ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MANUSCRIPT-BASED THESIS PRESENTED TO
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph.D.

BY
Eduardo VELLASQUES

INTELLIGENT WATERMARKING OF LONG STREAMS OF DOCUMENT IMAGES

MONTREAL, JANUARY 9, 2013

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Robert Sabourin, Thesis Director
Département de génie de la production automatisée à l'École de technologie supérieure

Mr. Éric Granger, Thesis co-Director
Département de génie de la production automatisée à l'École de technologie supérieure

Mr. Raynald Guilbault, Committee President
Département de génie mécanique à l'École de technologie supérieure

Mr. Marc Schoenauer, External Examiner
INRIA Saclay – Île-de-France

Mr. Tony Wong, Examiner
Département de génie de la production automatisée à l'École de technologie supérieure

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

ON DECEMBER 7, 2012

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

I am deeply thankful to my advisors, Dr. Robert Sabourin and Dr. Éric Granger. This work was only made possible by their invaluable guidance, encouragement and patience.

I would also like to express my sincere gratitude to the members of the examining committee: Dr. Raynald Guilbault, Dr. Tony Wong and Dr. Marc Schoenauer. Their comments helped me to improve the quality of this thesis.

Many thanks to my friends from LIVIA: Albert Ko, Ali Dewan, Bassem Guendy, Carlos Cadena, César Alba, Christophe Pagano, Clément Chion, Dominique Rivard, Éric Thibodeau, Eulanda dos Santos, Francis Quintal, George Eskander, Idrissa Coulibaly, Jonathan Bouchard, Luana Batista, Mathias Adankon, Marcelo Kapp, Miguel Gómora, Modhaffer Saidi, Paulo Cavalin, Paulo Radtke, Philippe Lamontagne, Rafael Menelau, Riadh Ksantini, Vincent Doré and Wael Khreich.

This thesis would not exist without the support of my family. Special thanks to Gislaine who is not only my wife but my very best friend. Her love, care and understanding have inspired me during both my masters and my doctoral studies. To her I dedicate this thesis. From my parents Antonio Carlos and Suely I learned the most important lesson that a child could learn: think big but be humble. My parents-in-law Valdenir and Ilda have also provided me invaluable words of wisdom and prayers during all these years. I also thank my siblings Luiz Fernando, Carolina and my sister-in-law Josianne for encouraging me. I could not forget our dear friends Rodrigo França, his wife Rachel França, Samuel Teixeira and his wife Simony Teixeira for being our family here in Canada.

I appreciate the financial support provided by Banctec Canada and Natural Sciences and Engineering Research Council (NSERC) of Canada. Special thanks to Banctec Canada V.P., Simon Fisher who believed in this project since before my arrival in Montreal.

Finally, I would like to thank God, the author of this work called Eduardo Vellasques. *Soli Deo gloria.*

INTELLIGENT WATERMARKING OF LONG STREAMS OF DOCUMENT IMAGES

Eduardo VELLASQUES

ABSTRACT

Digital watermarking has numerous applications in the imaging domain, including (but not limited to) fingerprinting, authentication, tampering detection. Because of the trade-off between watermark robustness and image quality, the heuristic parameters associated with digital watermarking systems need to be optimized. A common strategy to tackle this optimization problem formulation of digital watermarking, known as intelligent watermarking (IW), is to employ evolutionary computing (EC) to optimize these parameters for each image, with a computational cost that is infeasible for practical applications. However, in industrial applications involving streams of document images, one can expect instances of problems to reappear over time. Therefore, computational cost can be saved by preserving the knowledge of previous optimization problems in a separate archive (memory) and employing that memory to speedup or even replace optimization for future similar problems.

That is the basic principle behind the research presented in this thesis. Although similarity in the image space can lead to similarity in the problem space, there is no guarantee of that and for this reason, knowledge about the image space should not be employed whatsoever. Therefore, in this research, strategies to appropriately represent, compare, store and sample from problem instances are investigated. The objective behind these strategies is to allow for a comprehensive representation of a stream of optimization problems in a way to avoid re-optimization whenever a previously seen problem provides solutions as good as those that would be obtained by re-optimization, but at a fraction of its cost. Another objective is to provide IW systems with a predictive capability which allows replacing costly fitness evaluations with cheaper regression models whenever re-optimization cannot be avoided.

To this end, IW of streams of document images is first formulated as the problem of optimizing a stream of recurring problems and a Dynamic Particle Swarm Optimization (DPSO) technique is proposed to tackle this problem. This technique is based on a two-tiered memory of static solutions. Memory solutions are re-evaluated for every new image and then, the re-evaluated fitness distribution is compared with stored fitness distribution as a mean of measuring the similarity between both problem instances (change detection). In simulations involving homogeneous streams of bi-tonal document images, the proposed approach resulted in a decrease of 95% in computational burden with little impact in watermarking performace. Optimization cost was severely decreased by replacing re-optimizations with recall to previously seen solutions.

After that, the problem of representing the stream of optimization problems in a compact manner is addressed. With that, new optimization concepts can be incorporated into previously learned concepts in an incremental fashion. The proposed strategy to tackle this problem is

based on Gaussian Mixture Models (GMM) representation, trained with parameter and fitness data of all intermediate (candidate) solutions of a given problem instance. GMM sampling replaces selection of individual memory solutions during change detection. Simulation results demonstrate that such memory of GMMs is more adaptive and can thus, better tackle the optimization of embedding parameters for heterogeneous streams of document images when compared to the approach based on memory of static solutions.

Finally, the knowledge provided by the memory of GMMs is employed as a manner of decreasing the computational cost of re-optimization. To this end, GMM is employed in regression mode during re-optimization, replacing part of the costly fitness evaluations in a strategy known as surrogate-based optimization. Optimization is split in two levels, where the first one relies primarily on regression while the second one relies primarily on exact fitness values and provide a safeguard to the whole system. Simulation results demonstrate that the use of surrogates allows for better adaptation in situations involving significant variations in problem representation as when the set of attacks employed in the fitness function changes.

In general lines, the intelligent watermarking system proposed in this thesis is well adapted for the optimization of streams of recurring optimization problems. The quality of the resulting solutions for both, homogeneous and heterogeneous image streams is comparable to that obtained through full optimization but for a fraction of its computational cost. More specifically, the number of fitness evaluations is 97% smaller than that of full optimization for homogeneous streams and 95% for highly heterogeneous streams of document images. The proposed method is general and can be easily adapted to other applications involving streams of recurring problems.

Keywords: Digital Watermarking, Binary Watermarking, Intelligent Watermarking, Evolutionary Computing, Particle Swarm Optimization, Dynamic Optimization, Surrogate-Based Optimization, Change Detection, Gaussian Mixture Models, Gaussian Mixture Regression

TATOUAGE INTELLIGENT DE QUANTITÉS MASSIVES DE DOCUMENTS NUMERISÉS

Eduardo VELLASQUES

RÉSUMÉ

Le tatouage des images de documents sert plusieurs applications telles que l'authentification et la détection de documents numériques falsifiés. L'insertion d'un marqueur nécessite l'ajustement de plusieurs paramètres qui sont dépendants du contenu de chaque image. Le choix de la meilleure solution résulte du compromis à faire entre la qualité de l'image tatouée et la robustesse aux attaques du processus de marquage. Les systèmes de tatouage basés sur les algorithmes d'optimisation évolutionnaires sont appelés systèmes de tatouage intelligents (IW – Intelligent Watermarking). Le principal désavantage de ces systèmes est le coût computationnel prohibitif pour une utilisation grande échelle. L'approche adoptée dans cette thèse est de considérer une quantité massive de documents numériques à tatouer comme un flux de problèmes d'optimisation récurrents à traiter. Le système proposé est basé sur le concept de mémoire, qui permet de conserver une représentation des problèmes d'optimisation qui ont déjà été résolus afin de trouver rapidement une bonne solution pour une nouvelle image à tatouer.

L'approche adoptée dans cette thèse consiste à formuler le processus de tatouage d'un flux d'images de documents comme une séquence de problèmes d'optimisation récurrents. L'objectif principal de cette thèse est de concevoir un système de tatouage intelligent, doté de la capacité d'apprendre incrémentalement dans le temps les caractéristiques des problèmes d'optimisation à traiter. L'utilisation de la connaissance acquise dans le temps permettra de choisir une solution satisfaisante sans recourir systématiquement au processus d'optimisation qui est très coûteux en temps de calcul.

Pour être en mesure d'atteindre cet objectif, nous avons étudié plusieurs types de représentation en mémoire afin de traiter efficacement une quantité importante de documents à tatouer, tout en minimisant le coût computationnel. Pour ce faire, les stratégies proposées permettent de regrouper les problèmes d'optimisation de même nature en classes dans l'espace des paramètres. Un mécanisme de détection de changement permet alors de trouver rapidement en mémoire une bonne solution sans recourir au processus coûteux de l'optimisation des paramètres effectué sans connaissance a priori du problème à résoudre.

Dans un premier temps, le processus de tatouage d'un flux de documents numériques a été formulé comme une séquence de problèmes d'optimisation récurrents. Une nouvelle méthode basée sur le DPSO (Dynamic Particle Swarm Optimization) permet de résoudre efficacement ce type de problèmes d'optimisation récurrents. La technique proposée repose sur une mémoire associative qui est composée de plusieurs classes de solutions obtenues sur les images déjà traitées. Un mécanisme de détection de changement basé sur la distribution des valeurs de fitness des solutions en mémoire permet de vérifier rapidement si le problème d'optimisation en cours a déjà été résolu. Dans l'affirmative, alors une bonne solution est trouvée avec une

fraction du coût computationnel requis pour une optimisation complète. Sinon, le processus d'optimisation est activé et la mémoire associative est mise à jour afin de tenir compte de cette nouvelle information. La performance du système proposé évaluée sur une base d'images binaires homogène permet de diminuer de 95% le coût computationnel tout en conservant la même qualité de solutions obtenues par le processus d'optimisation activé sur chaque image à tatouer.

Dans un deuxième temps, le problème du choix d'une représentation mémoire plus compact a été adressé. Cette fois les solutions individuelles sont remplacées par une représentation basée sur une modélisation de l'espace des paramètres. Les mixtures de gaussiennes (GMM – Gaussian Mixture Models) sont ajustées à partir de toutes les solutions évaluées par le PSO en cours d'évolution. Les GMMs sont très efficaces pour représenter les classes de problèmes d'optimisation de même nature. La mise en oeuvre des mécanismes de gestion de la mémoire à long terme est simple, efficace, et consomme très peu de mémoire. Cette fois le mécanisme de détection de changement repose sur un échantillonnage de l'espace des paramètres du problème en cours de traitement et sur une mesure de similarité entre la distribution des fitness mesurées et celles mémorisées par la modélisation GMM pour chaque classe. Les résultats de simulation montrent que cette approche est plus flexible que celle basée sur les solutions individuelles conservées en mémoire. De plus, la performance obtenue sur des images de documents hétérogènes est nettement améliorée comparée à la méthode basée sur les solutions individuelles.

