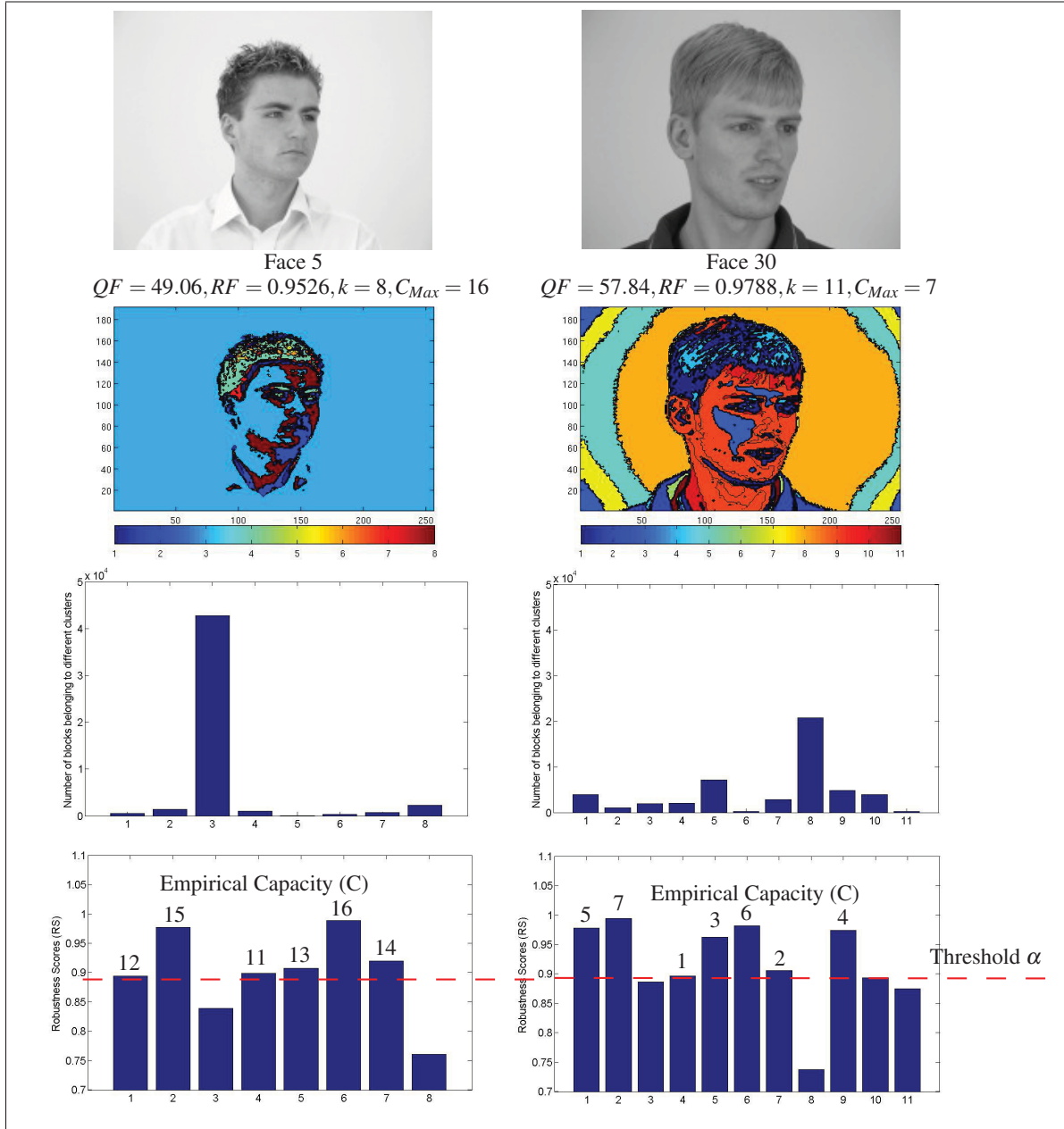


Figure 2.10 The impact of variations of the number of blocks belonging to similarly textured clusters on the proposed BMRC performance using uneven scheme.

BMRC using ranking solutions ensures the adaptability of the system to reach the maximum similarity among clusters of blocks.

Figure 2.10 shows two face images with different fitness levels, due to varying number of blocks belonging to similarly textured clusters. A histogram of the number of blocks belong-

ing to each cluster, and their relevant robustness scores RS for these two face images are shown in the figure. The robustness scores threshold is set to $\alpha = 0.89$, such that clusters of $RS < \alpha$ are excluded from embedding. Empirical embedding capacities C are calculated using Algorithm 5. The maximum empirical capacity C_{Max} of a blocks cluster is incremented until all watermark bits are distributed among the blocks belonging to the same cluster.

For face image 5, the number of blocks belonging to textured clusters is small. The smooth textured blocks represent around 90% of blocks of this face image shown by cluster 3 of blocks whose robustness score is less than α . The watermark bits are distributed among the few blocks belonging to textured clusters with larger empirical embedding capacities. The maximum capacity for blocks cluster is equal to 16 bits-per-block. This degrades the resulting fitness for both quality and robustness. On the other face image 30, the blocks belonging to textured clusters are large, and thus the empirical embedding capacities is equal to 7 bits-per-block. The lower embedding capacities result in better fitness. This concludes that BMRC performance is more sensitive to variations of number of blocks belonging to similarly textured clusters, rather than similarity between these clusters of blocks.

Table 2.4 Proposed system performance using uneven scheme with different watermark length.

WL	Quality	Robustness	k	$C_{Max}(g)$	$Max(C_{Max}(g))$
0.9 k	65.12±3.23	0.9997±0.0013	8±3.35	2±0.40	3
1.6 k	63.46±3.94	0.9985±0.0035	9±5.18	2±0.83	7
2.5 k	62.51±3.11	0.9964±0.0040	12±8.10	3±1.29	8
3.6 k	61.08±3.01	0.9931±0.0056	9±7.16	3±0.91	5
6.4 k	60.75±3.47	0.9885±0.0078	11±8.85	4±1.54	7
10.0 k	59.05±2.41	0.9844±0.0074	14±9.40	5±2.48	13
14.4 k	57.98±2.51	0.9796±0.0077	11±7.86	5±2.80	18
22.5 k	57.71±2.08	0.9735±0.0072	9±6.74	6±2.22	16
32.4 k	56.64±2.24	0.9702±0.0074	13±8.70	7±2.92	18
48.8 k	55.43±2.24	0.9651±0.0087	11±7.87	8±3.14	16
Baseline	57.93±2.71	0.9589±0.0240	N/A	N/A	N/A
MOGA (Deb, 2001)	58.41±3.35	0.9393±0.0061	N/A	N/A	N/A

Table 2.4 shows the performance of the proposed system for different length of the watermark WL . The performance is represented using mean values of quality and robustness fitness for validation set of face images along with the maximum embedding capacity $C_{Max}(g)$ for face image SI_g , and optimal number of clusters k as shown in Figure 2.12. Traditional approaches represented by baseline system, and MOGA are shown for watermark length $WL = 48.8k$ only due to the formulation of these traditional methods. The extracted watermarks for different WL using the proposed system are shown in Figure 2.11. The watermark fitness including quality and robustness is improved with watermarks of smaller lengths WL .











Original Watermark					
Extracted Watermark					
Watermark resolution	221x221	150x150	100x100	60x60	40x40
WL	48.8k	22.5k	10k	3.6k	1.6k
NC	0.9651	0.9735	0.9844	0.9931	0.9985

Figure 2.11 Extracted watermarks of different length WL using the proposed system with uneven scheme.

Figure 2.12 shows that the optimal message length for the best robustness is 1.6 k bits, where the average robustness for a stream of 40 face images is over 99.9%. This message length ensures the highest robustness for such high-resolution grayscale face images. Also in Figure 2.12, the number of clusters k is fluctuating reflecting the variance of k along the stream of validation set of face images. The time complexity for fitness evaluation $T_{fitness}$ is 190 k seconds for 506 fitness evaluations compared to 570 k seconds for 1520 fitness evaluations as shown on Tables 2.1 and 2.2, where the fitness evaluation is not needed once a robustness fitness

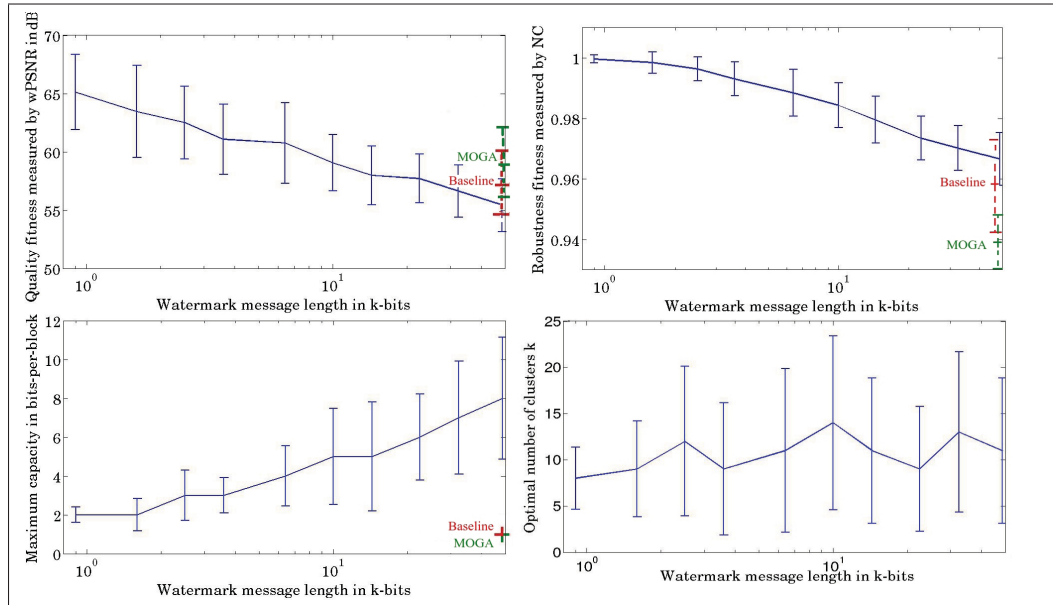


Figure 2.12 Proposed system performance with uneven scheme for different watermark lengths WL .

for k of 1 is reached, yielding to additional 67% complexity reduction. The performance of the proposed system using watermarks of different lengths up to 48.8 k-bits is compared against the baseline system performance for the same length.

Table 2.5 Impact of β on the total memory size of BCM Mem_{BCM} in k-bits, where $Rep_{float} = 32$ -bits as per IEEE 754 and $Rep_{int} = 6$ -bits to represent index of embedding coefficients $a \in [0, 1, \dots, 63]$.

β	Mem_{SC}	Mem_{RS}	Mem_{MFB}	Mem_{BCM}
30	1019	26	147	1192
20	1019	26	98	1143
10	1019	26	49	1094

Table 2.5 shows the impact of the tunable parameter β on the size of associative memory required for BCM Mem_{BCM} . The total memory Mem_{BCM} consists of memory required to store set of clusterings SC , robustness scores RS , and most frequent embedding bands MFB defined as Mem_{SC} , Mem_{RS} , and Mem_{MFB} respectively. The memory size is dependent on the binary representation of float and integers defined as Rep_{float} and Rep_{int} respectively, where

Mem_{SC} equals to $Rep_{float} \times 39 \times \sum_{k=K_{min}}^{K_{max}} k$, Mem_{RS} equals to $Rep_{float} \times \sum_{k=K_{min}}^{K_{max}} k$, and Mem_{MFB} equals to $Rep_{int} \times \beta \times \sum_{k=K_{min}}^{K_{max}} k$. As shown in Table 2.4, the maximum capacity for the stream $Max(C_{Max}(g))$ shows that β could be reduced to 10 instead of 20 for watermarks of length 6.4k. However changing β has minimal impact on the memory size specially the memory size is around only 1 Mega bits which can be afforded easily with ordinary computers.

Table 2.6 Impact of robustness scores threshold α on the proposed system performance for watermarks of different lengths WL for stream of 40 face images.

WL	α	Quality	Robustness	K	$C_{Max}(g)$	$Max(C_{Max}(g))$
1.6k	0.89	63.46±3.94	0.9985±0.0035	9±5.18	2±0.83	7
	0.92	63.53±3.94	0.9988±0.0031	8±4.93	2±0.45	4
	0.94	63.53±3.94	0.9988±0.0031	8±4.93	2±0.45	4
6.4k	0.89	60.75±3.47	0.9885±0.0078	11±8.85	4±1.54	7
	0.92	60.46±3.25	0.9899±0.0072	11±3.68	4±1.68	9
	0.94	60.39±3.24	0.9901±0.0071	10±7.63	4±1.76	9
22.5k	0.89	57.71±2.08	0.9735±0.0072	9 ±6.74	6±2.22	16
	0.92	56.75±2.43	0.9768±0.0082	9 ±5.08	7±3.52	17
	0.94	56.75±2.25	0.9771±0.0090	9 ±5.07	7±3.35	16

Table 2.6 shows the impact of the robustness scores threshold α to identify candidate embedding clusters on the proposed system with watermarks of different lengths WL . For small watermarks of length 1.6k, the quality and robustness fitness are not affected because the mean maximum capacity $C_{Max}(g)$ is 2 bits-per-block. This implies that only the two highest robustness scores clusters are used for embedding, and thus increasing the threshold up to 94% would not affect the fitness nor the clusters used for embedding. For medium watermarks of length 6.4k, the quality fitness is decreased slightly and robustness fitness is increased slightly when increasing α . This can be explained by the maximum embedding capacity used for the stream of the 40 face images $Max(C_{Max}(g))$ of 9 bits-per-block, compared to 7 bits-per-block for α equals to 0.89. This implies that increasing α for this watermark length WL would exclude some embedding clusters and increase the capacity of other clusters. For large watermarks of length 22.5k, the impact is similar to medium watermark however it is more significant increase

in mean robustness fitness, decrease in mean quality fitness, and increase in mean maximum capacity $C_{Max}(g)$.

Table 2.7 shows the complexity comparison between the fitness evaluation for both watermark quality and robustness using matlab, and migrating DCT transform to GPU implementation and the watermark embedding/extraction to C. The results show minimal Root Mean Square Error 0.1496, and 0.0054 for quality and robustness fitness respectively due to different representation for numeric formats between GPUs and matlab data structures. The experiment shows huge computational complexity reduction of cpu time for fitness evaluation of 40 face images stream, where GPU implementation is more than 100 times faster than CPU implementation. Time complexity is measured using the time to compute fitness of watermark quality T_{QF} and robustness T_{RF} for one face image. Watermark quality and robustness fitness evaluation for one face image involves 3 DCT transforms, and 2 inverse DCT transforms for the high-resolution face image.

Table 2.7 Impact of using GPU to perform DCT transform, and C for watermark embedding/extraction on fitness evaluation complexity $T_{QF} + T_{RF}$, accuracy for 40 face images, and optimization iteration complexity $T_{Iteration}$ in traditional approaches (Shieh *et al.*, 2004).

	GPU	CPU
Mean ($T_{QF} + T_{RF}$)[sec]	2.03±0.84	245.52±1.66
Stream ($T_{QF} + T_{RF}$)[sec]	81.10	9820.77
Quality RMSE		0.1496
Robustness RMSE		0.0054
$T_{Iteration} \times 10^3$ [sec]	29±1.63	34±2.12

However still the optimization iteration computation $T_{Iteration}$ in traditional methods is highly complex due to the constraints handling to avoid embedding on DC coefficient and embedding multiple bits in the same coefficient of the 8×8 block as shown in equation 1.8. And thus the proposed framework is well suited for GPU implementation more than traditional methods. This yields to a watermarking system throughput of 12 high-resolution face images per hour

compared to 0.0667 images per hour for traditional methods, with migrating only DCT costly transform into GPU implementation.

Table 2.8 Proposed system performance with different training set size N and embedding capacity C .

N	Quality	Robustness	k	$C_{Max}(g)$
Capacity=1-Bit per Block, $T_{Iteration}=1361\pm 13.98$				
1	66.56±8.72	0.9734±0.0139	17±9.19	3±0.62
2	80.29±2.73	0.9673±0.0176	19±10.35	3±0.90
3	80.03±2.85	0.9663±0.0192	18±8.66	3±0.73
7	78.05±3.39	0.9641±0.0172	18±9.84	3±1.26
Capacity=2-Bit per Block, $T_{Iteration}= 3524\pm 8.10$				
1	80.14±2.10	0.9675±0.0164	21±9.82	3±1.17
2	80.67±1.76	0.9687±0.0176	23±11.26	3±1.07
3	81.08±1.57	0.9645±0.0211	19±8.04	2±0.68
7	78.73±2.99	0.9637±0.0169	19±9.83	3±1.10
Capacity=4-Bit per Block, $T_{Iteration}= 6513\pm 573.76$				
1	69.32±3.57	0.9982±0.0031	14±8.70	3±0.90
2	80.37±2.35	0.9677±0.0202	21±9.58	3±0.78
3	79.80±2.49	0.9668±0.0215	17±9.67	3±0.87
7	77.92±2.65	0.9685±0.0166	24±10.72	3±1.08
Capacity=8-Bit per Block, $T_{Iteration}= 28978\pm 1628.56$				
1	63.46±3.94	0.9985±0.0035	9±5.18	2±0.83
2	63.48±4.99	0.9988±0.0026	9±3.78	2±0.60
3	64.08±4.81	0.9993±0.0016	10±7.29	2±0.62
7	66.70±3.34	0.9978±0.0038	13±6.64	3±0.98

Table 2.8 shows the impact of increasing training set size from 1 training face image to 2, 3, and 7 face images using different embedding capacities 1, 2, 4, 8 bits per block for training face images. The optimal performance is accomplished using 3 face images for training with capacity 8 bits per block with mean robustness equals to 0.9993, using more training face images would result in over-training. The cpu time for single iteration $T_{Iteration}$ for the full optimization step of the training is shown in the table as well. Figure 2.13 shows the impact of increasing training set size N and embedding capacity on the watermarking fitness produced by the proposed system.

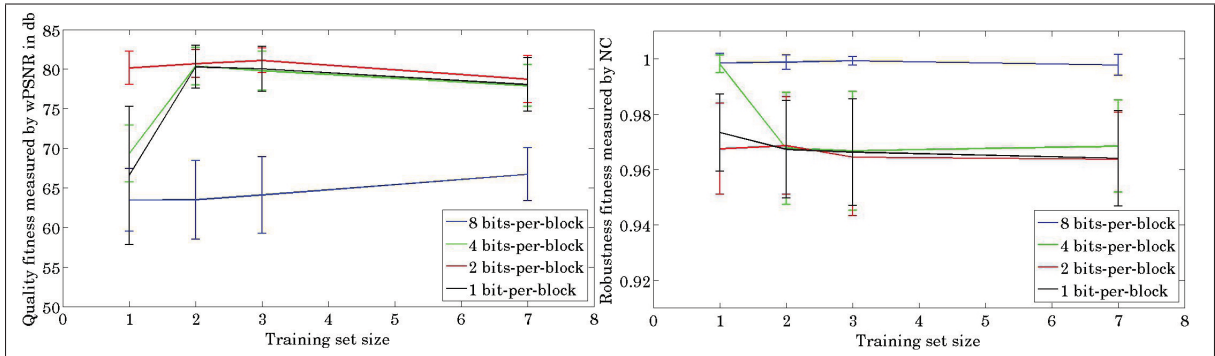


Figure 2.13 Impact of increasing training set size N on the proposed system performance.

Table 2.9 shows the performance of the proposed system using training set of size $N = 3$ and watermark length $WL = 1.6k$ concluded from previous experiments for two different testing sets. The stream of face images size is $M = 40$, and 198 faces respectively for the two testing sets. The quality of solutions are almost the same for training set face images, where the mean quality is slightly degraded from 64.08 dB to 63.96 dB for larger testing set, and the mean robustness fitness is still around the robustness fitness threshold of 99.9% for both sets.

Table 2.9 Proposed system performance with two different testing sets of size M equals to 40 and 198 face images respectively using training set size N equals to 3 face images.

M	Quality	Robustness	k	$C_{Max}(g)$
40	63.71 ± 4.59	0.9987 ± 0.0027	10 ± 7.03	2 ± 0.59
198	63.96 ± 4.05	0.9987 ± 0.0028	11 ± 8.68	2 ± 0.72

2.6 Conclusions and Future Directions

Intelligent watermarking for streams of high-resolution grayscale face images using evolutionary optimization is a very costly process, it has large dimension of search space due to representing all image blocks in the population of candidate solutions. The positional representation of these blocks in traditional methods for grayscale face images results in expensive re-optimizations when shifting face image pixels inside the image. Also the application domain

priorities variations result in costly re-optimizations for single objective optimization formulation to adjust the aggregation according to the priorities change.