Finalement, la version complète du système proposé intègre à la modélisation GMM un mécanisme de régression afin de diminuer le coût computationnel requis lorsque le système de tatouage intelligent ne trouve pas une solution adéquate en mémoire, et qu'il doit obligatoirement activer le processus d'optimisation PSO. La stratégie consiste à remplacer l'évaluation coûteuse de la fitness par une estimation basée sur la modélisation GMM. Ce type d'approche est appelée *surrogate-based optimization* dans la littérature. La méthode proposée repose sur deux niveaux de prédiction ce qui permet dans le pire des cas de trouver une bonne solution même si la modélisation du problème d'optimisation est imprécise. L'impact de ce mécanisme de prédiction sur le coût computationnel requis pour l'optimisation des paramètres de l'algorithme de tatouage est significatif et celui-ci permet une diminution globale du coût computationnel sur des images hétérogènes.

En résumé, le système de tatouage intelligent proposé dans cette thèse est bien adapté pour l'optimisation d'un flux de problèmes d'optimisation récurrents. La qualité des solutions obtenues sur des bases d'images de documents homogènes et hétérogènes est équivalente à l'optimisation systématique sur chaque image avec PSO. En plus, le coût computationnel est réduit en moyenne de 97% sur des images homogènes et de 95% sur des images fortement hétérogènes. La méthode proposée est générale et peut être adaptée facilement pour l'optimisation de problèmes d'optimisation récurrents.

Mots-clés: Tatouage numérique, images bitonales, tatouage intelligent, problèmes d'optimisation dynamiques, détection de changement, algorithmes évolutionnaires, essaims de particules, mélange de gaussiennes, régression

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AFPI	Average Fitness Evaluations per Image
BCR	Bit Correct Ratio
BER	Bit Error Rate
BPSM	Block Pixel Statistic Manipulation
CCITT	Comité consultatif international téléphonique et télégraphique
CER	Constant Embedding Rate
CVIU	Computer Vision and Image Understanding
CWA	Conventional Weighted Aggregation
DCT	Discrete Cosine Transform
DFE	Decrease in Fitness Evaluations
DFT	Discrete Fourier Transform
DOE	Design Of Experiments
DOP	Dynamic Optimization Problem
DPI	Dots per Inch
DPSO	Dynamic Particle Swarm Optimization
DRDM	Distance Reciprocal Distortion Measure
DS-DPSO	Dual Surrogate Dynamic Particle Swarm Optimization
DWT	Discrete Wavelet Transform
EA	Evolutionary Algorithms
EC	Evolutionary Computing

ECC	Error Correction Code
ECDF	Empirical Distribution Function
EDA	Estimation of Density Algorithms
EM	Expectation Maximization
FER	Fixed Embedding Rate
GA	Genetic Algorithms
GECCO	Genetic and Evolutionary Computation Conference
GMM	Gaussian Mixture Model
GMR	Gaussian Mixture Regression
GP	Genetic Programming
HVS	Human Visual System
IDCT	Inverse Discrete Cosine Transform
IW	Intelligent Watermarking
JPEG	Joint Photographic Experts Group
KL	Kullback-Leibler
KS	Kolmogorov-Smirnov
LPF	Low Pass Filter
LSB	Least Significant Bit
LTM	Long Term Memory
LUT	Look Up Table
MLP	Multilayer Perceptron

MML	Minimum Message Length
MOGA	Multi Objective Genetic Algorithm
MOO	Multi Objective Optimization
MOOP	Multi Objective Optimization Problem
MOPSO	Multi Objective Particle Swarm Optimization
MSE	Mean Squared Error
NC	Normal Correlation
NSGA	Non-Dominated Sorting Genetic Algorithm
PDE	Partial Differential Equation
PDF	Portable Document Format
PLR	Perceptual Lossless Ratio
PSNR	Peak Signal-to-Noise Ratio
PSO	Particle Swarm Optimization
ROC	Receiving Operating Characteristics
SCS	Scalar Costa Scheme
SNDM	Structural Neighbourhood Distortion Measure
SNR	Signal-to-Noise Ratio
SOOP	Single Objective Optimization Problem
S & P	Salt and Pepper attack
STM	Short Term Memory
SVM	Support Vector Machines

TITI	Text/Image/Text/Image
UQ	Uniform Quantization
UQI	Universal Quality Index
VER	Variable Embedding Rate
WNR	Watermark-to-Noise Ratio

LIST OF SYMBOLS

$\boldsymbol{a}_1$	Independent (design) variable
$\boldsymbol{a}_2$	Dependent variable
$AFPI$	Average number of fitness evaluations per image
B	Block size
B_B	Maximum attainable block size
BCR	Bit Correct Ratio
BCR^{-1}	Inverse of the Bit Correct Ratio
BCR_F	BCR of the fragile watermark
BCR_R	BCR of the robust watermark
c_1	Cognitive acceleration constant
c_2	Social acceleration constant
c_α	Coefficient for confidence level α (KS statistic)
$C2(\Theta, \Theta')$	Distance between mixtures Θ and Θ'
Co	Cover image
$\lvert\mathbf{Co}\rvert$	Number of pixels in **Co**
$\boldsymbol{Co_S}$	Cover image employed to create the STM probe
CO	List of cover images
$Count_i$	Number of successfull recalls for probe i
Cr	Robust watermarked image
Crf	Robust/fragile (multi-level) watermarked image

Crfa	Multi-level watermarked/attacked image (Chapter II)
Crf′	Multi-level watermarked/attacked image (Chapter III)
Cw	Watermarked image
d	Number of dimensions of a given vector
d^*	Tuning constant for merge technique
d_B	Bhattacharyya distance
$d_{i,j}()$	Distortion for pixel (i, j)
$D_k(\boldsymbol{X}_{C,t-1}\|\|\boldsymbol{X}_{C,t})$	Kullback-Leibler (KL) divergence between between cumulatives sets of particles at instants t and $t-1$
D_{PW}^N	Normalized mean of the pairwise distance among individuals in the population
D_{PWM}^N	Normalized mean of the pairwise distance among probes in the LTM
D	Reciprocal distance matrix
$\mathfrak{D}$	Optimization history (set of all particle positions and fitness values for new image)
D_α	Critical value for confidence level α
DFE	Decrease in the number of fitness evaluations
$DRDM$	Distance Reciprocal Distortion Measure
E	Cumulative probability
e	Euler's constant
$f(\mathfrak{M}_j, \mathbf{Co})$	Evaluate set of solutions in image $\mathbf{C}_o$
$f(\boldsymbol{x})$	Exact fitness value

$\boldsymbol{f}$	Set of fitness values
$f_P(\boldsymbol{x})$	Predicted fitness value
$\hat{f}(\boldsymbol{x})$	Regression fitness value
$F(\mathbf{x}_i)$	Weighted aggregation function
F_{Evals}	Cumulative number of fitness evaluations
$F_{Evals,F}$	Cumulative number of fitness evaluations for full optimization
$F_{Evals,M}$	Cumulative number of fitness evaluations for the memory based approach
$\boldsymbol{F}(\boldsymbol{X}, \boldsymbol{Co})$	Fitness values of set of solutions $\boldsymbol{X}$ on image $\boldsymbol{Co}$
H_I	Image height
I	Identity matrix
i^*	Index of LTM probe with smallest C2 distance from new probe
I_o	Number of iterations required in the optimization of **Co**
I	Identity matrix
j^*	Best fit mixture component
k	Neighbourhood size (PSO)
k_{max}	Maximum number of components with $\alpha_j > 0$
k_{nz}	Number of components with $\alpha_j > 0$
K	Number of non-uniform (not all black or all white) blocks of size 8×8 (Chapter II), number of mixtures (Chapters III and IV)
$KS(\mathbf{A}, \mathbf{B})$	Kolmogorov-Smirnov statistic between vectors **A** and **B**
L	Number of LTM solutions

$\mathfrak{L}(\Theta, X)$	Log-likelihood of X on mixture Θ
$L_{\mathfrak{M}}$	Maximum number of probes in LTM
$\mathfrak{M}$	Long Term Memory
$\|\mathfrak{M}\|$	Number of probes in the LTM
$\mathfrak{M}_S$	Short Term Memory
m	Bit to be embedded
$\|\mathbf{m}\|$	Message length
$\boldsymbol{m}_i(\boldsymbol{x})$	Regression function for i^{th} mixture component
$\mathbf{m}_F$	Fragile watermark
$\mathbf{m}_{FD}$	Fragile watermark detected from the multi-level watermarked image
m_n	Detected bit
$\mathbf{m}_R$	Robust watermark
$\mathbf{m}_{RAD}$	Robust watermark detected from $\mathbf{C}_{RFA}$
$\mathbf{m}_{RD}$	Robust watermark detected from the multi-level watermarked/attacked image
n	Number of data points
N	Number of images (Chapter II), number of parameters (variables) in a given mixture (Chapter III), number of solutions (Chapter IV)
n_i	Number of points less than Y_i (KS statistic)
n_1	Number of elements in the first vector (KS statistic)
n_2	Number of elements in the second vector (KS statistic)
N_d	Number of data points needed to estimate a covariance matrix

N_g	Total number of generations of the level 3 optimization mechanism
N_{g^*}	Number of past solutions to employ in the evolution control
N_P	Number of black pixels (cover image)
N_R	Number of black pixels (watermarked image)
N_s	Number of sampled solutions
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Normal (Gaussian) probability density function of continuous random vector $\boldsymbol{x}$ given mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$NMDF$	Normalization (factor) with maximum diversity so far
$NMDF_{C2}$	Normalization (factor) with maximum diversity so far (applied to the $C2$ metric)
$\mathbf{p}_i$	Best position visited by particle i
$\mathbf{p}_g$	Best position visited by all neighbours of a given particle (global best solution)
$\boldsymbol{p}_{g,i}$	Global best solution stored in the i^{th} LTM probe
$\boldsymbol{p}_{g,S}$	Global best solution stored in the STM
$\boldsymbol{p}_{g,s1}$	Predicted global best for off-line surrogate
$\boldsymbol{p}_{g,s2}$	Predicted global best for on-line surrogate
$\boldsymbol{p}_{g^*,s1}$	Effective global best for all attempts of surrogate optimization
$\boldsymbol{P}_{j^*,g^*}$	Set of top N_{g^*} re-evaluated global best solutions
$p(\boldsymbol{a}_1, \boldsymbol{a}_2)$	Joint probability density function (pdf)
$p(x\|\Theta)$	Probability density function (pdf) of x on mixture Θ
$Q_\Delta\{\}$	Scalar uniform quantization

Q	Quantization step size
Q_R	Quantization step size (robust watermark)
Q_F	Quantization step size (fragile watermark)
r	Random number
r_1, r_2	Random numbers (PSO)
r_i	Reference point of the i^{th} objective i
$\boldsymbol{R}_s$	Vector with elements sampled from a normal distribution $N(0, \mathbf{I})$
s	New number of black pixels in watermarked image's block i
S	Shuffling seed.
$\mathfrak{S}_s$	Set of solutions obtained in the optimization of $\mathbf{C}_o[i]$
SNDM	Flippable pixels matrix
STM	Short Term Memory
T	Maximum size of $\boldsymbol{\delta}$
$\boldsymbol{U}$	Diagonal matrix with eigen-values of another matrix
$\mathbf{U}_B$	Upper boundary of the parameter space
$\mathbf{v}_i$	Velocity vector of particle i
$w_{ij}^{(t)}$	Posterior probability of the i^{th} datum on the j^{th} mixture element at instant t
w_a	Added watermark
w_n	Detected watermark
W	Flippability window size
W_I	Image width

$\mathbf{x}$	Point in the parameter space
$\mathbf{X}$	Search space
$\|X\|$	Number of particles in the swarm employed in the optimization process
$\boldsymbol{X}_{C,t}$	Set of all particles seen in all generations of all problem instances up to instant t
$\mathbf{x}_i$	Position vector of particle i
$\boldsymbol{X}_s$	Sampled solution
$\boldsymbol{X}_{S,S}$	Set of solutions sampled from the STM
α	Confidence level
$\alpha_{Crit,R}$	Confidence level (recall)
$\alpha_{Crit,C}$	Confidence level (evolution control)
α_s	Scaling factor of the quality measurement
α_i	Mixing weight of the i^{th} mixture component
α_M	Mixing weight of the merged mixture element
$\boldsymbol{\delta}$	Last T minimum $C2$ distance between a new probe and probes in the LTM
$\|\boldsymbol{\delta}\|$	Number of elements in $\boldsymbol{\delta}$
ΔQ	Difference between Q_R and Q_F
$\Delta Fitness$	Difference between fitness values
γ	Learning rate
γ_D	Learning rate decay
$\eta_{i,j}$	Second intermediate step to compute $C2$

θ	Recalled solution
θ_i	i^{th} mixture element
Θ_N	New mixture model
Θ_i	Mixture model of the i^{th} LTM probe
Θ_S	STM mixture model
$\mathbf{\Lambda}$	Set of eigen-vectors of a given matrix
$\mu_{\boldsymbol{\delta}}^t$	Mean of $\boldsymbol{\delta}$ at instant t
$\mu_{\boldsymbol{\delta}}^0$	Initial mean of $\boldsymbol{\delta}$
$\boldsymbol{\mu}_i$	Mean vector of the i^{th} mixture component
$\boldsymbol{\mu}_M$	Mean vector of the merged mixture component
ρ	Number of data points (KS statistic) in Chapter II, weighted normal (Gaussian) probability density function of continuous random vector $\boldsymbol{x}$ given mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$ in Chapter IV
σ_{ini}	Width of the initial covariance matrix
$\boldsymbol{\sigma}_i^2$	Width of regression kernel for the i^{th} mixture component
σ_δ^t	Standard deviation of $\boldsymbol{\delta}$ at instant t
σ_δ^0	Initial standard deviation of $\boldsymbol{\delta}$
$\mathbf{\Sigma}_i$	Covariance matrix of the i^{th} mixture component
$\mathbf{\Sigma}_M$	Covariance matrix of merged mixture component
Φ	First intermediate step to compute $C2$
χ	Constriction factor

ω_i	Weight of the i^{th} objective
$\varepsilon(\boldsymbol{x})$	Regression error

GENERAL INTRODUCTION

Numerous applications require the storage and transmission of document images. This leads to serious privacy concerns, specially considering the sensitive nature of the data stored in such images. Enforcing the security of document images is a paramount issue for many industries including financial, medical and legal. One easy strategy to enforce the security of document images is by means of cryptography. However, once an image has been decrypted, it can be easily manipulated and transmitted. Avoiding abuse (specially by insiders) requires a security mechanism that will "follow" the image wherever it goes and no matter what manipulation it suffers (as long as the manipulation does not affect its commercial value).

Digital watermarking allows the embedding of image-related data in a covert manner by manipulation of pixel values. This process is subject to a trade-off between robustness against image processing operations (attacks) and image quality. Since it is covert and involves manipulation of pixel values, a watermark provides means of enforcing the integrity and authenticity of a given image. The common approach is to employ a robust watermark (which can resist attacks) in order to enforce authenticity and a fragile watermark (which is easily destroyed by attacks) in order to detect tampering (enforce integrity).

Problem statement

The trade-off between quality and robustness can be adjusted by manipulation of heuristic parameters of the watermark embedder which means that digital watermarking can be formulated as an optimization problem. Different applications and images result in different trade-offs. Manual adjustment of such parameters is unfeasible in real world applications since it involves a lot of trial and error.

The common approach in the literature (Vellasques *et al.*, 2010a) is to employ evolutionary computing (EC) techniques such as Genetic Algorithms (GA) (Holland, 1992) and Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) in order to find the set of embedding parameters that results in an optimal trade-off between robustness and quality for each image

and/or application (set of attacks), an approach known as intelligent watermarking (IW). EC tackles optimization by evolving a population of candidate solutions during a certain number of generations. However, most IW approaches are limited to proof of concept scenarios (e.g.: less than 10 images) because of the high computational cost of EC. In practical applications, streams containing tens, hundreds or even thousands of document images are not uncommon.

One strategy to tackle the optimization of embedding parameters for such long streams of document images is to assume that a new case of optimization problem (associated with a new image) is somehow related to one or more previous cases of optimization and then, to employ knowledge of previous cases of optimization. In the EC literature such type of problem is known as a dynamic optimization problem (DOP). In a DOP the optimum (or optima for multi-modal problems) location changes with time. During a change, 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. There are two main scenarios for DOP: in the first one (periodical) the optimum suffers variations in fixed time intervals while in the second one (cyclical or recurrent) one or more fixed states (problem instances) occur repeatedly (Yang and Yao, 2008).

Since each image in a stream of document images corresponds to a single optimization problem, a stream of document images can be seen as a stream of optimization problems. In this research, it is hypothesized that because of similarities in image structure, some problem instances will re-appear over time which means that the optimization of embedding parameters for a stream of document images can be seen as a cyclic DOP. However a few remarks must be made. Firstly, it is reasonable to consider that two different images can share the same set of optimal embedding parameters. But it is extremely unlikely that two different images will result in the same combination of robustness and quality. This means that such cyclic DOP formulation involves similar rather than exact problem instances occurring repeatedly. Moreover, type I changes are also extremely unlikely. Therefore a new image might either correspond to a completely new problem instance (severe type III) or to a problem instance with the same optimum location as a previous instance but different fitness value (type II).

It can be said that there is an equivalence between optimal solutions obtained in cases of both, type II and non-severe type III changes (defined here as pseudo-type II). An optimal solution obtained for a given problem instance will still be optimal if that instance suffers a type II change. For a pseudo-type II change, other candidate solutions might provide a robustness/quality trade-off equivalent to what would be obtained through re-optimization without incurring in the heavy cost of EC. This means that for such cases, re-optimization can be avoided, leading to substantial decrease in the computational burden of EC.

This leads to three questions: How to preserve knowledge about previous problems? How to measure their similarity with new problem instances? How to update the knowledge of previous problems with knowledge obtained for new problem instances?

Such strategy of replacing costly re-optimization operations by ready-to-use solutions assumes a highly recurrent stream of optimization problems. However, as the amount of recurring problems decreases, tackling the cost of re-optimization operations becomes more important. This leads to the fourth question: How to employ previous knowledge in order to decrease the cost of re-optimization?

Objective and contributions

The main objective of this research is to decrease the computational cost of IW for streams of document images. In terms of volume, most real world applications rely on bi-tonal images. For this reason, the bi-tonal watermarking system of Wu and Liu (Wu and Liu, 2004) was employed as the baseline watermarking system in this research. The reason is that most bi-tonal watermarking systems found in the literature are specialized to certain applications while the system of Wu and Liu is considerably general and modular. The only limitation is that one of its modules (flippability analysis) is quite rigid to be employed in an optimization scenario and for this reason, the flippability analysis technique proposed by Muharemagic (Muharemagic, 2004) is employed in this research. Regarding EC technique, a diversity-preserving PSO is employed because of its fast convergence and ability to survey multiple optima (Kapp *et al.*, 2011). These two facts play an important role in preserving knowledge about a given optimization problem.

In a first moment IW is formulated as a DOP and the role of static solutions in preserving knowledge of previous cases of optimization problems for homogeneous streams of document images is investigated. Then, the use of density estimates of solutions found during optimization as a tool for preserving such knowledge for heterogeneous streams of document images is investigated. After that, a study is conducted on the use of previously learned density estimates as a mean of decreasing the cost of re-optimization in situations involving high variation between problem instances.

The main contribution of this research is the creation of a memory-based dynamic optimization technique that allows decreasing the cost of IW for streams of document images. The proposed approach has multiple levels with increasing computational cost. The architecture is organized into recall and optimization levels. A recall level comprises two main tasks: (1) comparing the similarity of new and previously seen problem instances, defined as change detection; (2) recalling ready-to-use solutions from the memory when the new problem is similar to a previously seen problem. An optimization level is only triggered if a similar problem case is not found in the memory and also comprises two main tasks: (1) performing optimization when a new problem is too different from previously seen problems; (2) building and updating a precise and compact representation of the stream of optimization problems up to that point. Knowledge about the stream of optimization problems is stored in two memory levels – Short Term Memory (STM) which contains knowledge about a single problem instance and Long Term Memory (LTM) which contains knowledge about multiple problem instances.

The first contribution is a technique that relies on a memory of static solutions as a mean of preserving knowledge about previous optimization problems. To this end, a novel strategy to employ memory solutions in order to perform change detection is proposed. This allows avoiding costly re-optimization operations for changes of type II (both real and pseudo). The focus here is on tackling optimization of embedding parameters for homogeneous streams of document images. However, an adaptive memory is essential for heterogeneous streams of document images, which leads to the second contribution.

The second contribution is a memory of density estimates of solutions found during optimization. Such memory provides a comprehensive model of a stream of optimization problems. A memory management mechanism which allows the knowledge of a stream of optimization problems to be accumulated in an incremental manner is proposed. Simulation results indicate that such memory is flexible enough to adapt to variations in heterogeneous streams of document images. Since re-optimization cannot be completely avoided, decreasing the cost of re-optimization is something crucial for industrial applications of IW, which leads to the third contribution.

Finally, in the third contribution of this thesis, the density estimates are employed in regression mode as a mean of replacing costly fitness evaluations during re-optimization, in a strategy known as surrogate-based optimization (Queipo *et al.*, 2005). This allows seeing optimization as a machine learning problem: surrogates are trained in a controlled environment and assigned to similar problems. It has been demonstrated empirically that such strategy is preferred in situations involving high variability in the problem stream (e.g. changing the sets of attacks) as surrogates allow decreasing the computational cost of re-optimization.

Organization of this Thesis

This manuscript-based thesis is organized into four chapters. In Chapter I a literature review on IW is presented. Proof-of-concept simulation results are provided in order to demonstrate the main advantages and limitations of IW. The content of this chapter was published as a book chapter in the Handbook of Pattern Recognition and Computer Vision, 4^{th} edition (Vellasques *et al.*, 2010a).

In Chapter II a memory-based Dynamic Particle Swarm Optimization (DPSO) technique is proposed. This approach relies on a memory of static solutions in order to decrease the computational burden of IW for homogeneous streams of document images by replacing costly re-optimization operations by memory recall. The performance of this approach is evaluated using streams of scientific journal pages. The content of this chapter was published at the 10^{th}

International Conference on Intelligent Information Hiding and Multimedia Signal Processing (Vellasques *et al.*, 2010b) and Applied Soft Computing (Vellasques *et al.*, 2011).

In Chapter III a memory of Gaussian Mixture Models (GMMs) is proposed, which is better suit to IW of heterogeneous streams of document images. To this end, specialized memory management operators were devised, which allow adapting the memory of GMMs to variations in the stream of optimization problems. It was demonstrated that such adaptive memory improves the performance of a memory of static solutions in scenarios involving heterogeneous streams of document images. The content of this chapter was published at the Genetic and Evolutionary Computation Conference (GECCO) 2012 (Vellasques *et al.*, 2012b) and accepted for publication in Applied Soft Computing (Vellasques *et al.*, 2012a).