In this chapter, we presented BMRC framework which replaces stream of similar optimization problems with BCM associative memory recalls. This BCM memory holds optimization solutions statistics for 8×8 pixels blocks grouped according to their texture, such that the most frequent embedding parameters for all blocks of the same texture are selected together during generalization phase. The training phase of BMRC is based on multi-objective optimization to populate BCM, such that the selected solution from Pareto front based on objectives priorities is used. If the objectives priorities vary, no costly re-optimizations are needed because only another solution from the Pareto front will be selected to populate BCM. The proposed BCM holds the optimization solutions for different clustering resolutions in texture feature space and postpone the hard decision for the optimal clustering resolution till the end of the watermarking process. A metric RS is proposed to rank block clusters for embedding and enable handling watermarks of different lengths. This metric identifies the optimal embedding blocks grouped according to their texture.

Simulation results show a significant complexity reduction measured in number of fitness evaluations including the training phase of BMRC. This complexity reduction is 93.5% for a stream of 40 face images, and it is increased up to 95.5% for a stream of 198 face images. The quality of solution produced by the proposed framework are almost of the same accuracy of full optimization. Sensitivity analysis shows the optimal message length of 1.6k when considering robustness fitness of 99.9% threshold, and evaluates the impact of other tunable parameters on framework performance and associative memory size.

The concept presented in this chapter can be generalized on any stream of optimization problems with large search space, where the candidate solutions consists of smaller granularity problems that affect the overall solution. The challenge for applying this approach is to find the significant feature for this smaller granularity that affects the overall optimization problem. In this chapter the texture features of smaller granularity blocks represented in the candidate solutions are affecting the watermarking fitness optimization of the whole image.

In a future work, the performance of the proposed method will be measured for heterogeneous streams of face images with various poses and backgrounds. Such heterogeneous streams will pose additional challenges for the proposed framework with more variations of textured areas and locations of face pixels inside the face image. The key challenge of this future work is the change detection mechanism. This mechanism should be capable of detecting different types of changes in the heterogeneous stream. The expected changes ranges from minor change in face pixels location till major change in the content or the topic of the image. In the case of major changes, the knowledge stored in the BCM associative memory could be insufficient for the current image, and thus full optimization is launched for this image and add the knowledge of optimizing this image to the associative BCM memory. Minor changes can be ignored if they are not highly affecting the resultant fitness. Migrating more functionalities of the proposed system to GPU implementation would be useful to increase the throughput to be more close to high speed reality applications with large streams.

2.7 Discussion

This chapter addressed the challenge of the similar optimization problems corresponding to intelligent watermarking of homogeneous streams of high-resolution facial images. BMRC relies on optimizing few facial images during training phase, and storing the results in associative memory. The embedding parameters optimization for high-resolution facial images is characterized by high dimension search space and computationally expensive fitness evaluations.

This optimization problem suffers from premature convergence due to the large dimension of the search space of EC methods. Resolving the premature convergence during training phase improves significantly the quality of solutions produced by BMRC. Chapter 3 describes a specialized algorithm to resolve this premature convergence based on cooperative coevolution, and speed up the optimization required during training phase.

CHAPTER 3

BLOCKWISE COEVOLUTIONARY GENETIC ALGORITHM (BCGA)

In biometric systems, reference facial images captured during enrollment are commonly secured using watermarking, where invisible watermark bits are embedded into these images. Evolutionary Computation (EC) is widely used to optimize embedding parameters in intelligent watermarking (IW) systems. Traditional IW methods represent all blocks of a cover image as candidate embedding solutions of EC algorithms, and suffer from premature convergence when dealing with high-resolution grayscale facial images. For instance, the dimensionality of the optimization problem to process a 2048×1536 pixel grayscale facial image that embeds 1 bit per 8×8 pixel block involves 49k variables represented with 293k binary bits. Such Large-Scale Global Optimization (LSGO) problems cannot be decomposed into smaller independent ones because watermarking metrics are calculated for the entire image. In this chapter, a Blockwise Coevolutionary Genetic Algorithm (BCGA) is proposed for high dimensional IW optimization of embedding parameters of high-resolution images. BCGA is based on the cooperative coevolution between different candidate solutions at the block level, using a local Block Watermarking Metric (BWM). It is characterized by a novel elitism mechanism that is driven by local blockwise metrics, where the blocks with higher BWM values are selected to form higher global fitness candidate solutions. The crossover and mutation operators of BCGA are performed on block level. Experimental results on PUT face image database indicate a 17% improvement of fitness produced by BCGA compared to classical GA. Due to improved exploration capabilities, BCGA convergence is reached in fewer generations indicating an optimization speedup.

The content of this chapter is published in *Expert Systems with Applications* (Rabil *et al.*, 2013c).

3.1 System Overview

In an earlier work, Blockwise Multi-Resolution Clustering was proposed by the authors (Rabil *et al.*, 2013a) for rapid IW for streams of facial images. During the training phase of this framework, a limited number of face images are optimized, and the optimization results are stored in associative memory according to texture features. During generalization, streams of facial images are watermarked using the recalled optimal solutions from this associative memory. This framework provided a significant reduction of complexity up to 95.5% fitness evaluations compared to full optimization of all images in a homogeneous stream of high-resolution facial images (Kasinski *et al.*, 2008). However the quality of solutions are dependent on the full optimization performed on images during training. Efficient optimization for high dimension problems would improve the quality of solutions produced by this proposed framework.

In this chapter, specialized algorithm based on GA is proposed for LSGO applied on the optimization of embedding parameters in IW of high-resolution images. To avoid embedding bits in smooth background that have low embedding capacity, only the textured blocks are considered as optimization candidate solution. The watermarking metrics are calculated globally on the whole image, however the local watermarking metrics for blocks can be used to guide the search through solutions. BCGA utilizes the cooperative coevolution between different solutions at the block level using the local watermarking metrics of these blocks. The embedding parameters of each block corresponding to higher local metrics are selected from different candidate solutions to construct new elite candidate solutions to be kept for next optimization generation. The crossover and mutation operators of BCGA are performed on block level rather than the entire candidate solution.

Proof of concept simulations are performed using the PUT database (Kasinski *et al.*, 2008) of high-resolution face images. This dataset provides homogeneous high-resolution face images with smooth backgrounds. Reduced resolutions face images are experimented to identify the premature convergence limitation for traditional GA with higher resolution images. Then the original resolution of face images are used to evaluate the performance of the proposed approach. BCGA performance is compared against traditional GA and PBIL in terms of qual-

ity of solutions, and fitness evolution. Different fitness aggregating methods are evaluated to avoid anomalies with Pareto front for conflicting objectives of watermark quality and robustness. User-defined parameters of the proposed BCGA are tuned in the experimentation, and the impact of these parameters on BCGA fitness evolution is analyzed.

The optimization of embedding parameters for high-resolution images shown in Figure 3.1 has high dimension of search space. It starts with representing only textured blocks TB in the candidate solution using perceptual texture masks, then an application specific GA algorithm called BCGA is used for embedding parameters optimization. It utilizes cooperative coevolution for local watermarking metrics of blocks, and performs GA operators crossover and mutation on block level. This coevolution on the block level improves GA algorithm capabilities for large optimization problems using cooperation of smaller problems.

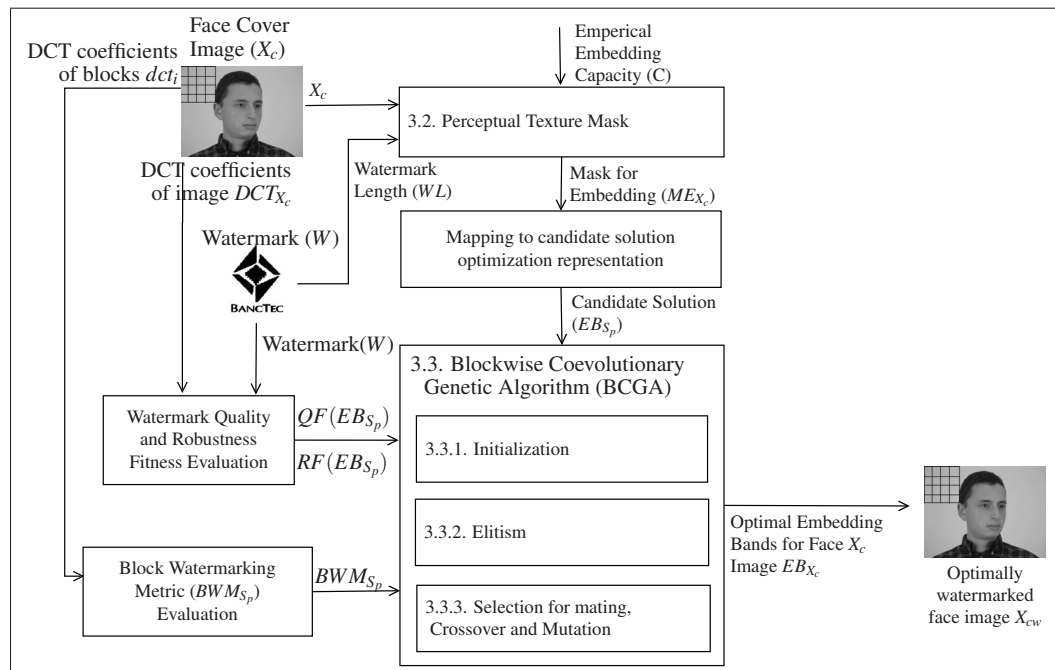


Figure 3.1 Block diagram for the proposed watermarking system for high-resolution face images.

The perceptual texture mask produces the mask for embedding ME_{X_c} for the face image X_c , where $ME_{X_c} = \{me_1, me_2, \dots, me_i, \dots, me_{NB}\}$, and $me_i \in \{0, 1\}$. All face image X_c blocks

b_i are represented in the mask for embedding, where $me_i = 1$ indicates that the block b_i is selected for embedding based on texture criteria defined by the perceptual texture mask. And thus the total number of blocks whose $me_i = 1$ is equal to the number of textured blocks selected for embedding TB , where $TB = \sum_{i=1}^{NB} me_i$. Only the blocks of me_i equals to 1 are represented in the candidate solution for optimization, this mapping ensures optimizing the embedding parameters in textured blocks only. For even embedding scheme with equal embedding capacity for all blocks, all blocks are represented in the candidate solutions for embedding. The proposed chromosome representation is shown in Figure 3.2.

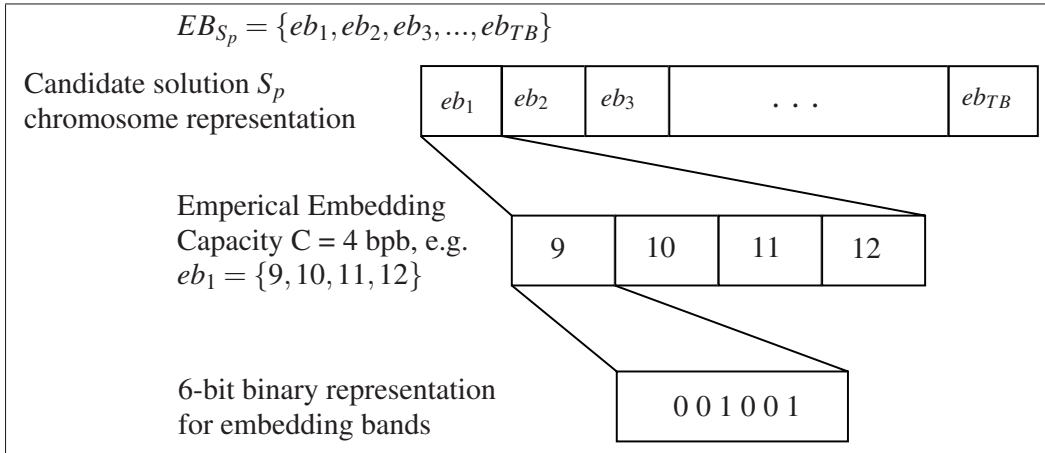


Figure 3.2 Chromosome representation of the proposed BCGA.

Using EC methods, the optimization process relies on a population of candidate solutions traversing search space to find the optimal solution. The population of candidate solutions S_p consists of pop solutions. Each candidate solution S_p consists of the sets of embedding bands for different blocks b_i , where i ranges from 1 to TB , and thus $EB_{S_p} = \{eb_1, eb_2, eb_3, \dots, eb_{TB}\}$. The size of the set eb_i equals to empirical embedding capacity C . And thus the dimension of the optimization problem to find the optimal embedding bands equals to $TB \times C \times 6$ bits, where TB is the number of textured blocks selected by perceptual texture mask, C is the empirical embedding capacity, and 6-bits are used to represent embedding bands ranging from 0 to 63 for blocks of size 8×8 pixels. The candidate solution of the best global fitness EB_{S_p} is

selected as the optimal solution EB_{X_c} for face image X_c , representing the optimal embedding bands for this face image such that $gf(EB_{X_c}) = \max_{p=1 \rightarrow pop} gf(EB_{S_p})$.

The global fitness to be optimized $gf(EB_{X_c})$ is the aggregated fitness for the whole face image X_c using both watermark quality and robustness against different attacks. For each candidate solution EB_{S_p} , block watermarking metric bwm_i are calculated for each block b_i using the embedding bands eb_i defined in EB_{S_p} . This bwm_i metric is a local aggregated fitness for metrics representing both watermark quality and robustness against different attacks using weighted sum aggregation. It is utilized in cooperative coevolution on block level for GA, where the blocks of higher bwm_i are assumed to form better candidate solutions EB_{S_p} having better global fitness $gf(EB_{S_p})$. The set of BWM for candidate solution S_p is defined as BWM_{S_p} , where $BWM_{S_p} = \{bwm_1, bwm_2, \dots, bwm_{TB}\}$

3.2 Perceptual Texture Masks

Biometric high-resolution face images are characterized by having areas of smooth textures representing the background, and the face image for individuals with more textured areas. Embedding watermarks in smooth textured areas results in degradation for watermark fitness (Wu, 2001). Traditional IW methods (Shieh *et al.*, 2004) relies on embedding bits in all cover image blocks, and thus these are not efficient with high-resolution face images with smooth backgrounds. Using these traditional methods, handling watermarks of small lengths would involve padding these watermarks bits to represent all cover images in candidate solutions. And thus the minimum search space for these methods is equal to the number of cover image blocks.

The perceptual texture mask component selects the most textured blocks as blocks of interest for embedding the watermark. It utilizes texture metrics to find the most textured blocks TB . In this chapter, the number of textured blocks TB is calculated based on the watermark length WL , and the empirical embedding capacity C . The embedding capacity is assumed to be equal for all textured blocks, and thus the number of textured blocks equals to WL/C .

3.3 Blockwise Coevolutionary Genetic Algorithm (BCGA)

The proposed BCGA is an application specific optimization algorithm derived from GA. It utilizes cooperative coevolution for subcomponents represented by 8×8 pixels blocks. Local watermarking metrics are used for these blocks to conclude the best fitness optimization candidate solutions using the best fitness subcomponent. The BWM of block b_i is defined as $bwm_i(eb_i)$, where eb_i are the embedding bands of block b_i , and the global fitness of the whole face image using embedding bands defined in candidate solution S_p is defined as $gf(EB_{S_p})$, where EB_{S_p} are the embedding bands of all blocks of X_c for this solution. The global fitness $gf(EB_{S_p})$ is the aggregated fitness for watermark quality fitness and watermark robustness fitness against different attacks.

All the candidate solutions in the population are used to find *Elite* solutions, the number of these elite solutions is defined as EL . These *Elite* solutions are defined as $Elite = \{EB_{S_{p1}}, EB_{S_{p2}}, \dots, EB_{S_{pEL}}\}$, where $Elite(el)$ refers to the elite candidate solution of index el in the elite set. This set of solutions *Elite* is used in elitism algorithm on the block level using bwm_i , where the embedding bands for blocks corresponding to the highest bwm_i are concatenated to generate the blockwise elite solutions. Crossover is performed on the selected parents of highest $gf(EB_{S_p})$ on the block level using two-points crossover with probability $P_{crossover}$, then mutation is applied on block level as well by changing random bits of probability $P_{mutation}$. The fitness of the new offspring generated is calculated, where the candidate solutions of least fitness are replaced with the most fit new offspring.

The objectives are proposed to be aggregated using Chebyshev weighted aggregation (Velasques *et al.*, 2012). This aggregation method is more robust to anomalies in the trade-off between the various fitness functions in a multi-objective optimization problem. In the Chebyshev approach, fitness values are aggregated according to their distances from reference points, under which the values of these fitness are considered good (Collette and Siarry, 2008) as shown in Equation 3.1.

$$\begin{aligned}
gf(EB_{X_c}) = \max_{l=1,2,3,4} \{ & (1 - w_1)(\gamma \cdot QF(EB_{X_c}) - r_1), \\
& (1 - w_2)(RF_1(EB_{X_c}) - r_2), \\
& (1 - w_3)(RF_2(EB_{X_c}) - r_3), \\
& (1 - w_4)(RF_3(EB_{X_c}) - r_4) \}
\end{aligned} \tag{3.1}$$

where $(1 - w_l)$ is the weight of the l^{th} objective with $(1 - w_l) = \frac{1}{4} \forall l$, r_l is the reference point of objective l , and γ is a scaling weight for quality fitness $QF(EB_{X_c})$

The proposed local watermarking metrics are Peak Signal To Noise Ratio (PSNR) and Bit Correct Rate (BCR), where these metrics are the least complex metrics applicable to block level. The aggregated weighted sum $bwm_i(eb_i)$ for block b_i is defined in Equation 3.2, where $PSNR_i(eb_i)$ is the peak signal to noise ratio for block b_i , $BCR_{i1}(eb_i)$ is the bit correct ratio for first attack on block b_i , $BCR_{i2}(eb_i)$ is the bit correct ratio for second attack on block b_i , and $BCR_{i3}(eb_i)$ is the bit correct ratio for third attack on block b_i .

$$bwm_i(eb_i) = PSNR_i(eb_i) + \lambda(BCR_{i1}(eb_i) + BCR_{i2}(eb_i) + BCR_{i3}(eb_i)) \tag{3.2}$$

where PSNR is defined in equation 1.1 and BCR measures the ratio of correctly extracted bits to total embedded bits as shown in Equation 3.3, where BC_i represents the number of correctly extracted watermark bits from block b_i , and C_i is the total number of watermark bits embedded in this block representing the empirical embedding capacity in bits per block.

$$BCR_i = BC_i / C_i \tag{3.3}$$

Figure 3.3 shows the data flow for BCGA to find optimal embedding bands EB_{X_c} for face image X_c , where the processing steps of Algorithm 6 are mapped to the data flow.

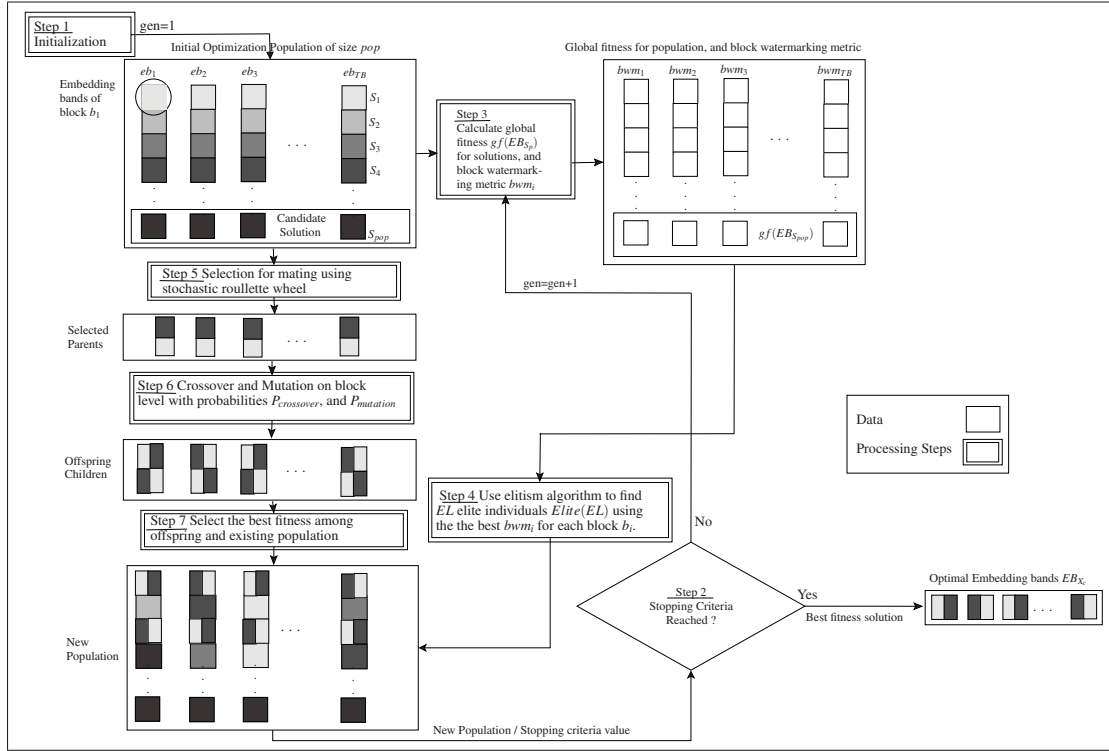


Figure 3.3 Data flow diagram for the BCGA to find optimal embedding bands EB_{X_c} for image X_c .

Algorithm 6 shows the main algorithm of BCGA to find optimal embedding bands EB_{X_c} for face image X_c . The population of candidate solutions is initialized (line 1) using Algorithm 7. Then the global fitness, and BWM are calculated for all candidate solution in the population (line 3). Using bwm_i , the blocks are sorted (line 4) as shown in Algorithm 8, where the blocks of higher local fitness from different candidate solutions are concatenated to generate new candidate solutions. The parent candidate solutions used for reproduction are selected using stochastic roulette scheme (line 5). Crossover and mutation operators use masks on block level to perform these operations to breed new candidate solutions (line 6) as shown in Algorithm 9. The global fitness of the new offspring is calculated (line 6), and the candidate solutions of least fitness from the population are replaced with the new offspring, then elite solutions $Elite(el)$ are inserted in the population (line 7). Until stopping criteria is reached (line 2), global fitness is improving iteratively along generations.

Algorithm 6 Finding optimal embedding bands EB_{X_c} for face image X_c using BCGA.

Input: Face image X_c blocks b_i , crossover probability $P_{crossover}$, mutation probability $P_{mutation}$, and number of elite solutions in population EL .

- 1: Initialize population of candidate solutions of size pop . The dimension of each candidate solution in this population equals to $TB \times C \times 6$ bits. Algorithm 7 shows the details of initialization process.
- 2: **while** Stopping criteria is not reached **do**
- 3: Evaluate global fitness $gf(EB_{S_p})$ for the whole candidate solution S_p for all solutions in the population, and BWM for blocks BWM_{S_p} .
- 4: The blocks are sorted using the metric $bwm_i(eb_i)$ as shown in Algorithm 8. The embedding bands corresponding to highest bwm_i are assumed to have good global fitness $gf(EB_{S_p})$ when concatenated to form new elite solution $Elite(el)$.
- 5: Select best fitness individuals for reproduction using stochastic roulette wheel bias method.
- 6: Breed new candidate solutions through crossover and mutation on the block level with probabilities $P_{crossover}$, and $P_{mutation}$ respectively to give birth to offspring candidate solutions.
- 7: Evaluate the global fitness of new offspring candidate solutions and replace the least fitness candidate solutions of the population with the best fitness offspring, then insert EL elite solutions from step 4 in the population.
- 8: **end while**

Output: Optimal embedding bands EB_{X_c} for face image X_c

3.3.1 Initialization

Algorithm 7 shows the initialization process for the population of candidate solutions. The embedding bands for all blocks in the same candidate solution S_p are the same at initialization time. For the first method (lines 1-7) applied on the first portion of the population of candidate solutions, the initial bands are chosen based on the index of candidate solution p , index of embedding bands j , and embedding capacity C . The embedding bands set is initialized to an empty set (line 2), then it is concatenated with different initial embedding bands (line 3-5) until the embedding capacity C is reached. This set of embedding bands eb_i is replicated TB times (line 6) to form the initial candidate solution S_p . For the second method (lines 9-16) applied on the second portion of the population, the initial bands are chosen using random integers ranging from 1 to $63-C$, and index of embedding bands j . For populations of smaller size, only the first method of initialization is used. For example when using embedding capacity $C = 4$ for a population of size $pop = 24$, the first method of initialization is applied on candidate solutions S_p , where $p = 1, 2, \dots, 14$, and the second method is applied on $p = 15, 16, \dots, 24$. This initialization algorithm ensures high diversity in the initial population of candidate solutions.

Algorithm 7 Initialization of the proposed BCGA, where $repeat(V, r)$ is a function that construct a vector by replicating r times the vector V .

Input: Candidate solution index p , population size pop , number of textured blocks for embedding TB , index of embedding bands j in embedding bands set eb_i for block b_i , and embedding capacity C

```

1: for  $p=1$  to  $(64/C)-1$  do
2:   Initialize empty set for embedding bands  $eb_i=\{\}$ .
3:   for  $j=1$  to  $C$  do
4:      $eb_i = \{ eb_i (p-1)*C+j \}$  #uniform empirical embedding bands based on  $p$ , and  $C$  for block  $b_i$ .
5:   end for
6:    $EB_{S_p}=repeat(eb_i, TB)$ ;
7: end for
8: if  $pop > (64/C) - 1$  then
9:   for  $p = (64/C) - 1$  to  $pop$  do
10:    Generate random variable  $rand$  of integer value between  $[1 \ 63-C]$ 
11:    Initialize empty set for embedding bands  $eb_i=\{\}$ .
12:    for  $j=1$  to  $C$  do
13:       $eb_i = \{ eb_i \ rand + j \}$  #random consecutive embedding bands initialization for block  $b_i$ .
14:    end for
15:     $EB_{S_p}=repeat(eb_i, TB)$ ;
16:  end for
17: end if

```

Output: Population of initial candidate solutions EB_{S_p} , where $p = 1, 2, \dots, pop$

3.3.2 Elitism

As shown in Figure 3.4, the sets BWM_{S_p} are calculated for all candidate solutions in the optimization population of size pop . For each block b_i , the corresponding bwm_i are sorted descendingly. The blocks of highest bwm_i are concatenated together to generate the elite solution $Elite(1)$, and the second best blocks are concatenated together to generate the second elite solution $Elite(2)$. Thus candidate solutions cooperatively exchange the information on the block level for improved exploration capabilities for the large dimension of the search space.

Algorithm 8 shows the proposed algorithm of elitism. The metric bwm_i is calculated for the blocks of all candidate solutions (line 1-5), where this metric represents a measure to rank blocks based on watermarking fitness. For each block b_i , the values of bwm_i are sorted to find the indices p of top values of bwm_i (line 6-8). The embedding bands corresponding to the top values of bwm_i are concatenated together to generate elite individuals $Elite(el)$ (line 9-11),

	bwm_1	bwm_2	bwm_3	bwm_4	bwm_5	bwm_6	bwm_7	bwm_8	...	bwm_{TB}
BWM_{S_1}	87	3022	3012	82	95	3017	81	3015	...	3030
BWM_{S_2}	3030	85	3022	92	3025	3019	3025	92	...	80
BWM_{S_3}	82	3027	92	3019	3029	3027	87	89	...	3012
...										
$BWM_{S_{pop}}$	92	3019	3030	3012	89	82	95	3022	...	90
$Elite(1)$	3030	3027	3030	3019	3029	3027	3025	3022	...	3030
$Elite(2)$	92	3022	3022	3012	3025	3019	95	3015	...	3012
$Elite(3)$	87	3019	3012	92	95	3017	87	92	...	90
$Elite(4)$	82	85	92	82	89	82	81	89	...	80

Figure 3.4 Proposed elitism on block level with $EL = 4$ using block watermarking metric bwm_i .

such that the embedding bands corresponding to the best values are used to generate $Elite(1)$, and the second best values used to generate $Elite(2)$.

3.3.3 Selection for Mating, Crossover and Mutation

The proposed selection for BCGA to find candidate solutions for crossover and mating is based on roulette wheel method. The global fitness of candidate solutions EB_{S_p} is normalized, then the cumulative normalized fitness values are stored in an array where the last fitness in this array is equal to 1. This method is used to select the parent candidate solutions $EB_{S_{p11}}$, and $EB_{S_{p22}}$ for crossover and mutation to generate the new offspring.

As shown in Figure 3.5, the crossover is performed on block level using 2-point crossover. The embedding capacity $C = 4$ bits-per-block, yielding to a size of 24 binary bits for embedding bands for each block eb_i . The embedding bands are reperedsted using 6 bits for AC coefficients in 8×8 blocks. For the example shown in the figure the 2-point crossover results in 3 equal

Algorithm 8 Elitism based on local fitness of BCGA.

Input: Candidate solutions S_p , where p is the index of candidate solution, BWM_{S_p} set of BWM for candidate solution S_p , where it consists of bwm_i for block b_i , textured blocks TB , index of elite candidate solution el , and number of elite solutions EL .

```

1: for  $p=1$  to  $pop$  do
2:   for  $i=1$  to  $TB$  do
3:     Calculate  $BWM_{S_p}$  of candidate solution  $S_p$  of index  $p$ , where each  $BWM_{S_p}$  consists of
        $bwm_i$  for all blocks  $b_i$ .
4:   end for
5: end for
6: for  $i=1$  to  $TB$  do
7:   Sort the blocks descendingly for each block  $b_i$  to find the indices  $p$  of the top values of  $bwm_i$  in
        $BWM_{S_p}$ .
8: end for
9: for  $el=1$  to  $EL$  do
10:  Concatenate the embedding bands  $eb_i$  of all blocks corresponding to the top values  $bwm_i$  from
        $BWM_{S_p}$  of different candidate solutions  $S_p$  to generate elite candidate solutions  $Elite(el)$  using
       the indices in Step 7.
11: end for

```

Output: EL elite candidate solutions of index el defined as $Elite(el)$.

portions of 8 bits to be crossovered. A mask for crossover of each block cm_i is generated to be concatenated to form the crossover mask for the whole solution CM . The whole crossover mask is applied on the whole solution to obtain the crossovered solutions. Then mutation is applied similarly on block level to obtain mutated crossovered solutions, where random bits are altered with probability $P_{mutation}$ for bits representing the embedding bands for block eb_i .

Algorithm 9 shows the crossover and mutation operators used to generate new offspring candidate solutions using selected parent solutions $EB_{S_{p11}}$, and $EB_{S_{p22}}$. The crossover is performed on block level (line 1-5), where the crossover mask cm_i is generated for blocks (line 1-3) using the number of crossover points cp with probability $P_{crossover}$. The block crossover masks cm_i are concatenated to form the whole candidate solution crossover mask CM (line 4). This mask CM is applied to the selected parents to obtain crossovered solutions $EB_{S_{p21}}$, and $EB_{S_{p12}}$ (line 5). Also mutation is performed on block level (line 6-10), where this starts with generating mutation masks for blocks mm_i (line 6-8) with probability $P_{mutation}$ to change bit values. These block masks are concatenated to generate the whole candidate solution mask MM (line

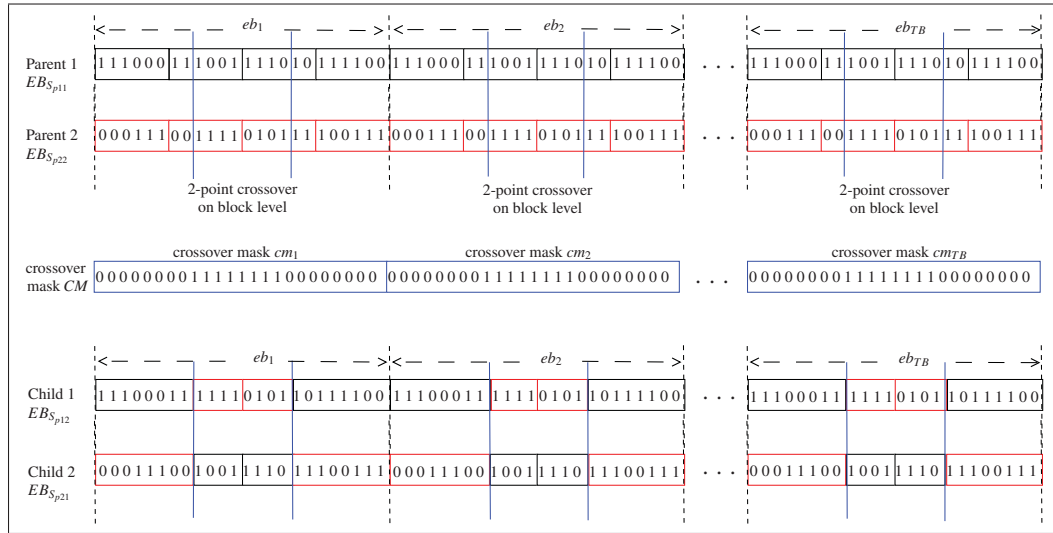


Figure 3.5 Proposed crossover on block level with $cp = 2$, embedding capacity $C = 4$, and 6-bit representation for embedding bands.

- 9). This mask MM is applied to crossed solutions to obtain the mutated crossed solutions $EB_{S_{p21m}}$, and $EB_{S_{p12m}}$ (line 10).

Algorithm 9 Proposed crossover and mutation operators on the block level of BCGA.

Input: Candidate solution indices $p11$ and $p22$ for selected parents, number of textured blocks for embedding TB , embedding capacity C , probabilities of crossover and mutation defined as $P_{crossover}$ and $P_{mutation}$ respectively, and number of crossover points cp .

- 1: **for** $i=1$ to TB **do**
- 2: Generate crossover mask cm_i for block b_i for selected parents candidate solutions $EB_{S_{p11}}$, and $EB_{S_{p22}}$ with probability $P_{crossover}$ based on cp .
- 3: **end for**
- 4: Concatenate block masks cm_i to obtain candidate solution crossover mask CM
- 5: Apply crossover mask CM on the parent candidate solutions $EB_{S_{p11}}$, and $EB_{S_{p22}}$ to obtain crossed solutions S_{p21} , and S_{p12}
- 6: **for** $i=1$ to TB **do**
- 7: Generate mutation mask mm_i for block b_i for the crossed solutions $EB_{S_{p21}}$, and $EB_{S_{p12}}$ with probability $P_{mutation}$.
- 8: **end for**
- 9: Concatenate block masks mm_i to obtain candidate solution mutation mask MM
- 10: Apply mutation mask MM on the crossed solutions $EB_{S_{p21}}$, and $EB_{S_{p12}}$ to obtain mutated crossed solutions of indices $p12m$, and $p21m$.

Output: New offspring candidate solutions of indices $p12m$, and $p21m$

3.4 Experimental Methodology

The database for face images used in experiments is the PUT (Kasinski *et al.*, 2008) face database which consists of 100 individuals where images feature 100 poses for each individual. Sample face images of the first pose of individuals are shown in Figure 3.6. The face images of this database are characterized by a smooth background for all individuals, and different light conditions and clothes. The percentage of background and foreground blocks are nearly the same for each individual, where the different poses represent head rotation and facial expression changes only. This ensures texture homogeneity for facial images to be used as cover images in watermarking application. Color face images of resolution 2048x1536 are converted to grayscale level. The first pose of all individual has the name pattern IIII1001.JPG where IIII is the individual number in 4 digits. Face images of this pose are used for proposed system development. The tuning set of face images consists of the first pose of the individuals starting from 0001 till 0040. This set is used for developing and tuning user-defined parameters of the proposed BCGA.



Figure 3.6 Sample face images from first pose of the PUT database (Kasinski *et al.*, 2008), where grayscale images have a resolution 2048×1536 pixels.

The watermark to be embedded is BancTec binary logo with different resolutions shown in Figure 2.8. The watermark embedding/extracting algorithm used in experiments is proposed by Shieh *et al.* (Shieh *et al.*, 2004) as illustrated in Section 1.2. The metrics used in experimentation for measuring watermark quality and robustness for the whole face image to find optimal bands

are wPSNR and NC respectively as defined in Section 1.1. BWM utilizes weighted sum of BCR and PSNR defined in Section 3.3. The attacks considered in the experimentation are JPEG compression with quality factor equals to 80%, median filtering, and low pass filtering. Entropy is used to select the most textured blocks.

The experiments are executed on gentoo linux based server of 24 GB memory size and 8 cores Intel Xeon CPU X5650 of speed 2.67GHz. The implementation utilizes Graphical Processing Units (GPU) for DCT transform using 8 NVIDIA Tesla C2050/C2070 GPU cards. The proposed methodology consists of experiments to compare BCGA to baseline system, then tuning aggregation weights, and finally sensitivity analysis on the key algorithm parameters.

Traditional GA is evaluated for different watermark lengths WL embedded in different resolutions facial images to study the convergence properties for different search space size. The default watermark length is 48.8 k, and the empirical embedding capacity for both BCGA and baseline is set to 1 bit-per-block. The weight λ used in aggregating bwm_i is equal to 30. The population size pop equals to 24, and the number of elite individuals EL equals to 6. The probabilities for crossover and mutation is set to 0.9 and 0.04 respectively.

The first experiment uses only the first face image to evaluate the convergence properties of the traditional GA. The embedding capacity is fixed to 1 bit per block, and the face image is resized to different reduced resolutions. The first face image is reduced to 20%, 40%, 60%, and 80% to characterize the evolution using different dimensions of the search space. Then the evolution of traditional GA is compared to the proposed BCGA evolution using the original size of the first face image which is 2048×1536 . In this experiment both GA and BCGA uses simple weighted sum aggregation with $\lambda = 20$, where $gf = QF + 20(RF_1 + RF_2 + RF_3)$. Using traditional GA with weighted sum aggregation represents the baseline system proposed by Shieh *et al.* (Shieh *et al.*, 2004). Two methods of crossover are experimented with the baseline GA to measure the impact of increasing crossover points on the performance of baseline GA. Single point crossover is compared to scattered crossover which uses random number of crossover points using randomly generated crossover masks. The baseline GA with the two crossover methods

are compared to PBIL and the proposed BCGA using the same population size of candidate solutions and the same weighted sum aggregation.

The next experiment use 40 face images to develop and tune different parameters of the proposed BCGA, where the watermark bits are embedded in the most textured 25% of the blocks using 4 bits-per-block. The second experiment is a sensitivity analysis for different aggregation methods. The aggregation methods evaluated are the simple weighted sum, and Chebyshev aggregation. The baseline GA fitness are considered reference fitness levels used in Chebyshev aggregation. These Chebyshev reference fitness values represents the minimum fitness values which are considered acceptable for decision maker.

The third experiment utilizes the tuned weights from the previous experiments to measure the impact of user defined parameters on the aggregated fitness evolution using Chebyshev aggregation. The evolution of fitness is measured for the first image for the most important parameters of BCGA. These parameters include the probability of both crossover and mutation defined as $P_{crossover}$ and $P_{mutation}$ respectively. Also the number of elite candidate solutions defined as EL is considered in this sensitivity analysis.

The fourth experiment evaluates the impact of using different texture metrics for grayscale images on the aggregated fitness evolution using Chebyshev aggregation. The fitness evolution is measured for the first image using Entropy, JND, NVF, and RS. These metrics represent different classes of grayscale texture measures defined in Section 1.5.2 in addition to the proposed RS in Chapter 2. These texture metrics are used to select the most textured blocks for embedding. The embedding capacity considered in this experiment is 4 bits-per-block, where the most textured 25% of the blocks are used for embedding 4 bits-per-block. For RS texture metric, the number of blocks groups is determined such that the number of blocks belonging to the most textured groups equals or slightly exceeding to 25% of the cover image blocks.

3.5 Results and Discussion

Table 3.1 shows the fitness produced, and the generation where convergence is assumed using traditional GA and PBIL compared to proposed BCGA for the first face image using its original

size. The fitness produced by BCGA is significantly higher robustness compared to GA. The evolution of fitness for methods is shown in Figure 3.7, where traditional GA is suffering from a premature convergence at a low fitness starting the generation 43. Using scattered crossover improves the baseline performance using single point crossover, however the traditional GA still suffers from premature convergence. This evolution of the baseline system shows that the GA parameters tuning would not improve the convergence properties in such high dimension of search space.

PBIL provides faster convergence than baseline GA with the two crossover methods. For PBIL, the intrinsic probability vector is evolving rather than candidate solutions. The best fitness candidate solution is used to update the evolving probability vector. This explains the fluctuation of the fitness of the global fitness gf using weighted sum aggregation for PBIL.

The average time complexity of BCGA generation is equal to 322 CPU seconds for original size of face image using GPU implementation for watermark fitness evaluations. BCGA has increasing fitness until the generation 20 on a weighted sum fitness which is 17% better compared to GA. The proposed elitism mechanism searches for the best embedding parameters for the same block across the population of candidate solutions, and thus it provides better exploration capabilities for BCGA. This indicates that few generations would be sufficient to produce fitness of satisfactory level.