In Chapter IV a technique that employs GMMs in regression mode is proposed, in order to replace costly fitness evaluations during re-optimization. In the proposed technique two levels of surrogates with increasing computational cost and precision are employed, where the first level tries to solve the optimization problem at the least possible cost while the second one works in a best-case scenario, behaving as an "insurance policy" for the previous level. It was demonstrated that such approach allows a significant decrease in the cost of re-optimization. Tackling the cost of re-optimization is a concern in scenarios involving high variation in the streams of document images. The content of this chapter was submitted to Applied Soft Computing (Vellasques *et al.*, 2012c).

CHAPTER 1

INTELLIGENT WATERMARKING

In this chapter we introduce the main aspects of intelligent watermarking systems to the unfamiliarized reader. Intelligent watermarking concerns the use of computational intelligence techniques as a mean of improving the performance of digital watermarking systems. Digital watermarking systems have become increasingly popular, specially due to the challenges behind the protection of multimedia documents in the Internet age. A crucial aspect of digital watermarking is that in real world applications, the performance of an embedder varies accross different images. In specialized watermarking systems, such issue can be tackled operationally, by limitting the type of image that a system will handle, for example. However, in a less constrained scenario, making sure that the watermarking is appropriately tunned for a specific image is a key element in protecting that image. Manually adjusting the watermarking system for each image is extremely expensive. In such case, the most appropriate strategy is to rely on techniques that can adapt the watermaking process automatically to variations in the data. The content of this chapter was published as a book chapter in the Handbook of Pattern Recognition and Computer Vision, 4^{th} edition (Vellasques *et al.*, 2010a).

1.1 Introduction

Managing digital versions of documents like bank cheques, invoices and printed forms has a significant role in modern economy. BancTec Inc[1] claims that its customers process 50 million documents a day, in 50 countries across the globe. The most common process involves transforming a continuous physical document into digitized image using an acquisition equipment (like a scanner), so they can be latter processed accordingly. These images are known as document images. Each specific application poses different requirements on the quality of these images. Some applications require high-definition, color images (e.g. over 16 million colors). In others, when storage and computational resources are limited, grey-level (e.g. 256 tones of grey) images are adequate. In many applications, black-and-white (bi-tonal) images

[1] http://www.banctec.com/

are adequate, which allow saving even more storing space and computational effort. This type of image is known as bi-tonal image. Bi-tonal images account for a significant share in the document management industry. According to BancTec, 95% of the 50 million document images processed by its customers daily are bi-tonal.

Enforcing the (1) integrity (has tampering occurred) and (2) authenticity (who is the author) of document images is considered a strategic issue by the financial industry, policy-makers and high-tech industry. A technique named digital watermarking allows enforcing these aspects, it comprises the covert embedding of information in an image through modifications on its pixel values.

The applicability of digital watermarking for the enforcement of aspects (1) and (2) has been shown in the literature (Cox *et al.*, 2002; Petitcolas *et al.*, 1999; Chen *et al.*, 2001). The most common approach is to use a watermark embedder to add side information in a subtle way so it can be latter read with the use of a detector. The integrity is usually achieved through robust watermarking – a watermark that can be still detected after the image has been modified (assuming that the modification has not affected the commercial value of the image) – while the authenticity is achieved through fragile watermarking – a watermark that is destroyed in the case of tampering.

In a digital watermarking system, substantial efforts are required to adjust system parameters to obtain an optimum trade-off between the robustness against attacks and the noise introduced by the watermarking process. Usually, an increase in robustness leads to an increase in the noise rate. Optimizing such parameters is not a trivial task. The most common strategy is to perform this optimization on each image with the use of evolutionary techniques such as Genetic Algorithms (GA) (Holland, 1992) and Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995).

Another issue with digital watermarking refers to making the processes involved in the embedding and detection of a watermark more adaptable to variations across different images.

This leads to another strategy which involves the application of suppervised learning in order to model these processes through regression.

Although many of the general concepts apply to different media like audio, video or images, there are many issues specific to each type of media. In this chapter, the state of the art in the use of computational intelligence in the watermarking of images is reviewed.

A method to optimize a system for bi-tonal image watermarking with the use of evolutionary computing is proposed as a case study. An adaptive watermarking system based on PSO is proposed for tuning the parameters used for watermarking of bi-tonal images. This baseline system embeds two watermarks – a robust one to enforce the integrity and a fragile one to enforce the authenticity. Bi-tonal images have some particularities. Since its pixels can only assume two values – black or white – the embedding must be carefully performed in order to preserve the imperceptibility of the watermark. This poses some constraints to the embedding capacity, since it is usually based on a trade-off between perceptibility and robustness against noise.

This chapter is divided into five sections. In Section 1.2 the main techniques and challenges of digital watermarking are presented. Section 1.3 covers the main aspects regarding the use of computational intelligence to devise adaptive digital watermarking systems. In Section 1.4, these concepts are illustrated in the optimization of a bi-tonal watermarking system (Muharemagic, 2004) with the use of PSO (Kennedy and Eberhart, 1995). Finally, Section 1.5 concludes this chapter.

1.2 Digital Watermarking

A watermark is an imperceptible (or minimally perceptible) mark, embedded into an image through modifications on pixel intensity values. There are numerous applications for digital watermarking such as broadcast monitoring, owner identification, proof of ownership, transaction tracking, authentication, copy control and device control (Cox *et al.*, 2002). It has two main objectives. The first is to ensure authenticity, and for this reason it must be robust to attempts of reproducing, removing or replacing. The second is to ensure integrity – any change

to watermarked image should create modifications also in the watermark so tampering could be latter detected.

There are two alternatives for adding a watermark to a digital image. The first is through the use of visible (but translucid) marks. Since visible watermarks affect the commercial value of an image, this option will not be considered in the scope of this research. The alternative, consists of adding side information, in an imperceptible manner, usually with some a perceptual model. The imperceptible mark can be added either with or without the partition of the host image into blocks, to allow the embedding of more than one bit. Regarding the domain, the encoding can be performed either by directly changing pixel values (spatial domain) or by mapping the image to a different domain (e.g. wavelet) and then changing the coefficients of this domain.

The typical structure of a digital watermarking system can be seen in Figure 1.1. The main components of such system are the embedder and the detector. Since in digital watermarking, a message is embedded into a media (image) and then recovered from that same image with the use of a detector, the most common approach is to model watermarking as a form of communication system (Cox *et al.*, 2002) (as depicted in Figure 1.1). In this figure, a message (m) is encoded into an appropriate signal, which is the watermark (w_a). The watermark is then embedded into a host or cover image (c_o), resulting in a marked image (c_w). The marked image is then compressed and/or processed and/or attacked. Then, a detector extracts the watermark from the watermarked/attacked image c_{wn} (here m_n, which might have been influenced by the compression/processing/attack) and uses it accordingly, for copyright control, tampering detection, etc. Data can be detected in two possible ways – with or without the use of cover image. The first is called **informed detection** while the second is called **blind detection**.
The fundamental problem of digital watermarking is to embed a certain amount of data into a cover image in accordance with two conflicting objectives – watermark robustness and image quality. That is, it is possible to make the embedded watermark more robust against certain types of attacks by increasing the power of the watermark signal. But this usually requires introducing more visual artifacts to the watermarked work.

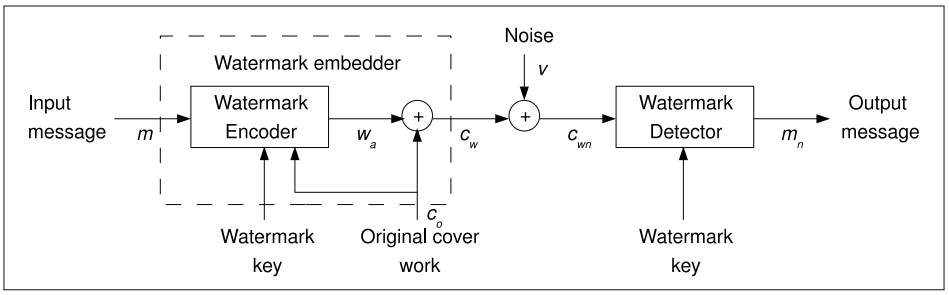


Figure 1.1 Communication model of digital watermarking (Cox *et al.*, 2002).

Through the rest of this section, a survey of watermarking techniques is presented. Since watermarking is limited by perceptual and robustness constraints, a review of the metrics and techniques employed in the evaluation of visual (perceptual) impact and robustness of the watermarking process will also be presented. Finally, this section will be closed with a discussion about the main challenges concerning digital watermarking of images.

1.2.1 Survey of Watermarking Techniques

Although digital watermarking research is still new, many efforts have been devoted to structure its fundamentals. Some of the concepts involved came from other areas of research as communications theory. Cox *et al* (Cox *et al.*, 2002) describe the main properties of a watermarking system.

A common approach is to model watermaking as a communication problem. Here, the watermark is treated as a message and the cover media is treated as communication channel. Through this approach it is possible to add layers to cover aspects like (Wu and Liu, 2003; Muharemagic, 2004):

- Security;
- How to embed and detect one bit;
- How to embed more than one bit using multiplexing/modulation;
- How to deal with parts of the host data that cannot embed data;

- How to detect which part of the data should be changed in order to ensure imperceptibility;
- What data to embed;
- What processing/embedding domain to use;

1.2.1.1 Embedding effectiveness

The embedding effectiveness of a watermarking system is related with the capacity of successfully adding a watermark into a cover image. That is, it is the probability that the output of the embedder will be watermarked (Cox *et al.*, 2002).

1.2.1.2 Fidelity

The fidelity of a watermarking systems is related with the similarity between the watermarked and the original image. Usually it comes at a price. A trade-off between fidelity and another property like embedding effectiveness or embedding rate must be considered when designing a watermarking system.

1.2.1.3 Embedding rate

Different images can present different embedding capacity (or payload). Some images contain smooth areas which make the embedding of data more difficult. With this in mind, there are two possible options in defining the payload of the watermarking system. One is to fix the embedding rate as low as possible, to deal with the cases where the image contains huge smooth areas. This approach is called Fixed Embedding Rate (FER). The other approach is to change the embedding rate accordingly and is called Variable Embedding Rate (VER). The problem with this approach is that control (side) information must be included, and it reduces the capacity of encoding watermark data.

1.2.1.4 Blind or informed embedding

During the embedding process, information about the cover image can be used, in order to improve system performance (imperceptibility, make watermark stronger to noise, etc). This approach is named informed embedding. For some other applications, there is no such huge demand on performance and for this reason, the embedding can be done without the use of cover image information. This type of embedding is called blind embedding.

1.2.1.5 Informed coding

During message coding, a source message, which is usually related with an specific watermarking application, is mapped into a message mark. This message mark is later embedded into the cover work through an addition or multiplication operation. Since it has been demonstrated in the literature that the embedding performance for a given cover work may vary for different messages, a very useful strategy is to use a mesage coding which uses information about the cover work and performs a one-to-many message-to-watermark mapping in order to improve the trade-off between the imperceptibility and robustness.