Also Table 3.1 shows the fitness produced for different dimensions of the search space using BCGA compared to the baseline GA with scattered crossover points. The fitness produced is degrading for dimensions larger than 47k bits. The generation where convergence is assumed to decline as image dimension grows. This concludes the efficiency of the baseline system proposed by Shieh *et al.* (Shieh *et al.*, 2004) for dimension 47k bits or less. Optimization problem for IW of biometric facial capture images of resolution higher than 608×816 representing 40% reduction of PUT face images is suffering from premature convergence as shown in Figure 3.8. For cover images of lower resolution represented by 20% reduction of PUT face image original size, BCGA outperforms also traditional GA. However for such lower resolutions the traditional GA is not suffering from premature convergence. This proves the efficiency of the

Table 3.1 Optimization of embedding 1 bit-per-block in reduced size of face image 0001 from PUT database (Kasinski *et al.*, 2008) using baseline method proposed by Shieh *et. al* (Shieh *et al.*, 2004), PBIL and BCGA with $gf = QF + 20(RF_1 + RF_2 + RF_3)$.

Resize %-Dimension	Optimization Method	wPSNR [dB]	NC1 (JPEG QF=80)	NC2 (Median Filtering)	NC3 (LPF)	Conv. Gen.
20%-12k	GA-scattered	57.8835	0.9292	0.7250	0.7875	90
	BCGA	63.0753	0.9340	0.9303	0.9830	26
40%-47k	GA-scattered	57.7174	0.9184	0.6133	0.7585	87
	BCGA	62.3770	0.9287	0.9212	0.9761	26
60%-106k	GA-scattered	56.9713	0.9042	0.5582	0.7075	77
	BCGA	61.0249	0.9546	0.9593	0.9862	12
80%-187k	GA-scattered	57.0923	0.9068	0.5472	0.6859	78
	BCGA	62.2147	0.9509	0.9192	0.9756	48
100%-293k	GA-single	56.9532	0.9023	0.5411	0.6949	10
	GA-scattered	57.3236	0.9138	0.5337	0.6947	43
	PBIL	62.2491	0.9167	0.5535	0.7196	89
	BCGA	59.0047	0.9721	0.9269	0.9749	20

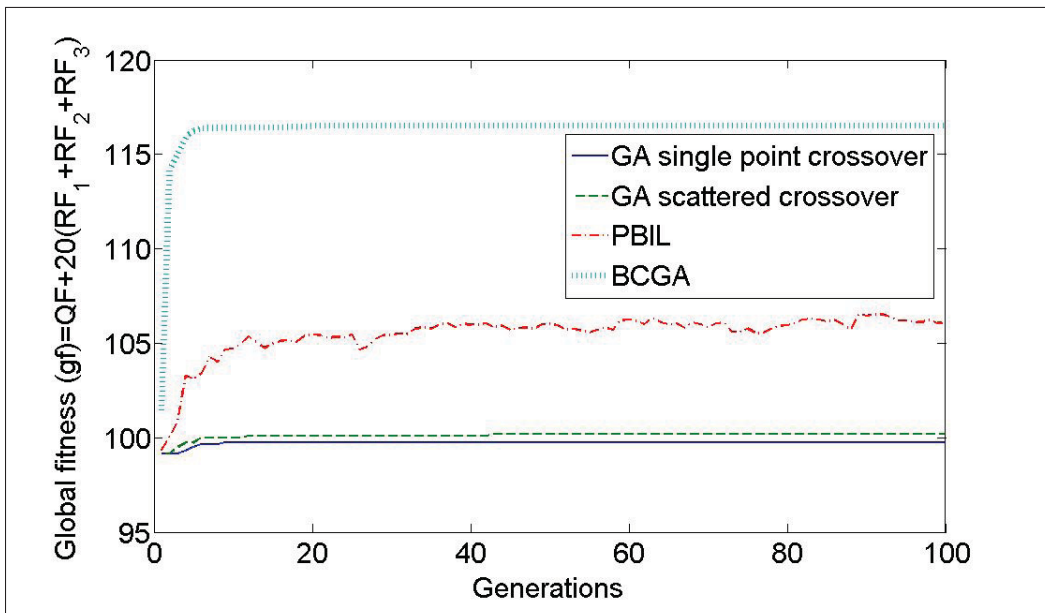


Figure 3.7 Evolution of baseline system using GA with two crossover methods, and PBIL compared to the proposed BCGA using embedding capacity of 1 bits-per-block for face image 0001.

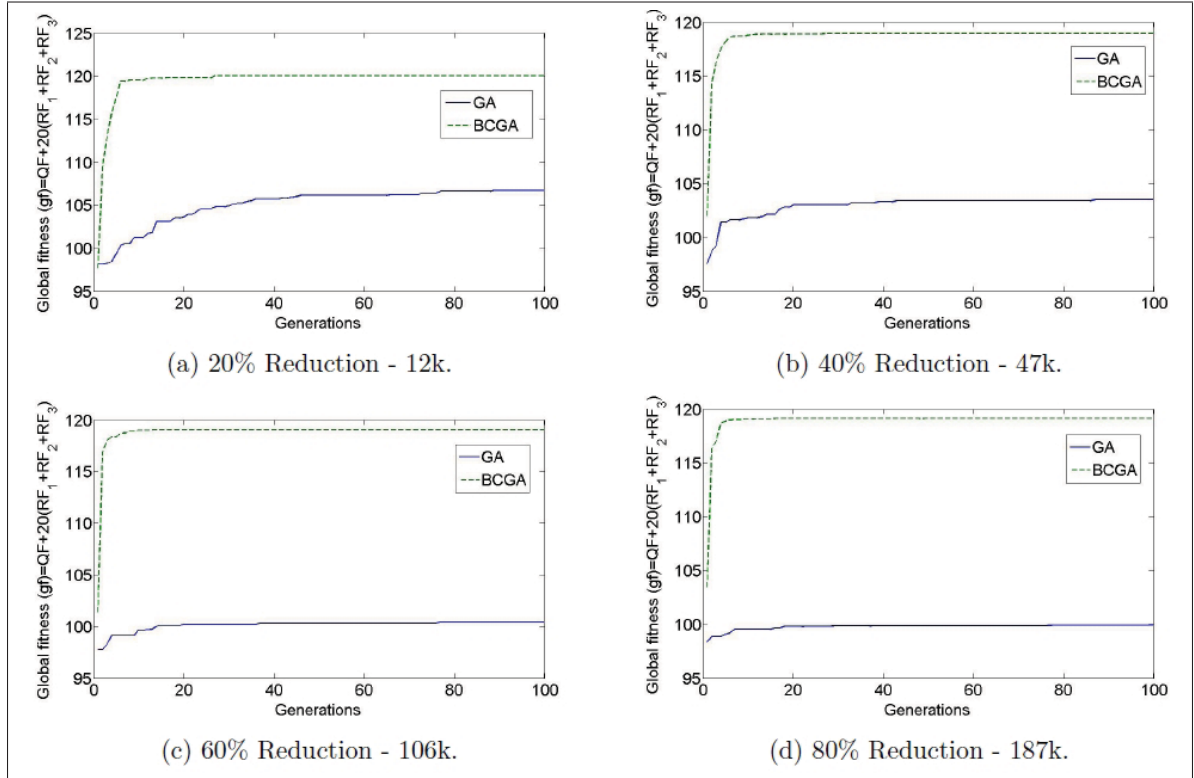


Figure 3.8 Evolution of reduced size face image 0001 using baseline system using GA proposed by Shieh *et al.* (Shieh *et al.*, 2004) with embedding capacity 1 bit-per-block using different sized watermarks.

baseline system proposed by Shieh *et al.* (Shieh *et al.*, 2004) only for low resolution grayscale images with no smooth textured areas. For lower resolution images used for experimentation of the baseline system, the traditional GA can converge for such dimension of search space as shown in Figure 3.8-a with 20% reduction of the high resolution face image.

Table 3.2 BCGA quality of solutions compared to traditional GA (Shieh *et al.*, 2004) using weighted sum aggregation for tuning set of 40 facial images.

Optim. Method	wPSNR [dB]	NC1 (JPEG QF=80)	NC2 (Median Filtering)	NC3 (LPF)
Baseline GA	57.27±1.59	0.9184±0.0063	0.5285±0.0085	0.6640±0.0325
Proposed BCGA	57.40±1.82	0.9472±0.0139	0.9223±0.0087	0.9691±0.0030

Table 3.2 shows the performance of both the baseline GA compared to the proposed BCGA for tuning set of face images including 40 face images. Both baseline GA and the proposed BCGA use weighted sum aggregation for quality and robustness fitness. The mean fitness for the baseline GA method is used in the next experiments as reference fitness values for Chebyshev aggregation defined as r_1, r_2, r_3 , and r_4 in equation 3.1.

Table 3.3 shows the performance of the proposed BCGA using Chebyshev aggregation with different weights for quality fitness γ as shown in equation 3.1. The resulting robustness fitness against different attacks are improved with decreasing the value of γ , on the other side quality fitness produced is degrading. Starting at the value of $\gamma = 1/200$, the robustness fitness against LPF and MF is no more improved, while the robustness against JPEG is slightly improved. The minimum quality fitness is over 42 dB which is considered acceptable level of watermark quality.

Table 3.3 Chebyshev aggregation using different quality fitness weights γ for embedding optimization using BCGA with embedding capacity equals to 4 bits-per-block for tuning set of 40 face images.

Weight γ	wPSNR [dB]	NC1 (JPEG QF=80)	NC2 (Median Filtering)	NC3 (LPF)
1/20	68.81±1.48	0.6286±0.0399	0.5020±0.0042	0.4789±0.0105
1/70	57.38±1.88	0.9452±0.0153	0.9265±0.0070	0.9694±0.0018
1/120	57.31±1.81	0.9475±0.0125	0.9271±0.0066	0.9698±0.0021
1/200	57.19±1.78	0.9525±0.0130	0.9270±0.0071	0.9698±0.0022

The next experiments deal with sensitivity analysis of different parameters on the BCGA evolution, where only the face image 0001 is used in these experiments. Figure 3.9 shows the fitness evolution in log scale of BCGA using Chebyshev aggregation shown in Equation 3.1 for different number of elite solutions EL . Table 3.4 shows the optimal fitness after 100 generations. The best fitness corresponds to using number of elite candidate solutions equals to 4 solutions. This value for EL is used in subsequent sensitivity experiments.

Table 3.4 Impact of the number of elite individuals EL on BCGA performance using Chebyshev aggregation for gf as shown in equation 3.1.

EL	wPSNR [dB]	NC1 (JPEG QF=80)	NC2 (Median Filtering)	NC3 (LPF)	gf
2	57.3322	0.9328	0.9172	0.9730	0.0968
4	57.3123	0.9367	0.9270	0.9722	0.0993
6	57.1159	0.9419	0.9202	0.9685	0.0975
8	57.4569	0.9369	0.9181	0.9673	0.0970

Figure 3.9 shows the fitness evolution in log scale of BCGA using Chebyshev aggregation for different probabilities of crossover on block level $P_{crossover}$. Table 3.5 shows the optimal fitness after 100 generations. The best fitness corresponds to using the probability $P_{crossover}$ equals to 90%. For probabilities less than this till 0.1% the evolution is nearly the same. The values of EL and $P_{crossover}$ producing the best aggregated fitness gf are used in subsequent sensitivity experiment. This shows minimal impact of $P_{crossover}$ and EL on the fitness evolution of BCGA.

Table 3.5 Impact of the probability of crossover $P_{crossover}$ on BCGA performance using Chebyshev aggregation for gf as shown in equation 3.1.

$P_{crossover}$	wPSNR [dB]	NC1 (JPEG QF=80)	NC2 (Median Filtering)	NC3 (LPF)	gf
0.9	57.3123	0.9367	0.9270	0.9722	0.0993
0.1	57.3711	0.9497	0.9232	0.9711	0.0983
0.01	57.2814	0.9482	0.9231	0.9712	0.0983
0.001	57.2814	0.9482	0.9231	0.9712	0.0983

Figure 3.10 shows the evolution of BCGA using Chebyshev aggregation for different probabilities of mutation on block level $P_{mutation}$. Table 3.6 shows the optimal fitness after 100 generations. The best fitness corresponds to using the probability $P_{mutation}$ equals to 4% which corresponds to (1/gene length). BCGA evolution is sensitive to probability of mutation $P_{mutation}$ more than EL and $P_{crossover}$ tunable parameters.

Figure 3.11 shows the evolution of BCGA using Chebyshev aggregation for different perceptual texture masks. Table 3.7 shows the optimal fitness produced after 100 generations. The

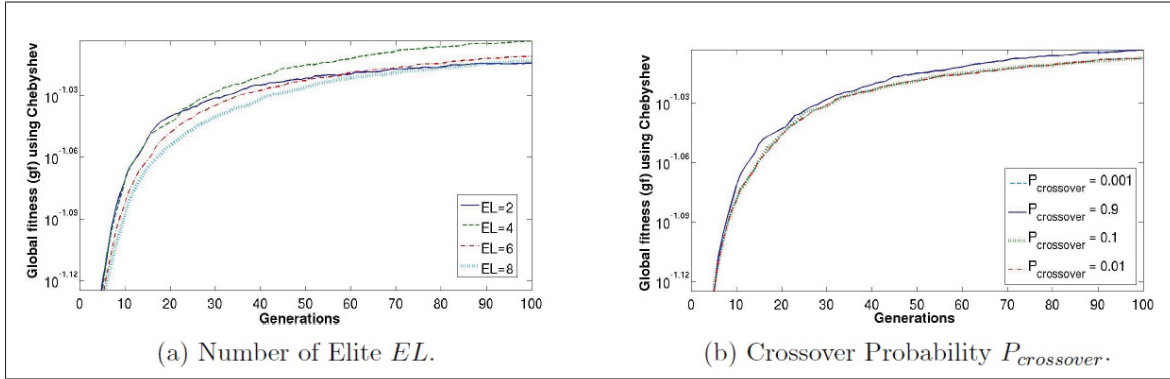


Figure 3.9 Impact of user-defined parameters of BCGA on fitness evolution using Chebyshev aggregation for gf as shown in equation 3.1. Log scale is used for gf due to the high similarity of fitness evolution for EL and $P_{crossover}$.

Table 3.6 Impact of the probability of mutation $P_{mutation}$ on BCGA performance using Chebyshev aggregation for gf as shown in equation 3.1.

$P_{mutation}$	wPSNR [dB]	NC1 (JPEG QF=80)	NC2 (Median Filtering)	NC3 (LPF)	gf
0.1	57.3989	0.9438	0.9238	0.9674	0.0985
0.04	57.3123	0.9367	0.9270	0.9722	0.0993
0.01	57.3031	0.9284	0.9133	0.9637	0.0958
0.001	55.5735	0.9216	0.8525	0.9186	0.0806

best global fitness is produced using entropy masks. Both JND and RS (Rabil *et al.*, 2013a) texture metrics are next to entropy, and finally NVF has the least global fitness however it produces the best quality fitness. The quality fitness measured in wPSNR of values over 42 dB is considered acceptable. All texture metrics produce acceptable quality fitness. The texture metric RS (Rabil *et al.*, 2013a) is well suited for full uneven embedding scheme, where it classifies blocks into different clusters and assign different embedding capacity for each cluster of blocks. However for simple uneven embedding scheme used in this chapter, it produces similar evolution to JND.

3.6 Conclusions and Future Work

Using traditional IW methods to secure facial captures, all the cover image blocks have to be represented in the population of candidate solutions. For high-resolution facial captures, these

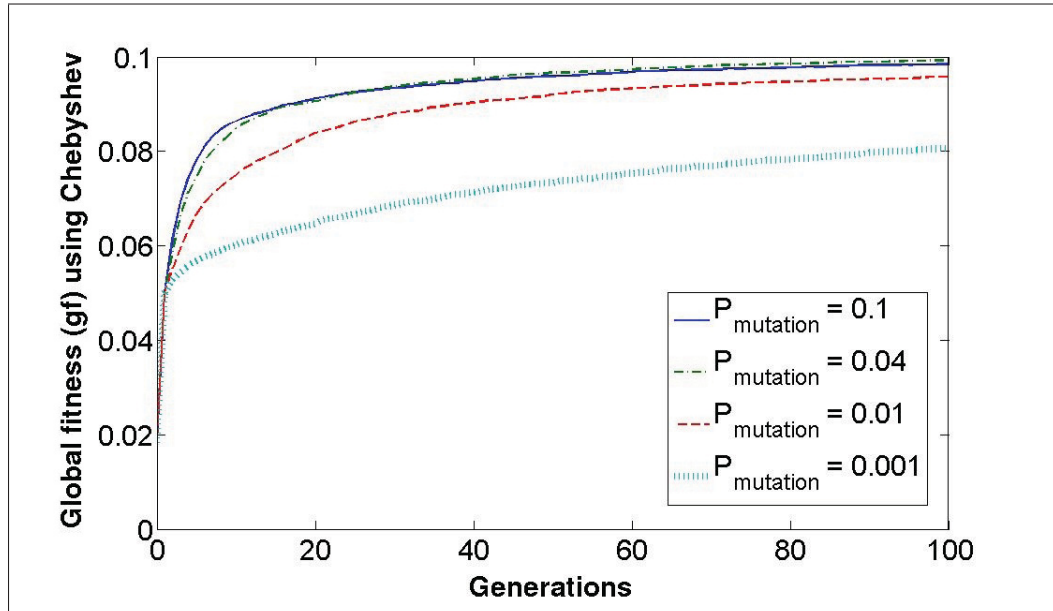


Figure 3.10 Impact of P_{mutation} on BCGA evolution using Chebyshev aggregation for gf as shown in equation 3.1.

Table 3.7 Impact of the perceptual texture mask on BCGA performance using Chebyshev aggregation for gf as shown in equation 3.1.

Texture Metric	wPSNR [dB]	NC1 (JPEG QF=80)	NC2 (Median Filtering)	NC3 (LPF)	gf
Entropy	57.3123	0.9367	0.9270	0.9722	0.0993
JND	57.5036	0.9569	0.9060	0.9659	0.0940
RS (Rabil <i>et al.</i> , 2013a)	57.2707	0.9538	0.9018	0.9636	0.0930
NVF	63.2379	0.9546	0.8859	0.9643	0.0890

methods involves handling an optimization problem with huge search space. This huge search space results in premature convergence for traditional EC algorithms.

In this chapter, an approach is presented to handle optimizing watermark embedding parameters for high-resolution facial captures. Perceptual texture masks are used to select only textured blocks to be represented in EC candidate solutions. Then an application specific GA algorithm based on co-evolution called BCGA is proposed to address this high dimensionality optimization problem. BCGA utilizes local watermarking metric BWM on block level to improve the convergence properties of GA. It performs the traditional GA operators on block

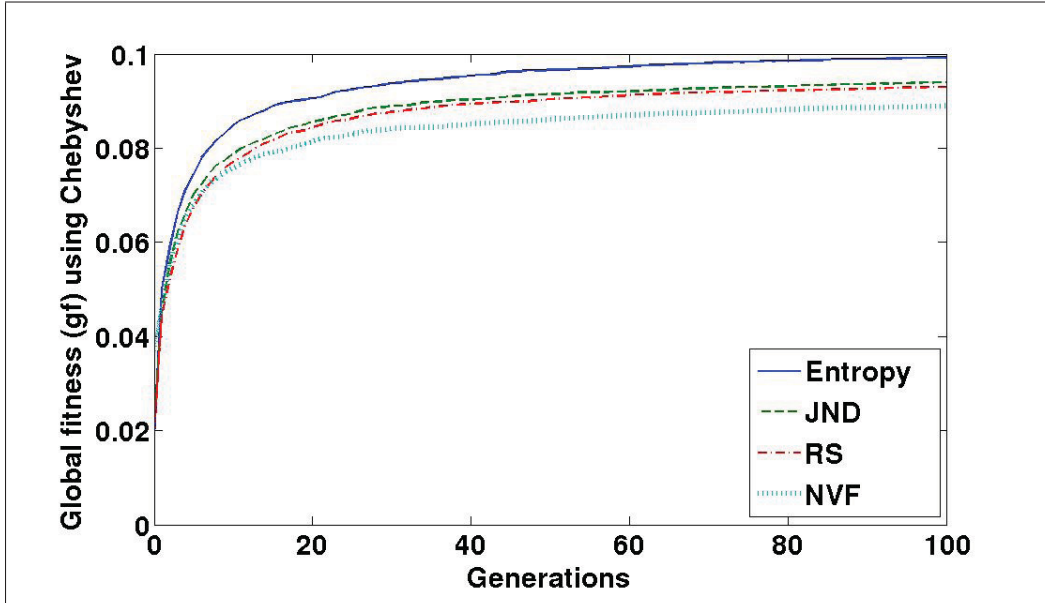


Figure 3.11 Impact of using different grayscale texture metrics as perceptual masks on BCGA evolution using Chebyshev aggregation for gf as shown in equation 3.1.

level rather than the whole candidate solutions. The proposed elitism mechanism assumes that embedding parameters of blocks corresponding to higher local metric are assumed to have better global fitness for the whole image when concatenated.

Simulation results on PUT face images database (Kasinski *et al.*, 2008) suggest that the proposed BCGA provides solutions (embedding parameters) with 17% better fitness compared to that of traditional GA. Indeed, BCGA avoids the premature convergence issues of traditional GA. The convergence for BCGA is assumed in fewer iterations, and thus speeds up the optimization. Chebyshev aggregation provides better quality of solutions compared to weighted sum aggregation. Sensitivity analysis is performed for user defined parameters of BCGA, and different texture metrics used in perceptual texture masks. BCGA evolution is more sensitive to probability of mutation and the texture metrics used for selecting embedding blocks compared to other user-defined parameters.

The approach presented in this chapter can be generalized on high dimensional optimization problems, where local metrics of subcomponents of this high dimension problem is affecting the overall optimization problem. BCGA elitism algorithm can be reused in optimization

algorithms to provide better exploration capabilities for search space, where it implements cooperative coevolution approach at subcomponent level. This can decrease the number of generations required to reach convergence even for lower dimensional problems.

In a future work, standard LSGO algorithms which are tested against optimization problems of 2k variables will be experimented on the IW of facial images using different resolutions. This identifies the search space size limitation for these methods for IW optimization problem. Also more detailed sensitivity analysis will be performed for the impact of BCGA user-defined parameters on the quality of solutions produced using long streams of high-resolution facial image captures.

3.7 Discussion

The proposed algorithm BCGA in this chapter is introducing better quality of solutions by resolving the premature convergence for this large dimension search space. It results in significant speed up for the optimization due to the improved search space exploration capabilities using the proposed elitism mechanism. BCGA is well suited for training phase of the proposed BMRC framework in Chapter 2. The performance of BMRC is boosted from both quality of solutions and computational complexity when using BCGA for full optimization during training phase.

In Chapter 3, the embedding capacity is equal for all textured blocks selected for embedding, and thus the common texture metrics like JND, NVF, and Entropy are used to rank blocks according to their texture metric. In Chapter 2, Robustness Scores (RS) metric is proposed as a grayscale texture metric for full uneven embedding. The blocks of higher texture have higher embedding capacities, and the embedding capacities decrease gradually for lower texture blocks using Algorithm 5.

The proposed RS metric is measured for clusters of blocks rather than individual blocks like standard texture metrics. The blocks of the cover image are clustered using their texture features extracted from the most significant DCT coefficients. The metric RS is calculated for each cluster of blocks by measuring the robustness fitness using NC when embedding 1 bit-per-

block in blocks belonging to this cluster. A threshold α for RS is chosen to avoid embedding in clusters of blocks of RS lower than this threshold. The sensitivity analysis for this threshold α is described in Section 2.5.

The sensitivity analysis for using standard grayscale texture metrics compared to RS on the evolution of BCGA is described in Section 3.5. The evolution of BCGA using RS metric to select embedding blocks is comparable to using JND. The proposed RS is more suitable for full uneven embedding where the texture metrics are calculated for clusters of blocks rather than individual blocks. Annex I complements the experimentation by evaluating standard grayscale metrics for selecting blocks for embedding. Watermark quality and robustness fitness are used for evaluation when using fixed embedding bands for all blocks. The fitness produced in this annex experimentation can be considered as initial solution for optimizing embedding parameters using BCGA for different watermark lengths and embedding capacities. As shown in Figure 3.11, the initial solutions using different metrics are similar using Chebyshev aggregation for watermark quality and robustness.

In Chapter 2 texture features are proposed to be extracted using DCT coefficients which are already available during watermark embedding and extraction. This ensures minimal computational complexity for this operation which is used during generalization phase of BMRC framework. The generalization phase is characterized by low complexity compared to training phase which involves complex optimization problem with huge search space. Using other standard texture metrics for grayscale texture feature extraction would add more complexity which is significant during generalization phase.

During training phase, the proposed BCGA algorithm for optimization use standard grayscale metrics to rank blocks based on their texture. The blocks of higher texture are selected for embedding watermark bits. The complexity introduced to rank blocks based on texture metrics is not significant compared to the complex optimization for embedding parameters for high resolution facial images, where the fitness evaluation is computationally complex.

CONCLUSION

In this thesis intelligent watermarking of streams of grayscale high-resolution facial images was investigated. The main objective of the research conducted was to reduce the computational cost of intelligent watermarking in such scenario. The objective was achieved by tackling first the complexity of the stream of similar optimization problems corresponding to intelligent watermarking of the stream. This is accomplished by optimizing few facial images, and storing optimization results in associative memory for later recall during generalization. Then the complexity of optimizing training facial images is addressed by resolving premature convergence and speeding up optimization.

The first contribution (Chapter 2) introduces BMRC framework which implements multi-hypothesis. The solutions produced using full optimization during training phase are clustered based on their texture features, and stored in associative memory for different number of clusters. During generalization phase, blocks are clustered and solutions are recalled from associative memory for different number of clusters. Solutions produced are ranked based on watermarking fitness, and the optimal number of clusters are selected at the end of generalization. In this chapter also, a new grayscale metric Robustness Scores (RS) is proposed for clusters of blocks. This metric is suitable for full uneven embedding where the embedding capacity is higher for blocks clusters of higher RS, and it decreases gradually for clusters of lower RS. Experimental results on PUT database (Kasinski *et al.*, 2008) indicate significant computational complexity reduction up to 95.5% measured in fitness evaluations.

The second contribution (Chapter 3) introduces a specialized algorithm BCGA to address high dimension optimization problem of intelligent watermarking for high-resolution facial images. This algorithm is based on coevolution on the block level using local watermarking metrics. Experimental results on PUT database (Kasinski *et al.*, 2008) prove that BCGA resolves the premature convergence due to the high dimensionality, and speeds up the optimization by reaching 17% better aggregated fitness in less iterations.

Future research directions:

The following three direction are proposed for future research:

Analysis for over training behavior for BMRC: This behavior is noticed with number of training face images more than three face images. This should be avoided by using a change detection mechanism for the training information stored in associative memory. Such mechanism decides if the system needs further training face images based on the fitness produced by the system.

Improve the clustering algorithm used in clustering face image blocks: This can achieved by adopting incremental clustering when dealing with multiple training face images. K-Means is sensitive to the selection of the initial partition and may converge to a local minimum of the criterion function value if the initial partition is not properly chosen (Jain *et al.*, 1999). Another clustering algorithms other than K-Means is proposed to be evaluated and compared to K-Means used in BMRC, leader clustering (Jain *et al.*, 1999) is proposed to be evaluated for its lower complexity.

Extend the concept presented in BCGA to multi-objective optimization: This will add more challenges to the coevolution and elitism proposed in BCGA. Diversity of solutions shall be preserved for efficiently producing Pareto front of non-dominated solutions. Pareto based fitness assignment proved efficiency with dealing with intelligent watermarking using Multi-Objective PBIL described in Annex II. An external archive is proposed to be used to preserve elite solutions beside the candidate solutions population.

ANNEX I

PERCEPTUAL TEXTURE MASKS EVALUATION

This annex describes some additional experimentation on perceptual texture masks evaluation. The texture masks considered in this experimentation are entropy, NVF and JND. Different grayscale texture metrics considered in this thesis are discussed in Section 1.5. The experimentation includes different watermarks lengths and embedding capacities for these texture masks. These masks are used to select the embedding blocks which are the most textured blocks of the cover image. This experimentation can be used to determine the suitable texture metric to be used for embedding blocks selection based on the watermark quality and robustness priorities. Entropy was used in Chapter 3 for embedding blocks selection because it produces better fitness and evolution compared to other metrics for the watermark length and capacity considered.

The experimentation performed in this annex uses fixed mid-range embedding bands for all blocks for 60 face images, the fitness produced is considered as initial solutions provided to the BCGA optimizer to improve this fitness. As shown in Figure 3.11, where the whole evolution using BCGA is compared for standard metrics with DCT based RS metric, the initial solutions are very similar for all metrics when using Chebyshev aggregation. However the evolution vary for the different metrics. Also Figure 3.11 shows that fitness has been improved significantly in the first few iterations for all metrics, which implies that the produced fitness is less sensitive to initial solutions which can vary slightly from one metric to the other.

Regarding the computational complexity, using standard metrics with additional complexity introduced during the training phase described in Section 2.2 which is characterized by high complexity is not considerable. The full optimization for high resolution facial images takes few hours of processing. On the other size the DCT coefficients are already extracted during watermark embedding and extraction and thus it does not add more complexity to extract the features. This can be more significant during the generalization phase described in Section 2.3 where the unseen facial images go through: feature extraction, associative memory recall

for sub-solutions with minimal complexity, and then fitness evaluations which can be reduced using parallel implementation (GPU). And thus using DCT for feature extraction with no additional complexity would be more computationally efficient specially during generalization.

There is a difference between standard metrics and the proposed RS based on DCT as described in Section 2.2. The standard metrics fit more into ranking the blocks to select the most textured blocks for embedding. This can be suitable in the uneven embedding scheme with smooth areas with no bits embedded and the textured blocks having equal capacity C bits-per-block. It is hard to define thresholds for different embedding capacities based on the texture features. On the other hand RS is calculated for clusters of blocks whose texture features are similar. So it ranks clusters of blocks rather than individual blocks. This makes it easier to have full uneven embedding scheme, where highest textured blocks have maximum embedding capacity C_{Max} , and the capacity decreases for lower textured blocks. Also RS is computationally efficient because it is based on DCT coefficients which are already extracted for watermark embedding and extraction.

1 Experimental Methodology

The database for face images used in experiments is the PUT (Kasinski *et al.*, 2008) face database which consists of 100 individuals with 100 poses for each individual. Sample face images of the first pose of individuals are shown in Figure 3.6. The face images of this database are characterized by having smooth background for all individuals, and different light conditions and clothes. The percentage of background and foreground blocks are nearly the same for each individual, where the different poses represent head rotation and facial expression changes only. This ensures texture homogeneity for facial images to be used as cover images in watermarking application. Color face images of resolution 2048x1536 are converted to grayscale level. The first pose of all individual has the name pattern IIII1001.JPG where IIII is the individual number in 4 digits. Face images of this pose are used for proposed system development, where the individuals starting from 0041 till 0100 are used for perceptual masks evaluation and tuning embedding and extraction algorithm using these 60 face images.

The watermark to be embedded is BancTec binary logo with different resolutions shown in Figure 2.8. The watermark embedding/extracting algorithm used in experiments is a modified algorithm of the one proposed by Shieh *et al* (Shieh *et al.*, 2004) as illustrated in Section 1.2. The metrics used in experimentation for measuring watermark quality and robustness for the whole face image to find optimal bands are wPSNR and NC respectively as defined in Section 1.1. The attacks considered in the experimentation are JPEG compression with quality factor equals to 80%, median filtering, and low pass filtering.

The experiments are executed on gentoo linux based server of 24 GB memory size and 8 cores Intel Xeon CPU X5650 of speed 2.67GHz. The implementation utilizes Graphical Processing Units (GPU) for DCT transform using 8 NVIDIA Tesla C2050/C2070 GPU cards. The proposed methodology consists of two phases to evaluate different components of the proposed system. The experiments evaluates and tunes the perceptual mask component.

During experiments, a set of 60 face images are used to evaluate different perceptual masks and tune watermark embedding and extraction algorithm. The evaluation of perceptual masks to find the most textured blocks TB for embedding is performed in isolation to finding optimal embedding bands for these blocks. Fixed embedding bands are used for embedding the watermark in the most textured blocks, where $eb_i \in [6, 16]$ depending on the embedding capacity.

The first experiment uses entropy texture mask to find the blocks of highest entropy for embedding, and use fixed bands for embedding the same watermark using different embedding capacities. For the same watermark of length $WL = 48k$ using $TB = 48k, 24k, 12k, 6k$, this yields to embedding capacities of $C = 1, 2, 4, 8$ bits-per-block. The first case represents the traditional methods where watermark bits are embedded in all face image blocks without avoiding smooth textured blocks for embedding. This experiment concludes the efficiency of using only textured blocks for embedding for different embedding capacities.

The second experiment evaluates different masks for different watermark lengths WL . The masks considered in this annex are entropy, NVF, and JND. The embedding capacity is fixed to 1 bit-per-block for all masks, and number of textured blocks $TB = 48k, 24k, 12k, 6k, 1.6k$.

This experiment concludes the most efficient mask to find the most textured blocks for embedding for different watermark lengths WL .

During the evaluation of different grayscale metrics, there are number of trade-offs where the decision maker should prioritize the objectives. In this annex, an acceptable level of quality is assumed for wPSNR of value 42 dB and higher. The robustness against JPEG is given higher priority compared to other attacks due to the popularity of JPEG compression in many domains.

2 Results and Discussion

Table I-1 shows the impact of using only textured blocks for embedding on the watermark fitness produced. The watermark quality and robustness fitness is measured using different number of textured blocks TB and subsequently the embedding capacity C , where the watermark length is fixed at $WL = 48$ k-bits. The minimum embedding capacity C is equal to 1 bit-per-block and thus all face image blocks are used for embedding, and the maximum embedding capacity C is equal to 10 bits-per-block and thus only the most textured 10% of the face image blocks are used for embedding. The TB textured blocks are selected using entropy mask.

The watermark quality measured using wPSNR is degrading with increasing embedding capacity. The minimum quality accomplished using 10 bits-per-block measured using wPSNR is more than 42 dB, which is considered acceptable quality. The best robustness fitness against JPEG is produced using embedding capacity 8 bits-per-block, the best robustness fitness against median filtering is produced using 2 bits-per-block, and finally the best robustness fitness against low pass filter is produced using 1 bit-per-block.

Figures I-1 shows the trend of watermark fitness using different empirical embedding capacities C . The degradation of watermark quality and improvement in robustness against JPEG are shown in Figure I-1. There is a turn point values for both quality and robustness against JPEG at embedding capacity 4 bits-per-block. The slope continues slightly till embedding capacity 8 bits-per-block till saturation. The quality fitness for all embedding capacities is over 42 dB which is considered acceptable quality. For robustness against MF shown in Figure I-1. There

Tableau-A I-1 Embedding a watermark of length $WL = 48.8k$ using different number of textured blocks TB selected using entropy mask.

TB	C	wPSNR [dB]	NC1 (JPEG QF=80)	NC2 (Median Filtering)	NC3 (LPF)
48k	1	54.53±1.04	0.8880±0.0331	0.5080±0.0035	0.7233±0.0083
24k	2	51.59±1.10	0.9456±0.0122	0.5715±0.0044	0.6919±0.0100
12k	4	49.05±1.23	0.9749±0.0059	0.5622±0.0076	0.6419±0.0108
8k	6	48.60±1.24	0.9759±0.0053	0.5323±0.0044	0.6470±0.0063
6k	8	48.10±1.22	0.9773±0.0052	0.5511±0.0043	0.6466±0.0053
4.8k	10	48.04±1.18	0.9767±0.0041	0.5356±0.0042	0.6429±0.0043

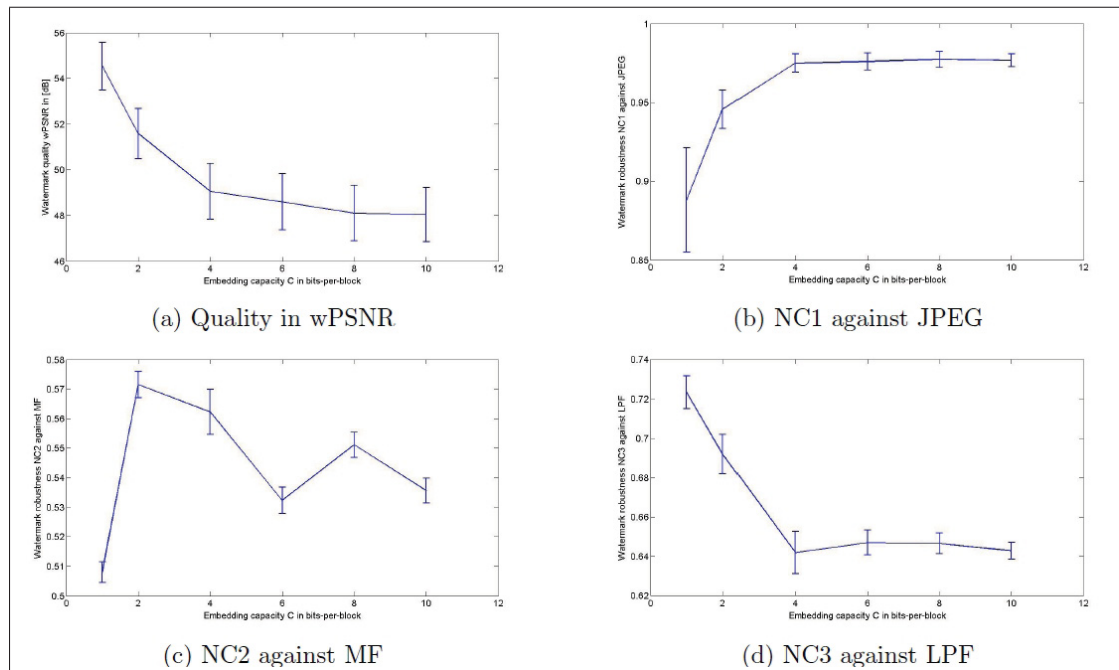


Figure-A I-1 Watermark fitness for different embedding capacities C and number of embedding textured blocks TB .

is a degradation trend of robustness fitness starting 2 till 6 bits-per-block, and then this trend continues after 8 bits-per-block. And finally for the robustness against LPF shown in Figure I-1, there is degradation trend till 4 bits-block, and then almost saturation after 4 bits-per-block.

Table I-2 shows the watermark fitness produced when using different texture masks for selecting the most textured blocks for embedding. The embedding capacity C is fixed at 4 bits-per-

block, and the watermark length WL is varying. The best robustness fitness against JPEG is accomplished when using entropy mask for selecting the most textured blocks for embedding for watermarks of different lengths. The least complexity mask is JND and the mask of the most complexity is entropy mask, however the time complexity of the masks considered are within the same range.

Tableau-A I-2 Embedding 4 bits-per-block for different watermark length WL .

Mask	wPSNR [dB]	NC1 (JPEG QF=80)	NC2 (Median Filtering)	NC3 (LPF)
$WL = 48k, TB = 12k$				
Entropy	49.05±1.23	0.9749±0.0059	0.5622±0.0076	0.6419±0.0108
NVF	55.10±1.95	0.9199±0.0167	0.5982±0.0039	0.7263±0.0106
JND	50.86±1.47	0.9617±0.0069	0.5721±0.0059	0.6651±0.0101
$WL = 24k, TB = 6k$				
Entropy	49.96±1.29	0.9841±0.0037	0.5554±0.0071	0.6241±0.0081
NVF	54.39±1.96	0.9561±0.0098	0.5860±0.0093	0.6848±0.0162
JND	53.60±1.68	0.9687±0.0056	0.5736±0.0069	0.6581±0.0109
$WL = 12k, TB = 3k$				
Entropy	51.40±1.36	0.9882±0.0031	0.5475±0.0080	0.6134±0.0077
NVF	54.60±1.71	0.9802±0.0044	0.5598±0.0083	0.6379±0.0073
JND	56.08±1.81	0.9717±0.0060	0.5755±0.0086	0.6554±0.0125
$WL = 6k, TB = 1.5k$				
Entropy	53.23±1.41	0.9905±0.0022	0.5395±0.0101	0.6035±0.0097
NVF	55.96±1.79	0.9864±0.0033	0.5478±0.0093	0.6232±0.0086
JND	58.76±1.86	0.9739±0.0066	0.5762±0.0103	0.6524±0.0137

Figure I-2 shows the best fitness produced using different texture masks for watermarks of different lengths WL using embedding capacity C equals to 4 bits-per-block. Figure I-2 shows the best fitness for watermark quality using JND mask for watermarks of length WL less than 24 k-bits, and NVF mask for watermarks of length more than 24 k-bits. The robustness against MF shown in Figure I-2 has the best fitness using JND mask for watermarks of length WL less than 24 k-bit, and using NVF for larger watermark lengths. And finally Figure I-2 shows the best robustness fitness against LPF using JND mask for watermarks of length WL less than 24 k-bit, and using NVF for larger watermark lengths.