1.2.1.6 Reliability

The reliability of a watermarking system relates with the capacity of detecting an embedded watermark. A very useful tool to assess it is the Receiving Operating Characteristics (ROC) curve analysis. A ROC curve presents the False Positive versus False Negative results for a sequence of experiments. The analysis of such curves allows understanding the effect of a given parameter (e.g. capacity) in detection performance.

1.2.1.7 Robustness

Robustness refers to the ability to detect the watermark after common signal processing operations (Cox *et al.*, 2002). It is assessed empirically, by evaluating the watermark detection probability after the application of distortion. The use of benchmarking tools for evaluating

robustness is widely accepted by the digital watermarking community. There are many benchmarking tools available such as Stirmark[2], Checkmark[3], Optimark[4] and Certimark[5].

1.2.1.8 Bi-tonal images

The watermarking of bi-tonal images is a particular class of watermarking problem. The main issue concerning such type of watermarking regards the range of values a pixel can assume. In a grey-scale image, a pixel can usually assume an integer value between 0 and 255. In a bi-tonal image instead, a pixel can assume only two values: 0 or 1. For this reason, modifications in pixel values in a bi-tonal image are likely to be more perceptible for a human viewer than modifications in pixel values in grey-scale or colour images. Numerous works have been devoted to this particular type of watermarking (Pan *et al.*, 2000; Tseng and Pan, 2001; Awan *et al.*, 2006; Zhao and Koch, 1995; Mei *et al.*, Jan. 2001; Ho *et al.*, 2004a; Yang and Kot, Dec. 2006; Zhang and Qiu, 2005). Chen *et al* (Chen *et al.*, 2001) provide a survey of such type of technique. Most of these methods are either limited to a certain class of application like printed text or to a certain class of watermarks (robust or fragile). Wu and Liu (Wu and Liu, 2004) proposed a general block-based method which allows embedding more than one watermark in the same image at the same time, with different levels of robustness. This approach allows, for example, adding at the same time a robust watermark to enforce the authenticity of an image and a fragile watermark to enforce the integrity.

Despite the specific issues regarding the watermarking of bi-tonal images, it is also possible to convert the image to a grey-scale representation and perform the embedding with the use of more general techniques, followed by a post-binarization (Lu *et al.*, 2002). Furthermore, most digital watermarking systems share a common modular framework, both in terms of embedding and detection. Despite the particularities of bi-tonal watermarking, it is possible to consider a general framework for watermarking. In such framework, each individual module (or group of

[2]http://www.watermarkingworld.org
[3]http://watermarking.unige.ch/Checkmark
[4]http://poseidon.csd.auth.gr/optimark
[5]http://www.certimark.org

modules) can be replaced accordingly in order to improve the watermarking system or adapt it to new applications (e.g. watermarking of color images).

The rest of this subsection presents a general framework for watermarking and state-of-the art techniques for each of its modules.

1.2.1.9 Embedder

Although each application has its own specificity, the general structure of an embedder is depicted in Figure 1.2.

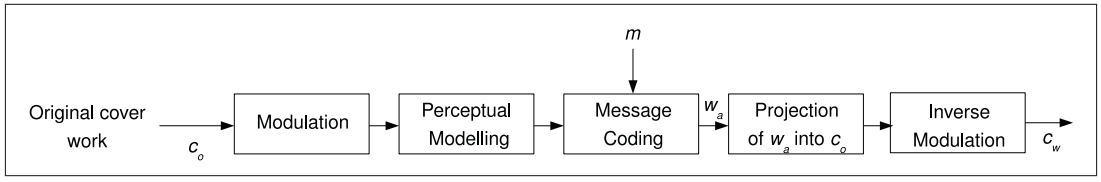


Figure 1.2 Structure of a watermark embedder.

1.2.1.9.1 Modulation

Modulation relates to choosing an appropriate representation for the image so information can be embedded into it. There are two main families of techniques – those that rely in the pixel representation of the image, namely spatial domain modulation and those that rely on a frequency representation of the image, namely frequency (or transformed) domain techniques. In the spatial domain techniques the pixel values are changed in order to embed one (Cox *et al.*, 2002) or many (Wu and Liu, 2004; Muharemagic, 2004) bits.

In the transformed domain techniques, the image is converted from its spatial representation to a frequency representation, using techniques such as Discrete Cosine Transform (DCT) (Cox *et al.*, 1996), Discrete Wavelet Transform (DWT) (Rezazadeh and Yazdi, 16-20 2006) and Discrete Fourier Transform (DFT) (ÓRuanaidh and Pun, 1998). These techniques apply better to grey-scale (Cox *et al.*, 1996; Wu *et al.*, 2003) and color (Zhao and Koch, 1995) images, since in the case of bi-tonal images, the post-binarization of the watermarked image can lead to loss of the embedded information. However, through appropriate choice of frequency spectrum

and binarization algorithm, this technique can be successfully applied to the watermarking of bi-tonal images (Lu *et al.*, 2002).

It is a common practice in bi-tonal watermarking to shuffle image pixels (with the use of a shuffling key) to distribute the flippable pixels through the image (Wu and Liu, 2004; Muharemagic, 2004). Figure 1.3 from Wu and Liu (Wu and Liu, 2004) gives an example of the effect of shuffling pixel positions in the distribution of flippable pixels.

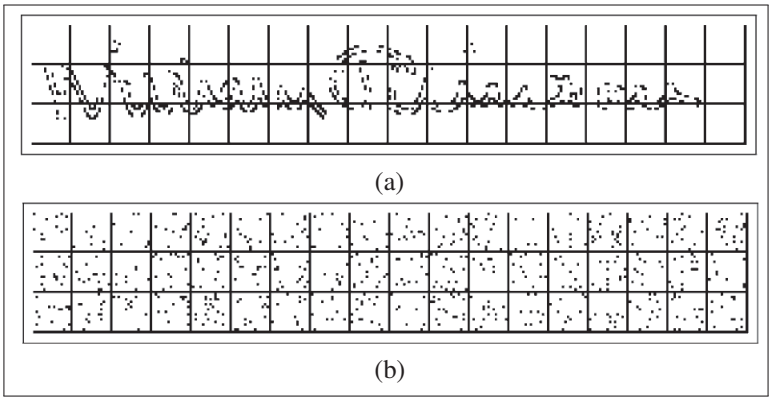


Figure 1.3 Effect of shuffling in the distribution of highly flippable pixels (Wu and Liu, 2004). (a) Highly flippable pixels before shuffling. (b) Highly flippable pixels after shuffling.

1.2.1.9.2 Perceptual modeling

Since the visual impact caused by the embedding process is one of the main constraints in most digital watermarking systems, an appropriate choice of frequency band (Cox *et al.*, 1996) or flippable pixels (Wu and Liu, 2004; Muharemagic, 2004; Zhang and Qiu, 2005; Ho *et al.*, 2004a) is crucial.

The Structural Neighbourhood Distortion Measure (SNDM) flippability metric proposed by Muharemagic (Muharemagic, 2004) illustrates the principle of perceptual modeling. This method uses a reciprocal distance matrix D_b in order to compute the flippability of a bi-tonal pixel, based on its $b \times b$ neighbourhood. A D_3 reciprocal distance matrix can be seen in Table 1.1.

Table 1.1 A 3×3 reciprocal distance matrix as seen in (Muharemagic, 2004). Each element corresponds to the distance from the central element.

0.7071	1.0	0.7071
1.0	0	1.0
0.7071	1.0	0.7071

The SNDM of a candidate pixel (cp) is computed as follows:

$$SNDM_{cp} = \frac{(cp \ \oplus \ N_b) \bullet D_b}{\mid D_b \mid} \tag{1.1}$$

where N_b represents the $b \times b$ neighbourhood of cp, D_b is $b \times b$ reciprocal distance matrix, $|D_b|$ is its number of pixels and $\oplus$ is the "exclusive or" (XOR) operator.

1.2.1.9.3 Message coding

Message coding is the process in which a message (m), which can be either a bit or a sequence of bits is transformed into a watermark that can be then, appropriately inserted into the modulated image. There are two main families of message coding techniques – direct message coding and multi-symbol message coding (Cox *et al.*, 2002). In direct message coding, a single bit is transformed into a message mark that is later embedded into the image through a sum or multiplication operation. Usually, a pre-defined reference mark with the same size of the cover image is required in both, embedding and detection.

In multi-symbol message coding, more than one bit must be encoded. There are three different approaches for transforming a multi-bit sequence into a message mark. The first is to break the multi-bit problem in many one-bit problems and apply direct coding to each bit. This approach is known as Time/Space Division Multiplexing and is practical only for small problems, since each one of the possible representation of the bit sequence requires a separate reference mark, that is, 2^N reference marks are required to encode a sequence of N bits. The second approach is frequency division multiplexing. In this approach the frequency domain is partitioned into several disjoint bands and a reference mark for each bit is encoded and then embedded on each band. Spread spectrum techniques rely on this type of encoding (Cox *et al.*, 1996; Wu, 2001).

The third approach is named code division. In this approach, several uncorrelated reference marks are embedded in the same work.

1.2.1.9.4 Projection

Projection is the effective modification of image pixels (or frequency coefficients) required to insert the coded watermark into the cover image. Although the intrinsic mechanisms by which the pixels or the frequency coefficients are modified is related with each coding technique, there are two main strategies to do these modifications (Wu, 2001). In the first (Type-I), the watermark signal is injected directly into the host signal, by either a sum or multiplication of the host signal (which can be a grey-level pixel value, a DCT coefficient) with the watermark signal. In the second (Type-II), the watermark is embedded by manipulating a given relationship within the host signal (e.g. ratio of black/white pixels).

1.2.1.9.5 Inverse modulation

In this step, the modulation process applied in the beginning must be reversed. In some situations, like in the shuffling case, it might be desirable to keep the image transformed (shuffled), and then in the detection side, reverse the shuffling upon a successful watermarking detection, in order to enforce the confidentiality of the image.

1.2.1.10 Detector

The detector basically extracts the watermark by applying the reverse process used on embedding. Generally, a detector has the structure shown in Figure 1.4.

1.2.1.10.1 Modulation

The modulation process is the same that was applied on embedding. If because of optimization either more than one modulation technique and/or parameters (e.g. shuffling key) are employed on embedding, the chosen technique/parameter must be known on detection.

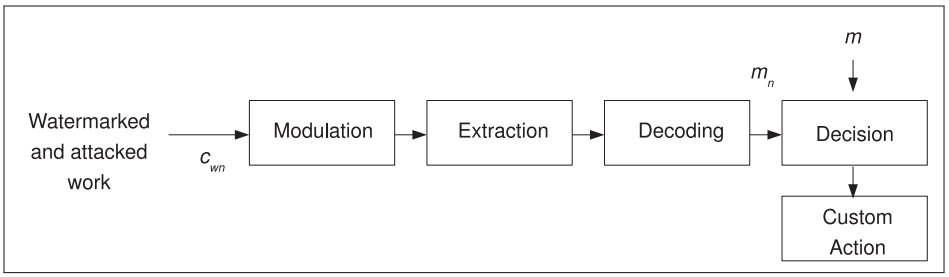


Figure 1.4 Structure of a watermark detector.

1.2.1.10.2 Extraction

The extraction process is in general, the inverse operation of projection. There are two main approaches – informed detection and blind detection. Informed detection requires a copy of the cover image. The difference between those images either in the spatial or frequency (Cox *et al.*, 1996) domain is employed in order to extract the watermark signal. In the blind detection otherwise, the original image is not required.