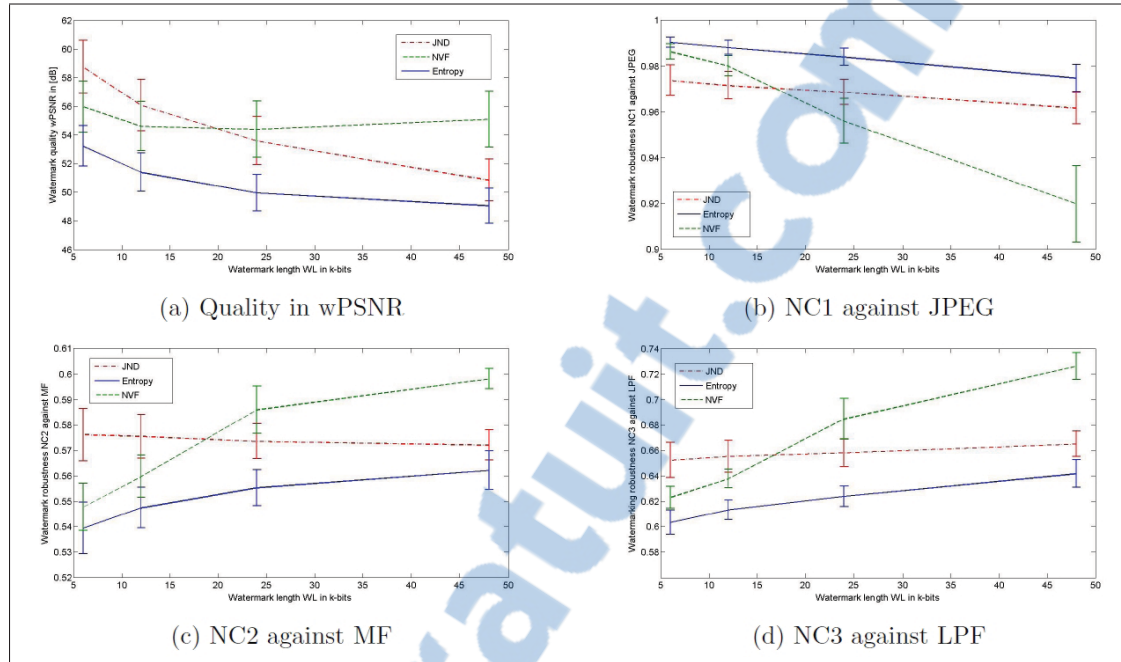


Figure-A I-2 Watermark fitness for different watermark lengths using different texture masks.

Figure I-3 shows the most textured blocks used for embedding using different texture masks. The selected blocks for NVF and entropy masks are similar specially for larger number of textured blocks TB , however NVF mask is more accurate in detecting the edges and the most textured blocks specially for $TB = 1.6k$. JND mask for smaller number of TB is able to detect the blocks where the changes are not introducing significant distortion rather than the edges of the face image.

Table I-3 shows the watermark fitness produced for different texture masks for a watermark of fixed length $WL = 24k$ using different embedding capacities C . The best watermark quality fitness is produced using NVF mask for all embedding capacities ranging from 2 till 8 bits-per-block. Also the best robustness against MF and LPF is produced using NVF mask for different embedding capacities C . Still the best fitness for robustness against JPEG is produced using entropy mask for all embedding capacities. Figure I-4 shows the watermark quality and robustness fitness produced using different masks with different embedding capacities. The time complexity of JND and NVF is comparable with entropy, however using entropy results

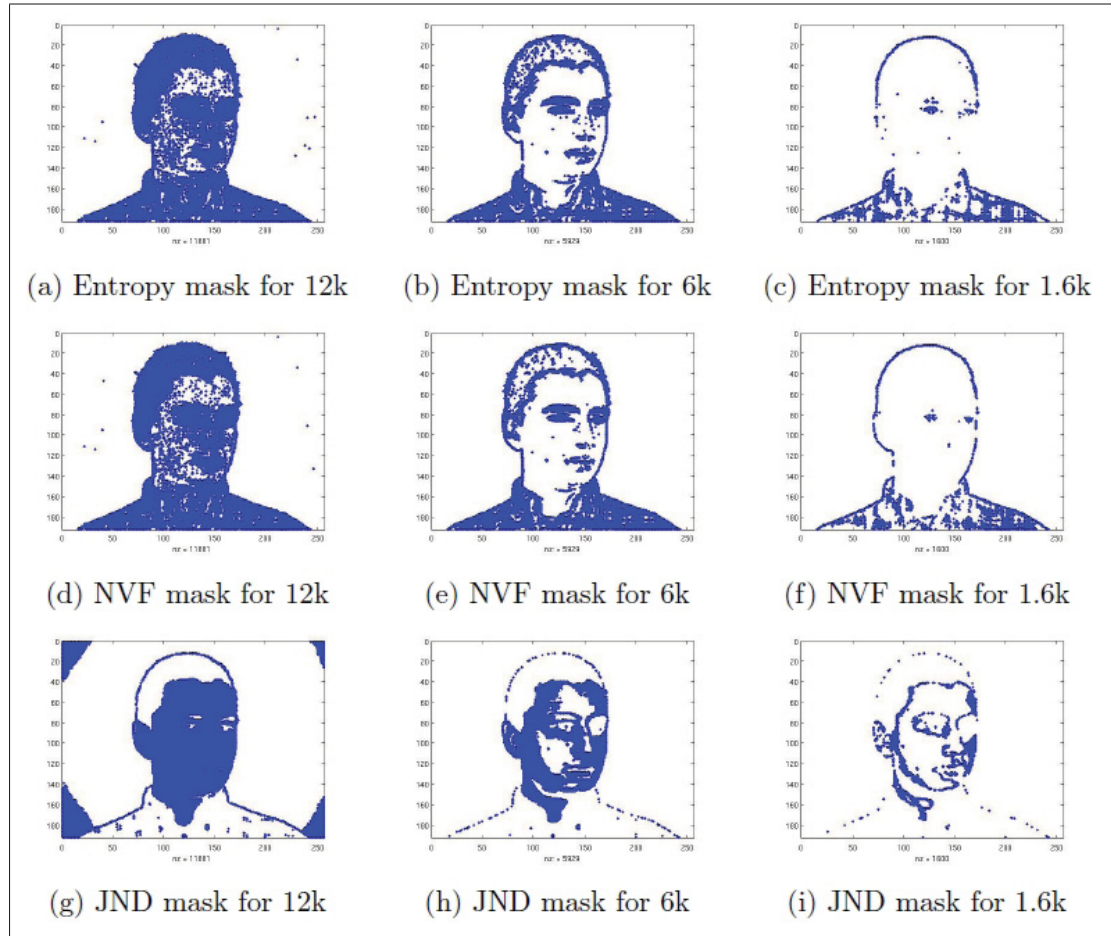


Figure-A I-3 Blocks selected for embedding using Entropy for 12k blocks

in better robustness against JPEG which is the most common attack while exchanging the information via Internet.

Tableau-A I-3 Embedding watermark of 24k using different embedding capacities C bits-per-block.

Mask	wPSNR [dB]	NC1 (JPEG QF=80)	NC2 (Median Filtering)	NC3 (LPF)
$C = 2, TB = 12k$				
Entropy	52.07±1.19	0.9726±0.0082	0.5648±0.0053	0.6569±0.0106
NVF	54.62±1.54	0.9479±0.0147	0.5787±0.0054	0.6930±0.0111
JND	54.03±1.55	0.9557±0.0103	0.5701±0.0061	0.6795±0.0105
$C = 4, TB = 6k$				
Entropy	49.96±1.29	0.9841±0.0037	0.5554±0.0071	0.6241±0.0081
NVF	54.39±1.96	0.9561±0.0098	0.5860±0.0093	0.6848±0.0162
JND	53.61±1.68	0.9687±0.0056	0.5736±0.0069	0.6581±0.0109
$C = 6, TB = 4k$				
Entropy	49.91±1.31	0.9831±0.0038	0.5308±0.0055	0.6374±0.0051
NVF	54.97±2.15	0.9507±0.0104	0.5452±0.0064	0.6869±0.0142
JND	53.99±1.66	0.9656±0.0062	0.5418±0.0067	0.6644±0.0096
$C = 8, TB = 3k$				
Entropy	49.61±1.30	0.9833±0.0039	0.5453±0.0051	0.6385±0.0048
NVF	55.54±2.46	0.9458±0.0119	0.5636±0.0059	0.6861±0.0141
JND	54.10±1.66	0.9640±0.0064	0.5645±0.0058	0.6629±0.0082

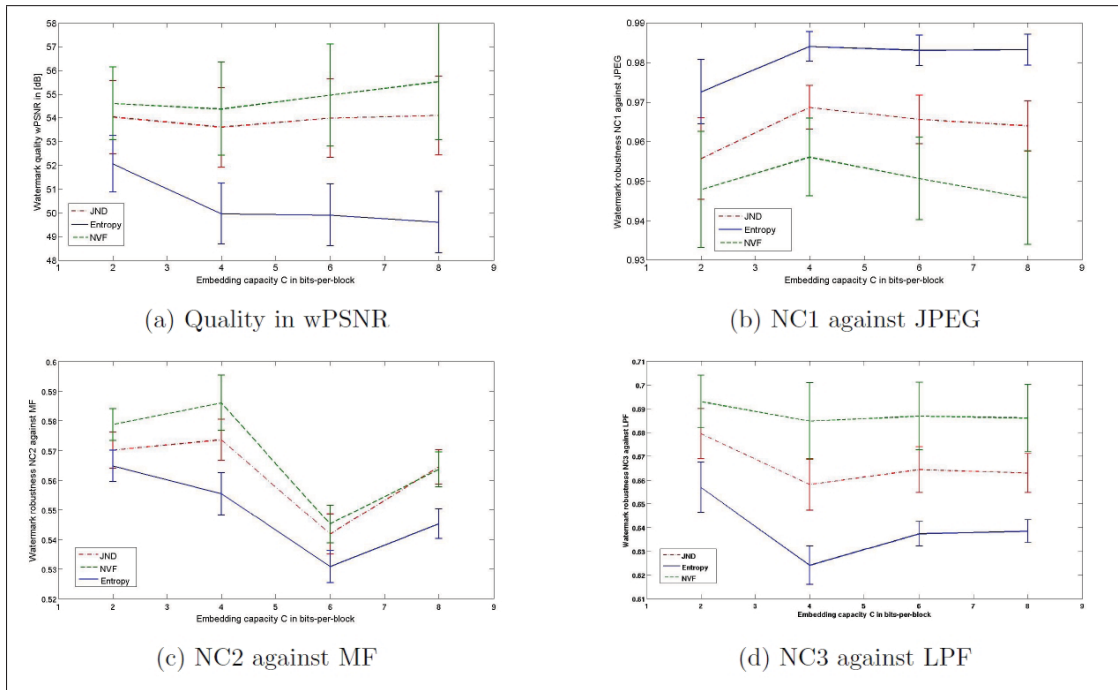


Figure-A I-4 Watermark fitness for different texture masks using different embedding capacities C for watermark of length $WL = 24k$.

ANNEX II

BIO-WATERMARKING SYSTEMS USING OFFLINE SIGNATURE FEATURES

This annex introduces a proposed application for access control domain to verify individuals crossing borders using their facial captures and the offline signature captured from signed forms. The offline signature features are discretized and embedded as binary watermarks in passport facial captures. This approach significantly improves verification rates by combining physical and behavioral biometric traits.

The impact of watermarking attacks on biometric verification system is important specially if the watermark represents biometric trait where the performance of the overall system is affected when integrating biometric verification system with bio-watermarking framework. There is a trade-off between quality and robustness fitness, and the performance of the verification system because adding more bits improving the accuracy of discretization or increasing the size of feature vectors improves the performance of the biometric verification system, however this degrades the fitness for both quality and robustness due to increasing the embedding capacity.

In this annex, we present a study for the impact of watermarking attacks on biometric offline signature verification systems and its relation with the impact on the quality and robustness fitness of the watermark. Offline signature feature vectors are discretized into binary stream, and embedded as binary watermarks into grayscale face images of high-resolution which have high embedding capacity and better face recognition verification rates. Watermark embedding is optimized using computational intelligence optimization method which is Population Based Incremental Learning (PBIL) for faster convergence as it employs previous optimization experience in subsequent generations. The impact of quantization and watermarking attack of different intensities is considered. Also the impact of using only region of interest (ROI) of higher texture for embedding watermarks rather than other smooth texture areas is studied for this class of face grayscale images. Also recommendations for proposed research directions to improve the overall intelligent bio-watermarking system for offline signature are described at the end of this annex.

The proposed intelligent bio-watermarking system for offline signature can be used to verify individuals crossing borders using their face images with their offline signatures features embedded as invisible watermarks, and verified from border filled forms. Also the proposed system can be used to protect biometric templates for both cover images representing face images, and the offline signature features which is embedded as invisible watermark where intruders are not aware of existing embedded features given good watermark embedding quality with minimal visual distortion.

1 Intelligent Bio-Watermarking with Multi-Objective Population Based Incremental Learning Using Offline Signature Feature Vectors

Biometric watermarking was first proposed by Jain in 2002, where Jain and Uludag (Jain and Uludag, 2003) suggested an amplitude modulation-based biometric watermarking. Bio-watermarking systems can be categorized into the following categories (Low *et al.*, 2009):

- a. *Watermarked biometrics*: For watermarked biometrics the host is a biometrics, whereas the watermark can either be a biometrics, or other proprietary notice. Intruders might not be aware of invisibly embedded traits.
- b. *Biometric watermarking*: For biometric watermarking the biometrics are employed as the watermark, whereas the host can be any copyrighted documents.

In this annex, the proposed bio-watermarking is considered of the first category where the cover image is representing biometric face template, and the watermark is extracted features of another biometric trait which is offline signature.

Bureerat and Sriworamas (Bureerat and Sriworamas, 2007) proposed changes to PBIL algorithm to handle multi-objective optimization problems. In this algorithm the probability vector is replaced with probability matrix, where each row in this matrix represents the probability vector to create sub-population of individuals. An external archive is proposed to be used to store the non-dominated solutions found throughout iterations. The detailed algorithm is described in Algorithm 10. An external archive is proposed to be used to store the non-dominated

solutions found throughout iterations. While updating each row of the probability matrix, m weighting factors are generated randomly where the sum of weights equals to '1', the binary solution from union set of current population and external Pareto set which gives the minimum weighted sum using the m weights is chosen to update the row probability vector.

Algorithm 10 Multi-objective PBIL adapted from (Bureerat and Sriworamas, 2007).

- 1: Initialize empty external Pareto set, and probability matrix whose elements are equal to '0.5'
 - 2: **while** Iteration < Max Iteration **do**
 - 3: Generate a sub population with each row in probability matrix.
 - 4: **while** Sub-population < Max number of sub-populations **do**
 - 5: Evaluate the corresponding objective values to the generated populations.
 - 6: Take non-dominated members sorted from the union set of the current population and the old external Pareto set as new external Pareto set, if the external Pareto set is full, remove some solutions using adaptive grid algorithm where members in most crowded solution regions is removed iteratively.
 - 7: In updating each row of the probability matrix, generate m weighting factors randomly where the sum of weights equals to '1', the binary solution from union set of current population and external Pareto set which gives the minimum weighted sum using the m weights is chosen to update the row probability vector
 - 8: **end while**
 - 9: The probability matrix and external Pareto set are improved iteratively until stopping criteria is reached.
 - 10: **end while**
-

The multi-objective PBIL component of the proposed bio-watermarking system for offline signature is shown in Figure 1.4, where quality objective and robustness against different attacks objectives are optimized simultaneously without favoring one objective over the other. This proposed mechanism provides multiple non-dominated solutions (Pareto front), which helps the operator to tune the watermarking system to be robust against certain attacks without computationally expensive re-optimization. This can be easily accomplished by choosing the best solution with regards to robustness against certain attack to be the optimal embedding to be used for the digitized document images.

The most significant advantage of handwritten offline signature over other biometric attributes is that it has traditionally been used for authenticating official documents and thus it is socially accepted and widely used in many domains. Extended Shadow Code (ESC) (Sabourin, 1997) is a global shape factor for offline signature which is used in the signature verification problem

because it permits the local projection of the handwriting without losing the knowledge of the location of measurements in the 2D space as shown in Figure II-1. That is why ESC seems to be a good compromise between global features related to the general aspect of the signature, and local features related to measurements taken on specific parts of the signature without requiring the low-level segmentation of the handwriting into primitives. ESC resolution represents number of row cells and column cells respectively used for projection, for example in Figure II-1 ESC resolution is 3x3.

Extended Shadow Codes (ESC) consists in the superposition of bar mask array over the binary image of a handwritten signature as depicted by Figure II-2. Each bar is assumed to be a light detector related to a spatially constrained area of the 2D signal. A shadow projection is defined as the simultaneous projection of each black pixel into its closest horizontal, vertical and diagonal bars. A projected shadow turns on a set of bits distributed uniformly along the bars. After all the pixels of a signature are projected, the number of on bits in each bar is counted and normalized to the range of [0, 1] before features are extracted. Given a virtual grid composed of I rows by J columns, the cardinality of the ESC feature vector is calculated using equation A II-1. For example resolution 2x3 which is considered in this paper corresponds to cardinality equals to 29.

$$\text{Cardinality} = 4IJ + I + J \quad (\text{A II-1})$$

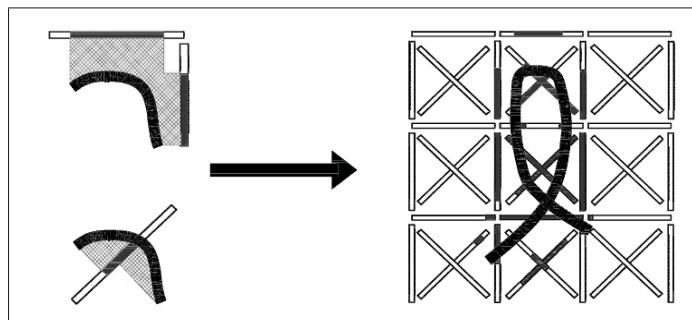


Figure-A II-1 Projection definition in Extended Shadow Code (Sabourin, 1997).

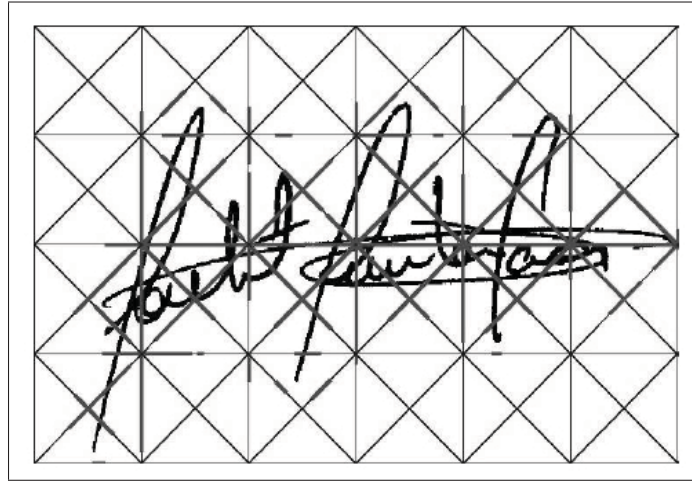


Figure-A II-2 Extended Shadow Code (ESC) for offline signature binary image (Bertolini *et al.*, 2010).

Extended Shadow Code (ESC) features are discretized to be converted to binary stream to be embedded as binary watermarks. The more bits used for discretization, the more accuracy is achieved for biometric verification, on the other hand this degrades the fitness for watermark quality and robustness by adding more bits to be embedded in the payload increasing the watermark capacity. Other authors (Haouzia and Noumeir, 2008) described using cryptographic digital signature as watermarks. Digital signatures aims to sign a document in its electronic form, such that the signature can be transmitted electronically with the signed documents. One of the methods to add the digital signature to the digitized document is discretizing the signature into binary watermarks to be embedded in original documents to be signed. However in this paper, the main focus will be on offline signature rather than digital signature and use extracted features as binary watermarks.