1.2.1.10.3 Decoding

Decoding can be seen as the inverse process of encoding. Here the extracted watermark signal is transformed into one or more bits.

1.2.1.10.4 Decision

On this step, the extracted mark is compared against a reference mark and then a decision is made. There are two possible outcomes of this decision – watermark is valid or watermark is invalid.

1.2.1.10.5 Custom action

During custom action, the extracted watermark and the decision are used to perform an application related task like preventing and image of being copied, reporting that the image is tampered or does not come from a certified sender.

1.2.2 Evaluation of visual impact

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. For example, it is a known issue that when the watermarking is performed in the frequency domain of an image (like the Discrete Cosine Transform), the modifications in lower frequencies are less perceptible by human viewers. This makes possible producing images with same fidelity but different quality.

Fidelity is computed using distortion metrics. Bellow, the most common distortion metrics are presented, where C_w is the watermarked image, C_o is the original image, $C_o[i]$ and $C_w[i]$ are the i^{th} pixels of C_o and C_w, respectively and $|C_w|$ is the number of pixels in C_w

a. Mean Squared Error (MSE):

$$MSE(C_w, C_o) = \frac{1}{|C_w|} \sum_{i=1}^{|C_w|} (C_w[i] - C_o[i])^2 \tag{1.2}$$

b. Signal-to-Noise Ratio (SNR):

$$SNR(C_w, C_o) = \frac{\sum_{i=1}^{|C_w|} C_o^2[i]}{\sum_{i=1}^{|C_w|} (C_w[i] - C_o[i])^2} \tag{1.3}$$

c. Peak Signal-to-Noise Ratio (PSNR):

$$PSNR(C_w, C_o) = max_{|C_w|}(\frac{\sum_{i=1}^{|C_w|} C_o^2[i]}{\sum_{i=1}^{|C_w|} (C_w[i] - C_o[i])^2}) \tag{1.4}$$

The quality of a watermarked image can be evaluated either by human observers (using standard test procedures, such as the *two alternatives, forced choice*), or by computational techniques that model the behaviour of the HVS.

For example, one of such techniques is the Distance Reciprocal Distortion Measure (DRDM) (Muharemagic, 2004). This metric has been specifically created to evaluate the difference between two bi-tonal images in terms of quality. Modifications in a bi-tonal image may affect the structure of elements within that image, affecting drastically the quality of the image. For this reason, care must be taken in order to avoid such modifications. The DRDM is based on the assumption that modifications in pixels close to viewer's focus are more noticeable. Also, due to particularities of human visual system, modifications in diagonal neighbours of a pixel are less noticeable than modifications on its immediate vertical and horizontal neighbours (4-neighbourhood).

A normalized weight matrix W_d, with size $d \times d$ is used to compute the distortion between two bi-tonal images. Each element of this matrix represents the reciprocal distance, relative to the center pixel. The distortion between two bi-tonal images is calculated as:

$$DRDM = \frac{\sum_{k=1}^{|C_w|} DRDM_k}{K} \tag{1.5}$$

where K is the number of non-uniform blocks (blocks that are neither all black nor all white) and $DRDM_k$ is a local distortion, calculated for each pixel, based on its $d \times d$ neighbourhood

$$DRDM_k = \sum_{d \times d} [|a_d - b_d| \times W_d] \tag{1.6}$$

1.2.3 Evaluation of robustness

As mentioned before, robustness refers to the ability to detect the watermark after the watermarked image has suffered common signal processing operations. These operations can be intentional or not. The intentional use of such type of operation in a watermarked image is called an attack. There are four main families of attacks: removal, geometric, cryptographic and protocol attacks (Voloshynovskiy *et al.*, 2001). In a removal attack, the embedded watermark is partially or completely removed either by a source of noise or with the use of image processing techniques such as de-noising, lossy compression, cropping, etc. In a geometric attack by its

way, the watermark is not removed but instead, the synchronization between the embedder and detector is affected with the use of affine transformations, such as rotation. In a cryptographic attack, the intention is to crack the security mechanisms employed on watermarking (such as the watermarking key). Finally, in a protocol attack, the objective is to threaten the validity of the system rather than its functionality. For example, in an protocol attack known as invertible watermark, an attacker extracts his own watermark from a watermarked image and claims he is the owner.

Intelligent watermarking usually aims at improving the robustness against removal attacks, since it is possible to increase the robustness against such attacks by adjusting embedding parameters (at the cost of adding more visible artifacts). Geometric attacks can be addressed either by detecting and inverting the distortion in the detector (Wu, 2001; Cox *et al.*, 2002) or by embedding the data in a domain resistant to affine transformations such as the Discrete Fourier Transform (DFT) (ÓRuanaidh and Pun, 1998). Cryptographic attacks can be made unfeasible by using large watermark keys. Finally, protocol attacks can be minimized by embedding signal-dependent watermarks, for example, a signature of the cover work(Yang and Kot, Dec. 2006).

Robustness against removal and geometric attacks is assessed empirically, by evaluating how does the watermark detector performs after the watermarked image has been attacked, that is, how similar are the embedded and detected watermarks. Fidelity metrics (such as the MSE) are employed to this end. Since an attack is only considered a concern when it does not affect the commercial value of the watermarked work, the embedded mark does not have to be resistant against attacks that affect the quality of the watermarked work. Usually, watermark-to-noise ratio (WNR), which gives the ratio between the power of the watermark signal and that of the noise introduced by attacks (Barni and Bartolini, 2004) is used in order to define the limit of the robustness

$$WNR(C_w, C_{wn}) = \frac{\sum_{i=1}^{|C_w|}(C_w[i] - C_o[i])^2}{\sum_{i=1}^{|C_w|}(C_{wn}[i] - C_w[i])^2} \tag{1.7}$$

where C_w is the watermarked image, C_o the original image, C_{wn} is C_w after processing/attack.

1.2.4 Challenges of watermarking

The use of digital watermarks makes possible the embedding of side information into a cover image in an imperceptible way. The embedding must be performed according to a trade-off between robustness and image quality. Watermarking can thus, be considered an optimization problem.

The main advantage of securing a document image with a digital watermark is that the protection provided is not ostensive. Depending on the perceptual model employed, the authenticity and integrity of a document are protected in an invisible manner. Despite these advantages, there are many known attacks to digital watermarking systems. For example, if a watermark detector is widely available, an attacker could use detection information to repeatedly make small changes to the watermarked work until the detector fails to detect the watermark (Muharemagic, 2004). Moreover, in a type of attack named ambiguity attack, someone can add a watermark to an already watermarked work in such a way that it would appear that this second watermark is the true watermark. In another type of attack named geometric attack, rotation, scale and translation transformations are applied to the watermarked image in a way that the synchronization between the embedded and detected watermark signal is lost, what could be a threat for an authenticity application.

The use of a robust watermark can mitigate the effects of most of these attacks (except for geometric attacks, which must be tackled with the use of registration marks (Cox *et al.*, 2002; Wu, 2001)), at the expense of adding more visual artifacts. This makes robust watermarks very attractive for authenticity applications. A fragile watermark can be very useful in the detection of intentional or unintentional modifications in the cover image (integrity enforcement). A watermark can be added in a fragile manner, and once its detection fails, it can be assumed that the image was tampered. The side effect of using fragile watermarks is that its detection will be affected by small variations in the image due to compression, processing or channel noise (here the cover image is considered a source of noise to the watermark signal). A balance between tampering protection and noise robustness must be considered.

Given these aspects, the combined use of fragile and robust watermarks may provide a very efficient way to protect both, the authenticity and the integrity of an image. However, the two main challenges in digital watermarking are (1) coping with variations across different types of images and (2) fine tune the embedding parameters to find an optimum balance between robustness and quality. As mentioned before, computational intelligence can be used in order to mitigate these problems.

1.3 Intelligent watermarking

In this section the main strategies concerning the use of computational intelligence in digital watermarking systems will be reviewed. The interference of channel and external noise in the message being transmitted is a known problem in information theory. There are several alternatives to tackle this problem. The most obvious is to increase the power of the message signal. However, in most channels, the power of the message signal is subject to constraints. This is specially the case in digital watermarking, where the power of the signal is subject to fidelity constraints. Another alternative is to spread the message signal through the host signal (spread spectrum) (Cox *et al.*, 1996). Since modifications in certain frequencies are less perceptible than in others, it is possible to increase the energy of the message signal in those frequencies without affecting the fidelity constraints. However, as demonstrated by Wu (Wu, 2001), although spread spectrum minimizes the influence of secondary sources of noise (attacks), it performs very poorly in what regards channel noise. Costa (Costa, 1983) demonstrated that if the properties of the host signal are known, it is possible to adapt the message coding to the host signal, minimizing the interference. These two examples show us that it is possible to explore properties of the cover work (side information) during embedding in order to make the watermarking process more adaptive to different cover works and types of attacks. In the literature, there are two main strategies to improve the adaptiveness of a watermarking system. The first is to use statistical or neural network classifiers for supervised learning of either a watermarking process, e.g. detection, or the evaluation of a given property of watermarking, e.g. imperceptibility. The second is to use evolutionary optimization in order to find a set of embedding parameters that result in near-optimal performance, according to one or more ob-

jectives such as robustness and fidelity. Both approaches are problem specific and thus, must be adapted to the specific watermarking systems. But the literature provides some guidelines for each of these approaches.

1.3.1 Supervised learning

In supervised learning, labelled samples collected for a problem are employed to estimate the parameters of a neural or statistical classifier. Assuming the samples have been assigned with two or more class labels, the trained model will provide a mapping of the samples into two or more regions corresponding to classes. This mapping is defined by the use of a linear or non-linear decision function. Once the parameters have been estimated with the use of training samples, it is possible to assign a class to unlabelled query samples in a task known as classification.

Classifiers such as the Multilayer Perceptron (MLP)(Bishop, 1996) and Support Vector Machines (SVM) (Vapnik, 1995) allow the estimation of very complex decision functions. These non-linear classifiers are very suitable to regression, as they can be considered universal function approximators. Thus, they can be applied in the task of learning a specific process of watermarking (e.g. detection) based on a set of labelled (training) data (e.g. a set containing cover works and their respective watermarked images). Moreover, they can be applied to the task of learning how to analyze a given property of a watermarked image (e.g. quality) based on labelled images.

The MLP is a very popular type of neural network classifier. It contains one input layer, one or more hidden layers and one output layer which produces either a label assignment or a function estimation. Each layer consists of one or more units, named neurons, which are connected to units in other layers by weights. Despite the simplicity of the heuristic employed, the ability of MLPs to learn any arbitrary decision function have been formally demonstrated (Duda *et al.*, 2000).

SVM is a large margin statistical classifier. It is based on the principle that given any two populations of labelled samples, the optimal decision function will maximize the distance between

the hyperplane separating the two sets of samples and the samples that are closest to this hyperplane. In SVM, a preprocessing phase projects the data to a higher dimensionality. Through preprocessing, non-linearly separable sets of samples become linearly separable. This allows the use of such type of classifier in the regression of non-linear functions.

As illustrated in Figures 1.2 and 1.4, a digital watermarking system is modular. The implication of this modularity is that each module usually handles a very specific aspect of the watermarking process. The modular nature of a watermarking system makes possible to isolate some of these modules and train non-linear classifiers to implement their functionality.