2 Experimental Methodology

The bio-watermarking system used in experiments is shown in Figure II-3, where the embedding optimization module is using multi-objective PBIL algorithm proposed by Bureerat and Sriwornamas (Bureerat and Sriwornamas, 2007) and the offline signature verification system proposed by Bertolini *et al* (Bertolini *et al.*, 2010). The databases used are PUT face database

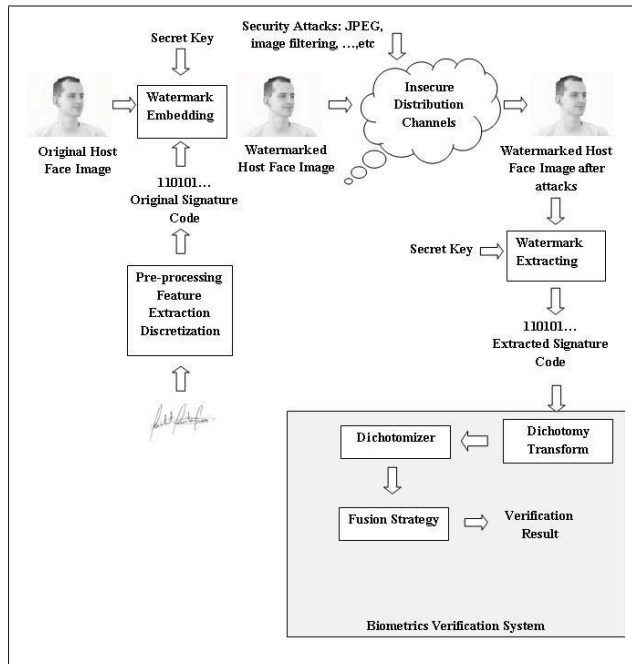


Figure-A II-3 Block diagram for bio-watermarking system with integrated verification system used in experiments
(Low *et al.*, 2009; Kasinski *et al.*, 2008; Bertolini *et al.*, 2010).

(Kasinski *et al.*, 2008), and offline signature database proposed by Bertolini *et al* (Bertolini *et al.*, 2010).

The offline signature verification system proposed by Bertolini *et al* (Bertolini *et al.*, 2010) deploys multi-modal verification and decision fusion for the biometric trait achieving better performance than uni-modal verification systems. It also addresses the challenges of offline signature verification systems which are: the large number of users, the large number of features, the limited number of reference signatures for training, the high intrapersonal variability of the signatures and the unavailability of forgeries as counterexamples.

Binary representation is used to encode individuals, where each individual have binary representation of embedding frequency bands per image 8x8 block multiplied by number of blocks in the host image. Each frequency band is represented by 6-bits to identify the AC frequency coefficient (0-63) to embed the watermark in. The optimization problem to maximize quality and robustness fitness has high dimension as the metrics of both quality and robustness are

calculated for the whole image while being dependent on the selection of embedding bands in each and every image block which have to be represented in each optimization individual. The best non-dominated solution from Pareto front with regards to robustness fitness is chosen as the optimal solution representing the optimal embedding bands for the host image.



Figure-A II-4 Face images used in experiments as host grayscale images (Kasinski *et al.*, 2008) of resolution 2048x1536.



Figure-A II-5 Signatures used in experiments whose feature vectors are embedded as watermarks (Bertolini *et al.*, 2010).

In the experiments the face images used as host images are converted to grayscale, and then downsized 50% to resolution 1024x768. The offline signatures feature vectors used in experiments as watermarks are Extended Shadow Code (ESC) features extracted by the system proposed by Bertolini *et al* (Bertolini *et al.*, 2010). ESC feature vectors of resolution 2x3 producing feature vectors of size 29 features are discretized using different number of bits to be used as binary watermarks with different accuracy levels.

The face images and signatures used in experiments are shown in Figures II-4 and II-5. The first 10 writers signatures feature vectors from the signatures database are embedded as watermarks in the first 10 persons face images from face images database respectively. The verification for the signature is performed against 39 signatures of the same writer as genuine samples and 39 signatures of random forgeries from other writers as negative samples for simplicity. Then the Genuine Accept Rate (GAR) and False Accept Rate (FAR) are calculated as metrics to measure the performance of the biometric verification system along with Equal Error Rate (EER), and Area Under Curve (AUC) at FAR equals to 0.05 to consider the system performance with maximum 5% falsely accepted samples. Receiver Operating Characteristics (ROC) curve is used to represent these metrics graphically.

First experiment shows the impact of discretization of ESC feature vectors on the biometric verification system, where different accuracy levels are experimented for discretizing feature vectors into binary string to be embedded as binary watermark into grayscale face images. In this experiment, the impact on the biometric verification system represented graphically using DET (Detection Error Trade-off) curve is studied for discretization using 6, 8, 10, and 12 bits. From this experiment, the best number of bits needed for discretization to be used in subsequent experiments is concluded. In this experiment, neither watermarking nor attacks is involved to isolate the impact of the quantization only.

Second experiment shows the impact of JPEG compression attack with different quality factors 100%, 80% and 60% on the biometric verification system along with the impact on the watermark fitness for both quality and robustness.

In this application domain, the embedded features bits could be less than the number of image blocks, and consequently the embedding algorithm has to take into consideration the choice of suitable image blocks for embedding. And thus, the third experiment shows the impact of JPEG compression attack of quality factor 80% on the verification system when embedding the watermark in the foreground which is the profile of the face image rather than the smooth background to maximize the watermark fitness. A simple region of interest used for embedding the watermark at the bottom of the foreground as shown in Figure II-6.

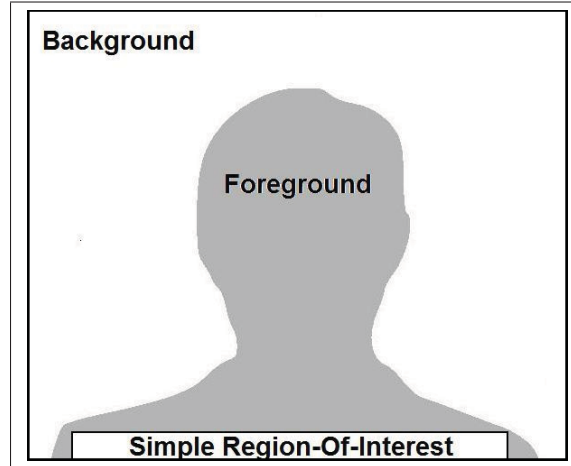


Figure-A II-6 Region Of Interest (ROI) used for experiments for face grayscale images.

3 Results and Analysis

Figure II-7 shows graphically the impact of quantization on the biometric verification system performance for ESC of resolution 2×3 feature vectors using 6, 8, 10, and 12 bits. The results show minimal impact for quantization on the biometric verification system performance, also the results show that using 10 bits have the most similar DET curve to that of using no quantization. Using more than 10 bits have nearly identical DET curve of using 10 bits. The area under the curve for ROC curves when FAR=0.05 equals to 0.0481, 0.0479, 0.0480, 0.0480, and 0.0480 for no quantization, 6, 8, 10, and 12-bits quantization respectively.

Table II-1 shows the impact of watermark attack of different intensities on the quality and robustness fitness and verification system performance, when embedding ESC feature vectors of resolution 2×3 using 10 bits for discretization. The impact on the biometric verification system efficiency is graphically represented using ROC curve in Figure II-8, where the area under the curve for ROC curves when FAR=0.05 equals to 0.0451, 0.0433, and 0.0424 for quality factors 100%, 80%, and 60% respectively. The results show that the fitness changes for quality and robustness are minimal, while the impact on the verification system performance represented by AUC and EER is more significant. This proves the importance of this study.

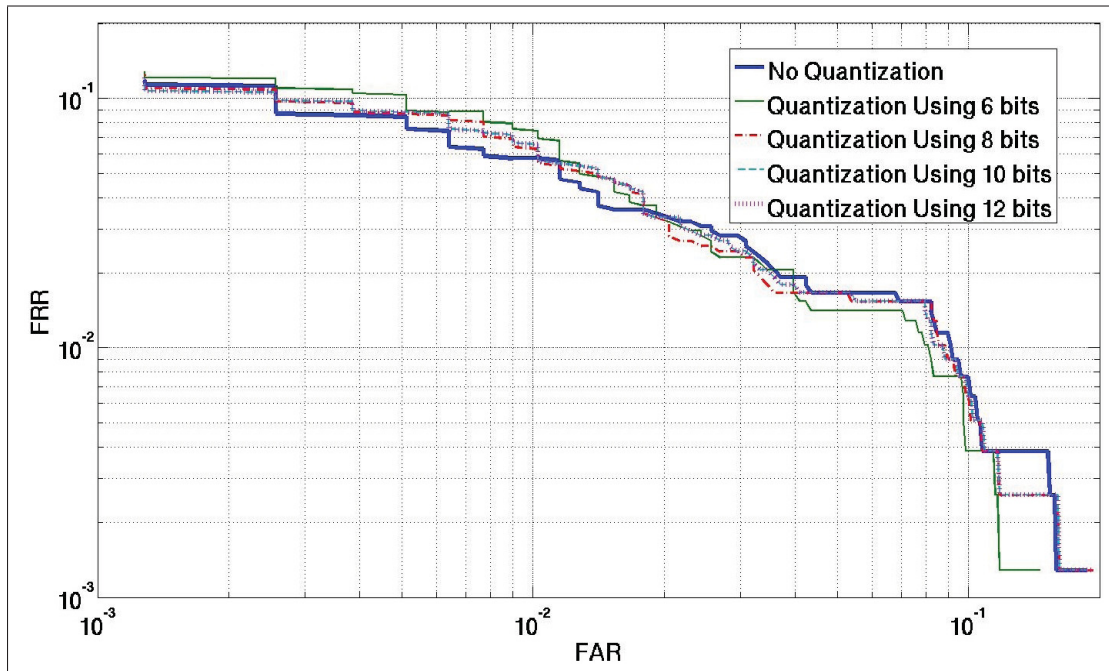


Figure-A II-7 DET curve for quantization of ESC 2x3 feature vectors using 6, 8, 10, and 12 bits representing the impact on the biometric verification system for different accuracy levels.

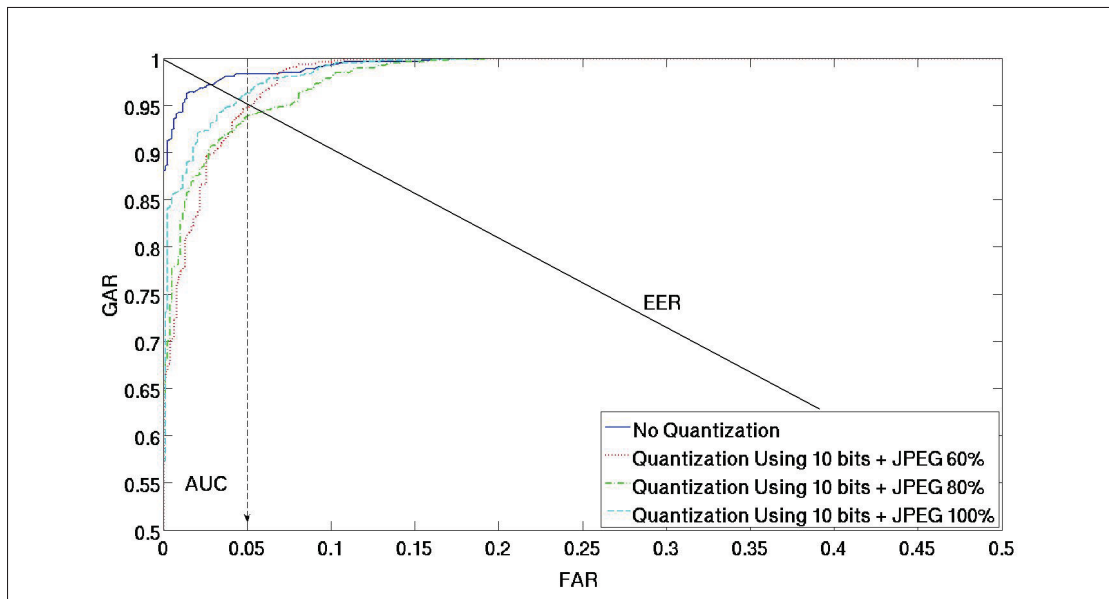


Figure-A II-8 ROC curve for JPEG compression attack impact with different quality factors when embedding 2x3 feature vectors using 10 bits for discretization.

Tableau-A II-1 JPEG compression attack impact with different quality factors when embedding ESC 2x3 feature vectors using 10 bits for discretization.

Q.F.	100%		80%		60%	
Fitness /User	PSNR	NC	PSNR	NC	PSNR	NC
1	86.1	0.99	84.1	1.0	82.8	0.99
2	84.5	1.0	84.3	1.0	84.7	1.0
3	83.9	1.0	82.3	1.0	80.8	1.0
4	83.5	1.0	83.4	1.0	83.8	1.0
5	85.1	0.96	86.9	0.96	85.9	0.95
6	82.8	1.0	85.5	1.0	81.8	1.0
7	82.6	1.0	82.0	1.0	82.3	1.0
8	84.9	1.0	87.4	0.99	85.6	1.0
9	83.9	1.0	84.0	1.0	86.3	0.98
10	83.2	1.0	83.7	1.0	82.6	1.0
Mean	84.1	1.0	84.3	1.0	83.7	0.99
EER	0.0436		0.0577		0.0526	
AUC_0.05	0.0451		0.0433		0.0424	
No Quantization + No Attack						
EER=0.0282 and AUC_0.05=0.0481						

Table II-2 shows the impact of using region of interest to embed the watermark on the quality and robustness fitness and verification system performance, when embedding ESC feature vectors of resolution 2x3 using 10 bits for discretization. The impact on the biometric verification system efficiency is graphically represented using ROC curve in Figure II-9, where the area under the curve for ROC curves when FAR=0.05 equals to 0.0433, and 0.0450 for embedding the watermark in random region, and embedding in ROI respectively. The results show slight improvement in the fitness for both quality and robustness and significant improvement for verification system performance represented by AUC and EER. Also the results show that the extracted watermark is affected by the watermarking attack for few users, meanwhile the impact of the attack is significant on the verification system performance compared to the effect of the discretization process, this means that the affected bits of the watermark were towards the Most Significant Bits (MSB), and thus the impact was severe on the values of feature vectors which degrades the verification system performance as Normalized Correlation (NC) representing the robustness is concerned with only the percentage of correct bits.

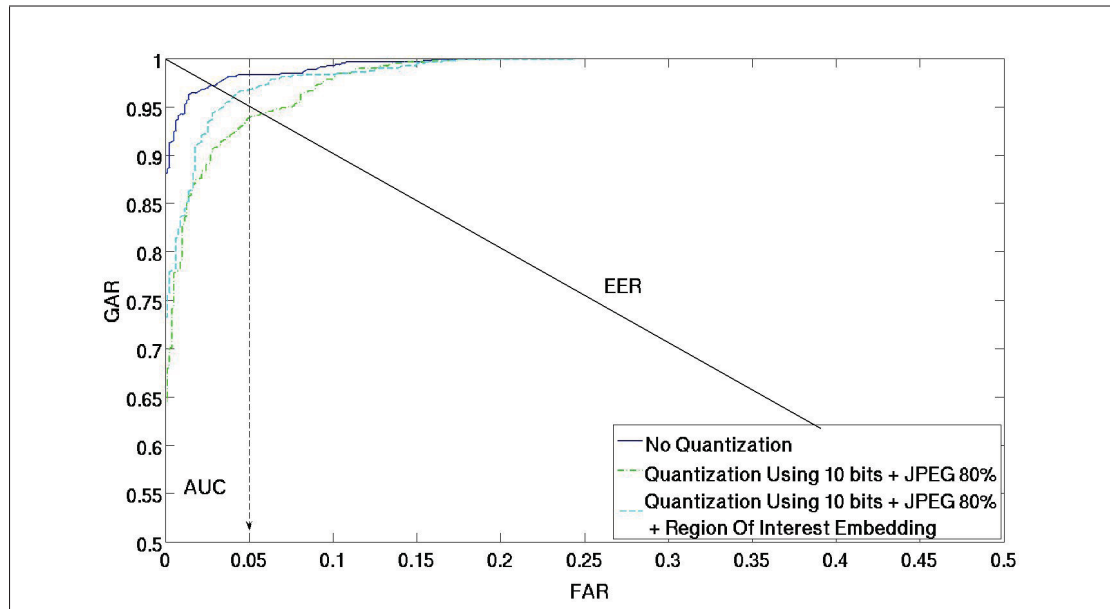


Figure-A II-9 ROC curve for the impact of JPEG compression attack with quality factor 80 while embedding the watermark in region of interest of higher texture when embedding ESC 2x3 feature vectors using 10 bits for discretization.

4 Conclusions and Recommendations

This annex presented a study for watermarking attacks on biometric offline signatures verification in intelligent bio-watermarking systems. Experiments clarified that using watermark fitness metrics is not sufficient to measure the overall performance of the bio-watermarking system. Also experiments showed minimal impact of quantization process on the biometric offline signature verification system, and the optimal quantization is achieved using 10 bits. The impact of watermark attack JPEG compression of different quality factors on the watermark fitness and the biometric verification system has been studied and the least impact on offline signature verification system is noticed for quality factor 100% and the most for 60%. Using region of interest of face images for embedding the watermark has proved better performance for the biometric offline signature verification system.

To improve the efficiency of the proposed intelligent bio-watermarking system using offline signature as embedded biometric trait, the following directions are proposed:

Tableau-A II-2 Impact of JPEG compression attack with quality factor 80% while embedding the watermark in Region Of Interest when embedding ESC 2x3 feature vectors using 10 bits for discretization.

ROI	No		Yes	
Fitness /User	PSNR	NC	PSNR	NC
1	84.1	1.0	82.0	1.0
2	84.3	1.0	85.2	1.0
3	82.3	1.0	85.3	1.0
4	83.4	1.0	84.7	1.0
5	86.9	0.96	86.6	0.96
6	85.5	1.0	83.5	1.0
7	82.0	1.0	82.6	1.0
8	87.4	0.99	85.3	1.0
9	84.0	1.0	84.4	1.0
10	83.7	1.0	85.4	1.0
Mean	84.3	1.0	84.5	1.0
EER	0.0577		0.0397	
AUC_0.05	0.0433		0.0450	
No Quantization + No Attack				
EER=0.0282 and AUC_0.05=0.0481				

- a. *More robust signature verification* : In the experiments, simplified experimental protocol for signature verification is used as a proof of concept. More robust results can be achieved when considering other types of forgeries like simple and skilled forgeries, use selected features from multiple ESC resolutions for more discriminant extracted features, consider the impact of the watermarking on all genuine signatures for writer instead of only one genuine reference signature, and finally increasing the number of writers for better generalization.
- b. *Embed more than one reference signature into face images*: In the experiments only one reference feature vectors per writer has been embedded in grayscale high-resolution face images. The performance of the verification system could be dramatically improved if more than reference signature for the same writer are embedded in the same face image. For better verification results with more discriminant features accomplished by several

- d. *Better formulation for watermark embedding optimization problem:* In the experiments it was clearly demonstrated that in most of the cases the watermark fitness including PSNR and NC, and biometric verification system performance are conflicting objectives which makes it hard to find a compromise between them.

The traditional formulation is not considering the impact on the biometric verification system during the embedding optimization phase, and this degrades the verification rate because NC is representing only the percentage of correct bits in the extracted watermark representing feature vectors of the biometric trait. The verification system is more sensitive to the position of the correct bits in the extracted watermark rather than the percentage. For example, the affected LSB will have minimal impact on the verification system, and vice versa with MSB. In the case of other encoding scheme like Gray code encoding, the sensitivity of the verification system is towards discriminant features bits.

To resolve this issue, it is proposed to add verification system fitness to the optimization process beside quality and robustness objectives as shown in Figure II-10. The fitness could be represented by the confidence value/score of the verification system. In this proposed formulation robustness fitness and verification fitness will complement each other as the robustness fitness minimizes the number of affected bits and meanwhile the verification fitness optimizes the position of affected bits (if any).

BIBLIOGRAPHY

- Ahonen, Timo, Abdenour Hadid, and Matti Pietikainen. 2006. "Face description with local binary patterns: Application to face recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, n° 12, p. 2037-2041.
- Areef, T. E., H. S. Heniedy, and O. M. O. Mansour. December 2005. "Optimal Transform Domain Watermark Embedding via Genetic Algorithms". In *Information and Communication Technology - ITI 3rd International Conference*. (Cairo, Egypt 2005), p. 617-621.
- Aslantas, Veysel, Saban Ozer, and Serkan Ozturk. April 2008. "A Novel Fragile Watermarking Based on Particle Swarm Optimization". In *IEEE International Conference on Multimedia and Expo (ICME 2008)*. (Hannover, Germany 2008), p. 269-273.
- Baluja, Shumeet. 1994. *Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning*. Technical report. CMU.
- Barni, M., F. Bartolini, A. De Rosa, and A. Piva. 1999. "Capacity of the watermark channel: how many bits can be hidden within a digital image?". In *Security and Watermarking of Multimedia Contents*, vol. 3657, p. 437-448.
- Bergh, Frans Van Den. 2002. "An analysis of particle swarm optimizers". PhD thesis, Dept. Computer Science, Univesrity of Pretoria, Pretoria, South Africa.
- Bergh, Frans Van Den and Andries P. Engelbrecht. 2004. "A cooperative approach to particle swarm optimization". *IEEE Transactions on Evolutionary Computation*, vol. 8, n° 3, p. 225-239.
- Bertolini, Diego, Luiz S. Oliveira, Edson Justino, and Robert Sabourin. January 2010. "Reducing Forgeries in Writer-independent Off-line Signature verification through ensemble of classifiers". *Pattern Recognition*, vol. 43, n° 1, p. 387-396.
- Bhattacharyya, Debnath, Samir Kumar Bandyopadhyay, and Poulami Das. July 2007. "Handwritten Signature Watermarking and Extraction Technique Inspired by Principle of Segregation". *International Journal of Security and Its Application*, vol. 1, n° 1, p. 35-44.
- Bureerat, Sujin and Krit Sriworamas. 2007. "Population-Based Incremental Learning for Multiobjective Optimization". *Soft Computing in Industrial Applications*, vol. 39, p. 223-232.
- Camara, Mario, Julio Ortega, and Francisco de Toro. March 2007. "Parallel Processing for Multi-objective Optimization in Dynamic Environments.". In *In Proc. of IEEE International Parallel and Distributed Processing Symposium*. (Long Beach, CA 2007), p. 1-8.

- Camara, Mario, Julio Ortega, and Francisco de Toro. June 2009. "Performance Measures for Dynamic Multi-Objective Optimization". In *10th International Work-Conference on Artificial Neural Networks (IWANN 2009)*. (Salamanca, Spain 2009), p. 760-767.
- Chen, Chien-Chang and Cheng-Shian Lin. June 2007. "A GA-Based Nearly Optimal Image Authentication Approach". *International Journal of Innovative Computing, Information and Control*, vol. 3, n° 3, p. 631-640.
- Chen, Wenxiang, Thomas Weise, Zhenyu Yang, and Ke Tang. September 11-15 2010. "Large-Scale Global Optimization Using Cooperative Coevolution with Variable Interaction Learning". In *Parallel Problem Solving from Nature, PPSN XI*. (Kraków, Poland 2010), p. 300-309. Springer Berlin Heidelberg.
- Collette, Yann and Patrick Siarry. 2008. "On the Sensitivity of Aggregative Multiobjective Optimization Methods". *Journal of Computing and Information Technology*, vol. 16, p. 1-13.
- Cox, Ingemar, Matt L. Miller, and Andrew L. Mckellips. July 1999. "Watermarking as communications with side information". *Proceedings of the IEEE*, vol. 87, n° 7, p. 1127-1141.
- Deb, Kalyanmoy, 2001. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons.
- Diaz, Diego Sal and Manuel Grana Romay. June 2005. "Introducing a Watermarking with a Multi-Objective Genetic Algorithm". In *Genetic and Evolutionary Computation Conference (GECCO 2005)*. (Washington D.C., US 2005), p. 2219-2220.
- Dong, Jing and Tieniu Tan. December 2008. "Effects of watermarking on iris recognition performance". In *10th International Conference on Control, Automation, Robotics and Vision (ICARCV 2008)*. (Hanoi, Vietnam 2008), p. 1156-1161.
- Farina, Marco, Kalyanmoy Deb, and Paolo Amato. October 2004. "Dynamic Multiobjective Optimization Problems: Test Cases, Approximations, and Applications". *IEEE Transactions on Evolutionary Computation*, vol. 8, n° 5, p. 425-442.
- Gao, Xiaochuan, Chun Qi, and Haitao Zhou. 2006. "An adaptive compressed-dct-domain watermarking". In *8th International Conference on Signal Processing (ICSP 2006)*. (Beijing, China 2006).
- García-Cano, Edgar, Bassem S. Rabil, and Robert Sabourin. September 2012. "A Parallel Watermarking Application on GPU". In *Congreso Internacional de Investigación en Nuevas Tecnologías Informáticas (CIINTI 2012)*. (Tunja, Columbia 2012), p. 37-49.
- Goh, Chi-Keong and Kay Chen Tan. June 2007. "An Investigation on Noisy Environments in Evolutionary Multiobjective Optimization.". *IEEE Transactions on Evolutionary Computation*, vol. 11, n° 3, p. 354-381.

- Greeff, Marde and Andries P. Engelbrecht, 2010. *Multi-Objective Swarm Intelligent Systems*, chapter Dynamic Multi-objective Optimization Using PSO, p. 105-123. Springer-Verlag.
- Haouzia, Adil and Rita Noumeir. 2008. "Methods for image authentication: a survey". *Mul-tiMed Tools Appl*, vol. 39, p. 1-46.
- Harik, Georges R., Fernando G. Lobo, and David E. Goldberg. November 1999. "The Compact Genetic Algorithm". *IEEE Transaction on Evolutionary Computation*, vol. 3, n° 4, p. 287-297.
- Hu, Xiaohui and R.C. Eberhart. May 2002. "Adaptive particle swarm optimization: detection and response to dynamic systems.". In *IEEE Congress on Evolutionary Computation (CEC 2002)*. (Hawaii, US 2002), p. 1666-1670.
- Huang, Hsiang-Cheh, Jeng-Shyang Pan, Yu-Hsiu Huang, Feng-Hsing Wang, and Kuang-Chih Huang. 2007. "Progressive Watermarking Techniques Using Genetic Algorithm". *Circuits Systems Signal Processing*, vol. 26, n° 5, p. 671-687.
- Huang, Wenjun, Xiamu Niu, Wei Song, and Meiyu Lu. November 2008. "Digital Watermark for 3D Face Model Authentication and EER Performance Analysis". In *Eighth International Conference on Intelligent Systems Design and Applications*. (Kaohsiung 2008), p. 550-554.
- Jain, Anil K. and Umut Uludag. 2003. "Hiding Biometric Data". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, n° 11, p. 1494-1498.
- Jain, Anil K., M Narasimha Murty, and Patrick J. Flynn. September 1999. "Data Clustering: A Review". *ACM Computing Surveys*, vol. 31, n° 3, p. 264-323.
- Ji, Rongrong, Hongxun Yao, Shaohui Liu, and Liang Wang. December 2006. "Genetic Algorithm Based Optimal Block Mapping Method for LSB Substitution". In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'06)*. (California, USA 2006), p. 215-218.
- Kamble, Sushila, Vikas Maheshkar, Suneeta Agarwal, and Vinay Srivastava. February 2010. "DCT based texture watermarking using GLCM". In *IEEE 2nd International Advance Computing Conference (IACC'10)*. (Patiala, India 2010), p. 185-189.
- Kasinski, Andrzej, Andrzej Florek, and Adam Schmidt. 2008. "The PUT Face Database". *Image Processing & Communications*, vol. 13, n° 3-4, p. 59-64.
- Kim, Seong-Whan and Shan Suthaharan. 2004. "An Entropy Masking Model for Multimedia Content Watermarking". In *Proceedings of the 37th Hawaii International Conference on System Sciences 2004*. p. 1-6.
- Kumsawat, Prayoth, Kitti Attakitmongcol, and Arthit Srikaew. December 2005. "A New Approach for Optimization in Image Watermarking by Using Genetic Algorithms". *IEEE Transactions on Signal Processing*, vol. 53, n° 12, p. 4707-4719.

- Lee, Zne-Jung, Shih-Wei Lin, Shun-Feng Su, and Chun-Yen Lin. 2008. "A hybrid watermarking technique applied to digital images". *Applied Soft Computing*, vol. 8, p. 798-808.
- Li, Xiaodong and Andries P. Engelbrecht. July 2007. "Particle Swarm Optimization Tutorial". In *Genetic and Evolutionary Computation Conference (GECCO 2007)*. (London, UK 2007).
- Li, Xiaodong and Xin Yao. April 2012. "Cooperatively Coevolving Particle Swarms for Large Scale Optimization". *IEEE Transactions On Evolutionary Computation*, vol. 16, n° 2, p. 210-224.
- Li, Xiaodong, Jurgen Branke, and Michael Kirley. September 2007. "On Performance Metrics and Particle Swarm Methods for Dynamic Multiobjective Optimization Problems". In *IEEE Congress on Evolutionary Computation (CEC 2007)*. (Singapore 2007), p. 576-583.
- Li, Xiaoxia and Jianjun Wang. 2007. "A steganographic method based upon JPEG and particle swarm optimization algorithm". *Information Sciences*, vol. 177, p. 3099-3109.
- Li, Yongzhong, Hua Zhu, Ruiqing Yu, Ge Yang, and Jing Xu. 2008. "An adaptive blind watermarking algorithm based on dct and modified watson's visual model". In *International Symposium on Electronic Commerce and Security 2008*. p. 904-907.
- Licks, Vinicius and Ramiro Jordan. July-September 2005. "Geometric Attacks on Image Watermarking Systems". *IEEE Multimedia, IEEE Computer Society*, p. 68-78.
- Low, Cheng-Yaw, Andrew Beng-Jin Teoh, and Connie Tee. 2009. "Fusion of LSB and DWT Biometric Watermarking Using Offline Handwritten Signature for Copyright Protection". *Advances in Biometrics - Lecture Notes in Computer Science*, vol. 5558, p. 786-795.
- Moulin, Pierre, M. Kvanck Mhca, and Gen iu alan Lin. September 2000. "An information theoretic model for image watermarking and data hiding". In *IEEE International Conference on Image Processing*. (Vancouver, Canada 2000), p. 667-670.
- Nickabadi, A., M. M. Ebadzadeh, and R. Safabakhsh. June 2008. "June 2008. "DNPSO: A Dynamic Niching Particle Swarm Optimizer for multi-modal optimization". In . p.". In *IEEE World Congress on Evolutionary Computational (CEC 2008)*. (Hong Kong 2008), p. 26-32.
- Nie, Feiping, Zinan Zeng, Ivor W. Tsang, Dong Xu, and Changshui Zhang. November 2011. "Spectral Embedded Clustering: A Framework for In-Sample and Out-of-Sample Spectral Clustering". *IEEE Trans. on Neural Networks (T-NN)*, vol. 22, n° 11, p. 1796-1808.
- Pelikan, Martin, David E. Goldberg, and Fernando G. Lobo. 2002. "A survey of optimization by building and using probabilistic models". *Computational Optimization and Applications*, vol. 21, p. 5-20.

- Pereira, Shelby, Sviatoslav Voloshynovskiy, Maribel Madueno, Stephane Marchand-Maillet, and Thierry Pun. 2001. “Second generation benchmarking and application oriented evaluation”. p. 340-353.
- Potter, Mitchell A. and Kenneth A. DeJong. 2000. “Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents”. *Evolutionary Computation*, vol. 8, n° 1, p. 1-29.
- Rabil, Bassem S., Robert Sabourin, and Eric Granger. October 2010. “Intelligent watermarking with Multi-Objective Population Based Incremental Learning”. In *IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2010)*. (Darmstadt, Germany 2010), p. 131-134.
- Rabil, Bassem S., Robert Sabourin, and Eric Granger. April 2011a. “Impact of Watermarking Attacks on Biometric Verification Systems in Intelligent Bio-Watermarking Systems”. In *IEEE Workshop on Computational Intelligence in Biometrics and Identity Management 2011 - IEEE Symposium Series On Computational Intelligence 2011*. (Paris, France 2011), p. 13-20.
- Rabil, Bassem S., Robert Sabourin, and Eric Granger. August 2011b. “Watermarking Stack of Grayscale Face Images as Dynamic Multi-Objective Optimization Problem”. In *International Conference on Mass Data Analysis of Images and Signals in Medicine, Biotechnology, Chemistry and Food Industry*. (New York, USA 2011), p. 63-77.
- Rabil, Bassem S., Robert Sabourin, and Eric Granger. 2013a. “Rapid blockwise multi-resolution clustering of facial images for intelligent watermarking”. *Machine Vision and Applications*, art. <http://dx.doi.org/10.1007/s00138-013-0493-1>.
- Rabil, Bassem S., Robert Sabourin, and Eric Granger. July 2013b. “Securing Mass Biometric Templates Using Blockwise Multi-Resolution Clustering Watermarking”. In *International Workshop on Intelligent Pattern Recognition and Applications, WIPRA 2013*. (New York, USA 2013), p. 89-98.
- Rabil, Bassem S., Safa Tliba, Eric Granger, and Robert Sabourin. December 2013c. “Securing High Resolution Grayscale Facial Captures using a Blockwise Coevolutionary GA”. *Expert Systems with Applications*, vol. 40, n° 17, p. 6693–6706.
- Rosin, Christopher D. and Richard K. Belew. March 1997. “New methods for competitive coevolution”. *Evolutionary Computation*, vol. 5, p. 1-29.
- Sabourin, Robert. 1997. “Off-Line Signature Verification: Recent Advances and Perspectives”. In *Brazilian Symposium on Document Image Analysis- BSDIA*. p. 84-98.
- Sal, D., Manuel Graña, and Alicia D’Anjou. August 2006. “A MOGA to place the Watermark in an Hyperspectral Image”. In *International Geoscience and Remote Sensing Symposium*. (Denver, USA 2006), p. 774-777.

- Sezgin, Mehmet and Bulent Sankur. 2003. "Survey over image thresholding techniques and quantitative performance evaluation". *Journal of Electronic Imaging*, vol. 13, n° 1, p. 146-165.
- Shan, Songqing and G. Gary Wang. 2010. "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions". *Structural and multidisciplinary optimization*, vol. 41, p. 219-241.
- Shieh, Chin-Shiuh, Hsiang-Cheh Huang, Feng-Hsing Wang, and Jeng-Shyang Pan. 2004. "Genetic watermarking based on transform-domain techniques". *Pattern Recognition*, vol. 37, p. 555-565.
- Shih, Frank Y. and Yi-Ta Wu. 2005. "Enhancement of image watermark retrieval based on genetic algorithms". *Visual Communication and Image Representation*, vol. 16, p. 115-133.
- Sofge, Donald, Kenneth De Jong, and Alan Schultz. May 2002. "A blended population approach to cooperative coevolution for decomposition of complex problems.". In *IEEE Congress on Evolutionary Computation (CEC '02)*. (Hawaii, US 2002), p. 413-418.
- Sorwar, Golam and Ajith Abraham. June 2004. "DCT Based texture classification using a soft computing approach". *Malaysian Journal of Computer Science*, vol. 17, n° 1, p. 13-23.
- Tang, Ke, Xiaodong Li, P. N. Suganthan, Zhenyu Yang, and Thomas Weise. 2009. *Benchmark functions for the cec'2010 special session and competition on large scale global optimization*. Technical report. Nature Inspired Computation Application Lab., Univ. Sci. Technol. China, Hefei, China.
- Veldhuizen, David A. Van and Gary B. Lamont. July 2000. "On Measuring Multiobjective Evolutionary Algorithm Performance.". In *IEEE Congress on Evolutionary Computation (CEC 2000)*. (San Diego, US 2000), p. 204-211.
- Vellasques, Eduardo, Eric Granger, and Robert Sabourin, January 2010. p. 1-40. Intelligent Watermarking Systems: a Survey, Handbook of Pattern Recognition and Computer Vision, ed. 4th.
- Vellasques, Eduardo, Robert Sabourin, and Eric Granger. December 2011. "A high throughput system for intelligent watermarking of bi-tonal images". *Applied Soft Computing*, vol. 11, n° 8, p. 5215-5229.
- Vellasques, Eduardo, Robert Sabourin, and Eric Granger. July 2012. "Gaussian Mixture Modeling for Dynamic Particle Swarm Optimization of Recurrent Problems". In *Genetic and Evolutionary Computation Conference (GECCO 2012)*. (Philadelphia, USA 2012), p. 73-80.
- Voloshynovskiy, Sviatolsav, Shelby Pereira, Thierry Pun, Joachim Eggers, and Jonathan K. Su. 2001. "Attacks on Digital Watermarks: Classification, Estimation-Based Attacks, and Benchmarks". *IEEE Communications Magazine*, p. 118-126.

- Voloshynovskiy, Sviatoslav, Alexander Herrigely, Nazanin Baumgaertner, and Thierry Punz. 1999. "A stochastic approach to content adaptive digital image watermarking". In *Proceedings of the 3rd International Workshop on Information Hiding 1999*. (Dresden, Germany 1999), p. 211-236.
- Wang, Ran-Zan, Chi-Fang Lin, and Ja-Chen Lin. 2001. "Image hiding by optimal LSB substitution and genetic algorithm". *Pattern Recognition*, vol. 34, p. 671-683.
- Wang, Zhou and Alan C. Bovik. March 2002. "A Universal image Quality Index". *IEEE Signal Processing Letters*, vol. 9, n° 3, p. 81-84.
- Wang, Zhou and Alan C. Bovik. 2004. "Image Quality Assessment: From error visibility to structural similarity". *IEEE Transactions on Image Processing*, vol. 13, n° 4, p. 1-14.
- Wang, Ziqiang, Xia Sun, and Dexian Zhang. 2007. "A Novel Watermarking Scheme Based on PSO Algorithm". *Bio-Inspired Computational Intelligence and Applications*, vol. 4688, p. 307-314.
- Wei, Zhicheng, Hao Li, Jufeng Dai, and Sashuang Wang. July 2006. "Image Watermarking based on Genetic Algorithms". In *IEEE International Conference on Multimedia and Expo*. (Toronto, Canada 2006), p. 1117-1120.
- Wu, Min. 2001. "Multimedia Data Hiding". PhD thesis, Princeton University.
- Wu, Yi-Ta and Frank Y. Shih. February 2006. "Genetic Algorithm Based Methodology for Breaking the Steganalytic Systems". *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 36, n° 1, p. 24-31.
- Yaghmaee, Farzin and Mansour Jamzad. 2010. "Estimating Watermarking Capacity in Gray Scale Images Based on Image Complexity". *EURASIP Journal on Advances in Signal Processing*, vol. 2010, p. 1-9.
- Yang, Yi, Dong Xu, Feiping Nie, Shuicheng Yan, and Yueting Zhuang. 2010. "Image Clustering using Local Discriminant Models and Global Integration". *IEEE Transactions on Image Processing (T-IP)*, vol. 10, p. 2761-2773.
- Yang, Zhenyu, Ke Tang, and Xin Yao. 2008. "Large scale evolutionary optimization using cooperative coevolution". *Information Sciences*, vol. 178, p. 2985-2999.
- Yixin, Yan, Cao Wei, and Li Shengming. May 2009. "Block-based adaptive image watermarking scheme using Just Noticeable Difference". In *IEEE International Workshop on Imaging Systems and Techniques (IST '09)*. (Shenzhen, China 2009), p. 377-380.
- Yu, Feng, Damalie Oyana, Wen-Chi Hou, and Michael Wainer. March 2010. "Approximate Clustering on Data Streams Using Discrete Cosine Transform". *Journal of Information Processing Systems*, vol. 6, n° 1, p. 67-78.

- Yuan, Hua and Xiao-Ping Zhang. 2006. "Multiscale Fragile Watermarking Based on the Gaussian Mixture Model". *IEEE Transactions on Image Processing*, vol. 15, n° 10, p. 3189-3200.
- Zeng, Sang-You, Guang Chen, Liang Zheng, Hui Shi, de Garis H., Lixin Ding, and Lishan Kang. July 2006. "A Dynamic Multi-Objective Evolutionary Algorithm Based on an Orthogonal Design." In *IEEE Congress on Evolutionary Computation (CEC 2006)*. (Vancouver, Canada 2006), p. 573-580.
- Zhang, Fan, Xinhong Zhang, and Hongbin Zhang. 2007. "Digital image watermarking capacity and detection error rate". *Pattern Recognition Letters*, vol. 28, n° 1, p. 1-10.
- Zheng, Bojin. August 2007. "A new Dynamic Multi-objective Optimization Evolutionary Algorithm". In *Third International Conference on Neural Networks*. (Haikou, China 2007), p. 565-570.
- Zhu, Shaomin and Jianming Liu. 2009. "A Novel Adaptive Watermarking Scheme Based on Human Visual System and Particle Swarm Optimization". *Information Security Practice and Experience*, vol. 5451, p. 136-146.