Although digital watermarking is based on a solid theoretical framework, one of its weaknesses is that the noise sources (both, host channel and external) are always assumed to have a given form (usually Gaussian). The alternative found was to use non-linear classifiers, to make some of the watermarking processes more adaptable to the real form of the noise sources. The use of a classifier in this process is straightforward. A classifier is trained with labelled data, where the raw data is usually the same data that the real module receives as an input while the label (or target data) is the output. There are two approaches in what regards target data. The first is to use the data provided by the module to be replaced (i.e. someone could pick the message coding module described in sub-section 1.2.1.9 and generate a set of target data for a given range of input) or use data provided by humans (e.g. a score for the perceptual modeling module).

1.3.1.1 MLP

The watermarking system proposed by Chang and Lin (Chang and Lin, 2004a) illustrates the first approach. In the baseline watermarking system, a watermark is embedded in the Discrete Wavelet Transform (DWT) coefficients of an image. The DWT decomposition breaks the image into a hierarchical set of coefficients, where each coefficient has four children (quad-tree). Each level of this quad-tree corresponds to a given level of resolution. A pseudo-random number sequence (based on a seed) is employed in the task of choosing the set of coefficients where the embedding will be performed. Given a coefficient s_k, the embedding is performed

by adding (to embed a '1') or subtracting (to embed a '0') a constant α to each of the four children of s_k. For example, given an hypothetical coefficient $s_k = C_3$ in Figure 1.5, four bits are embedded by adding or subtracting this constant to D_1, D_2, D_3 and D_4. In this Figure, each letter (A, B, C and D) corresponds to a given resolution while each index (1, 2, 3 and 4) corresponds to each of the subbands for that resolution.

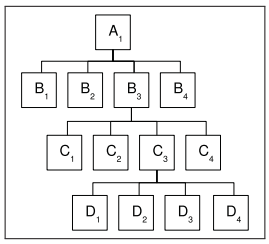


Figure 1.5 Example of a quad-tree structure.

However, after modifying these coefficients and inverting the DWT transform, a reference image is necessary in order to compute the difference between both coefficients and detect the embedded data. Instead of this, the authors use a MLP in order to map the relationship between the coefficients. Basically, this network contains eight input neurons – the parent of s_k (B_3), the three siblings of its parent (B_1, B_2 and B_4), the three siblings of s_k (C_1, C_2 and C_4), and s_k itself (C_3). The output neurons are the four children of s_k (D_1, D_2, D_3 and D_4). The MLP learns the mapping between a given coefficient and its children so the data can be detected latter without the use of the cover image.

During detection, the trained MLP is employed in order to recover the previous coefficient values (that is, their value before embedding) and the data is extracted from these coefficients by computing the difference between the output of the MLP and the children of each coefficient.

1.3.1.2 SVM

Chang and Lin used this principle in the task of creating a SVM-based perceptual modeling module (Chang and Lin, 2004b). In the baseline watermarking system technique, the embedding is performed by manipulating the pixel values in blocks of 3×3 pixels. However, the extent at which each pixel can be manipulated is limited by image quality constraints. To cope

with this, a SVM is used in the task of providing a score which will define the amount of modification each pixel in any given 3×3 window can suffer. In this technique, the four most significant bits of each one of the nine pixels in a given 3×3 are employed in the task of training a SVM classifier. The target data, which is a score, is manually provided by a human specialist.

Tahir *et al* (Tahir *et al.*, 2005) also employed a SVM on digital watermarking but in the decoding side. The basic principle is to use SVM in order to improve the detection performance under Gaussian attack. The baseline watermarking system embeds a message in an image by manipulating its DCT coefficients. If no attack has occurred, the embedded bits will form two distinct Gaussian distributions. However, after an attack, these two distributions will overlap. SVM can be employed in order to make these two overlapped distributions separable in a higher dimension, during detection. This was the approach employed by the authors. Basically, for each bit, 22 statistical features are computed and used as a feature vector. The bit value is used as target data. After the SVM has been trained, during detection, the same features are computed from each bit and fed into the trained classifier.

Davis and Najarian (Davis and Najarian, 2001) employed an MLP in order model the Human Visual System (HVS). In the proposed technique, each image is subdivided in blocks of 64×64 pixels. The image is transformed to a wavelet domain (DWT). These 4096 coefficients, along with a given watermark strength are used in order to train an MLP. The target data is a score provided by a human viewer. The trained MLP can be employed in the task of analyzing the visual impact of a given watermarking task.

1.3.2 Optimization of watermarking parameters

This family of techniques relies in the use of optimization in order to finetune the parameters of embedding algorithms, aiming thereby, increasing the robustness of the embedded watermark and decreasing the visual distortion caused by the embedding process.

Optimization can be categorized in three approaches. In the first approach, theoretical properties of the watermarking system are explored, using mathematical analysis (e.g. Cox *et al* (Cox

et al., 2002) uses mathematical analysis in order to adjust the embedding strength parameter so the distance between embedded "1" and "0" bits can be increased).

In the second, parameters are assumed to be independent and then, local optimization (greedy algorithm) is performed on each parameter. This was the strategy employed by Muharemagic (Muharemagic, 2004) in his adaptive system.

In the third approach, Evolutionary Computing (EC) techniques such as Genetic Algorithms (GA) (Holland, 1992) or Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) are used in order to adjust the embedding parameters according to constraints in the robustness of the watermark and/or the fidelity/quality of the watermarked image. This is the most common approach in the literature, mainly due to the simplicity of techniques and the ease in adapting them to many different types of watermarking systems. Moreover, EC, does not assume a distribution of the noise source or parameter space, as with mathematical analysis/-greedy search. Figure 1.6 illustrates the general structure of a system based on this approach (the watermark embedder corresponds to the embedding system depicted in Figure 1.2 while the detector corresponds to the detection system depicted in Figure 1.4).
Usually, one or more embedding parameters are encoded either as a chromosome (GA) or as a particle (PSO). The objective functions usually involve at least one fidelity/quality (e.g. PSNR, as seen in Equation 1.4) and one robustness (e.g. MSE, as seen in Equation 1.2, between embedded and detected watermarks) metrics. To evaluate robustness, one or more attacks are applied to watermarked image. Then, the detected watermark is compared with the embedded one with the use of fidelity metrics. The objective of the optimization algorithm is to minimize simultaneously, (1) the visual impact caused by the embedding procedure and (2) the difference between the embedded and detected watermarks under a given set of attacks.

Some authors however do not follow this multi-objective optimization approach and use only one objective function (either noise or robustness). In some methods also, although robustness is evaluated, no attack is applied to watermarked image (there is theoretical basis to assume that the cover image itself is a source of distortion to the embedded watermark (Cox *et al.*, 2002)).

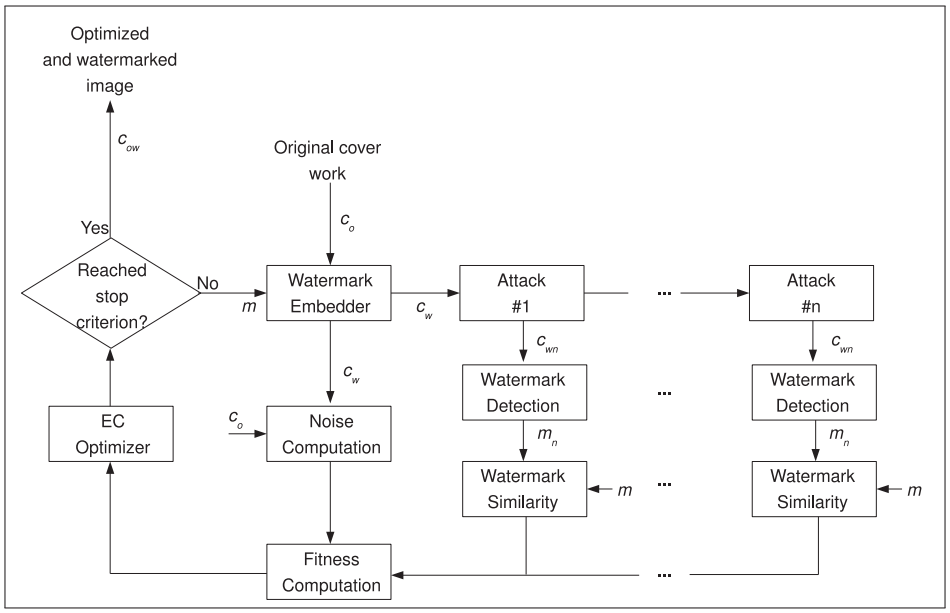


Figure 1.6 General structure of a system based on EC optimization strategy (based on the model proposed in (Shieh *et al.*, 2004)).

The robustness evaluation consists of a simple distance computation between the detected and embedded watermarks.

Shieh *et al* (Shieh *et al.*, 2004) optimize a DCT-based watermarking system with GA. A DCT block transformation is applied to a grey-scale image. After that, the embedding is performed through the manipulation of the polarity between the watermark and the DCT coefficients. The authors employ GA to find the DCT coefficients that result in the best combination of robustness and image quality. The robustness is computed by embedding a watermark into an image, applying one of three different attacks (Low Pass Filtering, Median Filtering and JPEG compression with quality factor 80%), detecting the watermark and computing its normal correlation (NC) against the original watermark. The quality is computed with the use of PSNR.

Lee *et al* (Lee *et al.*, 2007) employed a hybrid GA/PSO technique in the optimization of a DWT-based watermarking system. Heuristic weights are used to deal with the trade-off between robustness and fidelity in the modulation process. Authors proposed using GA and PSO in a paralel to optimize the heuristic weights. The authors applied various classes of attacks

only after the optimization procedure, to evaluate the effectiveness of the proposed method (Filtering, geometrical, JPEG compression and image enhancement).

Ji *et al* (Ji *et al.*, 2006) optimized a Least Significant Bit (LSB) substitution steganography method with the use of GA. Since the embedding procedure is based on the use of a mapping function, the authors employed GA in order to obtain a mapping function that provides robustness and fidelity, at the same time. Distortion metric is employed as fitness function.

Li and Wang (Li and Wang, 2007) employed PSO in the task of optimizing a DCT-based steganographic method. This method embeds a secret image into the least significant bits of the DCT coefficients of a cover image and relies in the use of a substitution matrix during the message encoding step of the embedding process. The authors employed PSO to find an optimal substitution matrix. The objective function is based on a distortion metric (PSNR).

Wei *et al* (Wei *et al.*, 2006) applied GA to the task of identifying the best coefficients in a spread spectrum DCT watermarking system. The combination of the similarity metrics between original and extracted mark is fed into the GA algorithm as a fitness function. Four different attacks – Low Pass Filtering, Scaling, Gaussian Noise, JPEG compression – are employed in this method.

Pan *et al* (Pan *et al.*, 2004) applied GA to the task of optimizing a Block Pixel Statistic Manipulation (BPSM) method. In this BPSM watermarking method, the mean of the grey-level values of the 8-neighbourhood surrounding pixels of a given central pixel is computed. Then, the embedding is performed by manipulating that value. The manipulated value is stored in the central pixel. Authors used GA to search for a near optimal set of pixels, in terms of robustness and fidelity. The Bit Correct Ratio (BCR) between original and extracted watermark (JPEG compression is applied in the watermarked image) as well as the PSNR of the watermarked image are employed as fitness functions.

Sal *et al* (Sal *et al.*, 2006) applied NSGA-II(Deb *et al.*, 2002), which is a Pareto-based Multi-Objective Genetic Algorithm (MOGA), in the task of optimizing a DCT-based watermarking system. The parameters being optimized are the DCT coefficients where embedding will be

performed. The distortion and robustness are measured directly on the DCT coefficients. The authors do not apply attacks during optimization process.

Chen and Lin (Chen and Lin, 2007) employed GA in the detection of nearly optimal embedding positions on DCT blocks. The fitness evaluation is based only in the MSE between original and watermarked images. No similarity between embedded and detected watermarks is employed. Also, the proposed method uses no attack during optimization procedure.

Areef and Heniedy (Areef *et al.*, 2005) apply GA in the optimization of a DWT-based watermarking method. Basically, the cover image is decomposed with the use of the Haar wavelet filter. Then, the watermark signal is embed into a given set of wavelet coefficients as a multiplicative watermark. GA is applied then, in order to identify a set of coefficients which maximizes the robustness against JPEG compression (BCR computation is performed on watermarked/JPEG compressed images for this purpose) and minimizes the embedding noise (measured with the use of MSE).

Shih and Wu (Shih and Wu, 2004) applied GA in order to create a rounding rule for DCT embedding. The basic problem is that on DCT embedding, integer pixel values are transformed into real-valued DCT coefficients. The watermark is embedded on these coefficients which are then transformed back to integer pixel values by an Inverse DCT (IDCT). During this process, information might be lost due to rounding error. The authors proposed the use of GA to tackle this problem. Basically, a gene is used for each DCT coefficient, where '1' means that the resulting value from the IDCT process must be truncated and added to 1 ($\phi_i^* = Trunc(\phi_i) + 1$) and '0' means that the resulting value must be just truncated ($\phi_i^* = Trunc(\phi_i)$), where ϕ_i is the DCT coefficient at location i. Two fitness functions – one based on the Normalized Correlation (NC) between embedded and detected watermark and another based on the PSNR between cover and watermarked images – are employed.

Kumsawat and Attakitmongcol (Kumsawat *et al.*, 2005) proposed the use of GA to optimize a Multilevel Wavelet Transform watermarking method. In the proposed method, GA is employed in order to identify the coefficients that improve the performance of the base method. Here,

the authors make use of the Universal Quality Index (UQI) to measure the similarity between watermarked and cover images. The robustness is evaluated with the use of correlation.

Diaz and Romay (Díaz and Romay, 2005) applied NSGA-II (Deb *et al.*, 2002) in the optimization of a DCT-based watermarking method. Normalized correlation is applied to measure the robustness of the proposed solution against JPEG compression and smoothing. MSE is applied to measure the noise between watermarked and cover images.

Khan and Mirza (Khan and Mirza, 2007) proposed the use of Genetic Programming to achieve an adaptive perceptual modeling algorithm for a DCT-based watermarking system. In the proposed method, genetic programming operators are employed in the task of creating a perceptual modeling function for a given embedding task. The Structure Similarity Index (a quality measure) and the Bit Correct Ratio (a robustness measure) are used as objective functions.

Wu and Shih (Wu and Shih, 2006) applied GA in order minimize the occurrence of statistical features that are used for steganalysis purposes. In the proposed method, the modifications to be done to a DCT block in order to embed a given message are coded as chromosomes. During optimization, a message is embedded into the DCT coefficients of a cover image. Then, Bit Error Rate (BER) is used to evaluate the difference between extracted and detected watermarks. Analysis functions based on the type of steganalysis attack the system must resist are used in order to evaluate the robustness against such attacks. These two metrics are employed as fitness functions in the GA optimizer.

1.3.3 Key Issues

Among all the existing families of optimization techniques, those based on EC have been successfully employed in many different scenarios involving the optimization under uncertainty (stochastic optimization). As mentioned before, the number of parameters to be adjusted in a digital watermarking system is indeed a concern. Adjusting these parameters according to an optimum tradeoff between robustness and quality can help to make watermarking more suitable to industrial applications. But it is difficult to know the exact form of the problem beforehand.

A strategy to tackle this problem is to consider watermarking as stochastic optimization problem and apply EC to such end (optimization of embedding parameters).

The main reason for the use of EC in the optimization of watermarking systems is that the objective functions are usually noisy (multi-modal). Since EC techniques are based on population of candidate solutions, it is less likely to the optimization algorithm to get stuck in a local optimum. Moreover, due to the modular nature of a watermarking system (with numerous different techniques for each module) the use of EC provides flexibility to the optimization process, since it does not require gradient information of the function under consideration (Parsopoulos and Vrahatis, 2002). There are many methods based on this strategy in the literature (Table 1.2). Actually, the majority of the intelligent watermarking methods are based on this strategy. Regarding the number of objective functions employed, there are two main optimization strategies – one consisting of the use of a single objective function (e.g. fidelity), known as Single Objective Optimization Problem (SOOP) and another one consisting of the combination of many objective functions, known as Multi Objective Optimization Problem (MOOP). With respect to the GA or PSO algorithms employed to deal with MOOP, there are two strategies. One which consists of aggregating many objective functions into one through weighted sum – and then use classical GA and PSO – and another which consists of handling many conflicting objectives during optimization – which is the case of Multi Objective GA (MOGA) and Multi Objective PSO (MOPSO).

The optimization of a watermarking system is a multi-objective problem, since it must handle at least two conflicting objectives – fidelity/quality and robustness. However, the vast majority of research has been directed towards the use of single-objective optimization algorithms. The most common approach to handle multi-objective optimization in a single-objective optimization algorithm is to combine all fitness functions into one with the use of weighted sum. However, such approach usually favours one objective in detriment of the others. Multi-objective optimization algorithms such as the NSGA-II (Deb *et al.*, 2002) and MOPSO (Coello *et al.*, 2004) rely on Pareto dominance and can be employed in order to mitigate the problem of favouring one objective.

Table 1.2 Summary of EC-based digital watermarking techniques.

Method	Watermarking Strategy	Optimization Method / Algorithm	Parameter	Distortion	Attack
Shieh *et al* (Shieh *et al.*, 2004)	DCT	MOOP/GA	Coefficients	PSNR	Low Pass Filtering Median Filtering JPEG Compression
Lee *et al* (Lee *et al.*, 2007)	DWT	SOOP/ Hybrid (GA/PSO)	Coefficients	Perceptual Lossless Ratio (PLR)	Median Filtering Wiener Filtering Average Filtering Gaussian Filtering Rescaling Rotation Cropping Jittering StirMark JPEG Compression Image enhancement (6 different algorithms)
Li and Wang (Li and Wang, 2007)	DCT	SOOP/PSO	Encoding (substitution matrix)	PSNR	None
Ji *et al* (Ji *et al.*, 2006)	Least Significant Bit	SOOP/GA	Substitution matrix	PSNR	None
Wei *et al* (Wei *et al.*, 2006)	DCT	MOOP/GA	DCT coefficients	None	Low Pass Filtering Scaling Noise JPEG Compression
Pan *et al* (Pan *et al.*, 2004)	Block Pixel Statistic Manipulation (BPSM)	MOOP/GA	Embedding blocks	PSNR	JPEG Compression
Sal *et al* (Sal *et al.*, 2006)	DCT/Hyperspectral images	MOOP/NSGA-II	DCT coefficients	Coefficient values	Low Pass Filtering
Wu and Shih (Wu and Shih, 2006)	DCT	MOOP/GA	Coefficient values	None	Steganalisys
Chen and Lin (Chen and Lin, 2007)	DCT	SOOP/GA	DCT coefficients	MSE	None
Areef and Heniedy (Areef *et al.*, 2005)	DWT	MOOP/GA	Frequency bands	PSNR	JPEG Compression
Shih and Wu (Shih and Wu, 2004)	DCT	MOOP/GA	Coefficient rounding rule	PSNR	None
Kumsawat and Attakitmongcol (Kumsawat *et al.*, 2005)	Discrete Multiwavelet Transform	MOOP/GA	Coefficients	Universal Quality Index	JPEG Compression LPF Wiener Filtering Gaussian Noise Image Cropping Image Rotation
Diaz and Romay (Díaz and Romay, 2005)	DCT	MOOP/NSGA-II	Coefficients	Coefficient Value	JPEG Compression Smoothing

Regarding the use of supervised learning, there are two main approaches. The first is to learn a watermarking process, as in (Chang and Lin, 2004a) where the mapping between the original and embedded DWT coefficients is performed with the use of a MLP. The main benefit of this type of approach is that it allows knowing the model of a given property of the the cover image on the detection side, which can boost detection performance, but without the burden of transmitting the cover image to the detector (informed detection). The second approach is to learn how to evaluate a given property of the watermarking process such as the visual impact or the robustness of the embedded watermark. The main benefit of this approach is that it makes possible modeling the visual impact according to evaluations provided by human viewers.

Another issue is that as mentioned before, bi-tonal images account for 95% of the use in industrial applications. Nevertheless, no single work was found in the literature, regarding the

Table 1.3 Summary of intelligent watermarking techniques based on supervised learning.

Method	Watermarking Strategy	Classifier	Feature Set
Chang and Lin (Chang and Lin, 2004a)	DWT	MLP	DWT coefficients.
Chang and Lin (Chang and Lin, 2004b)	Spatial domain.	SVM	Significant bits of a pixel.
Tahir *et al* (Tahir *et al.*, 2005)	DCT	SVM	Detection statistics.
Davis and Najarian (Davis and Najarian, 2001)	DWT	MLP	DWT coefficients.

use of evolutionary computing in the optimization of bi-tonal documents. Although most of the techniques presented are based on single-channel (grey-scale) images, they can be easily adapted to multichannel images (like RGB).

In the next section, the applicability of using an evolutionary computing algorithm (PSO) to this task is demonstrated.

1.4 Case study – optimization of a bi-tonal watermarking system using PSO

In this section a system that optimizes the bi-tonal watermarking system is proposed based on the system of Muharemagic (Muharemagic, 2004). The adaptive technique proposed by Muharemagic (Muharemagic, 2004) is based on greedy search and thus, does not consider the effect of choosing one parameter on the remaining parameters. A strategy to address this issue is to employ PSO to optimize these parameters simultaneously. Compared to the adaptive method proposed by Muharemagic, the use of PSO for this task allows a global search in the parameter space. Compared to other Evolutionary Algorithms (EA), PSO is considered to have a quick convergence.

In the proposed method, three embedding parameters (block size, shuffling seed and SNDM window size) are encoded as a particle position in the PSO algorithm. The first parameter (block size), a limited set of block sizes $B = \{b_i | i = 0, ..., (|B| - 1)\}$ is employed. The index of this set is used as one of the dimensions of the search space. For this reason, this given axis must be clipped to the $[0 - (|B| - 1)]$ range. The second parameter is the seed used to shuffle the image. The index of the set of seeds $K = \{k_j | j = 0, ..., (|K| - 1)\}$ is used as another dimension of the search space. Finally, the third parameter is the size of the SNDM window.

Let us define a set of SNDM window sizes B_s. As for the other parameters, the index of the set is used as a dimension of the search space. In a canonical PSO, the search space will have though, three dimensions. An alternative is to employ a discrete PSO (Kennedy and Eberhart, 1997) for this task, where each parameter can be encoded as a sequence of bits. These are exactly the same parameters used by Muharemagic.

As in the baseline adaptive watermarking system, there are three objectives to be minimized – the MSE (Equation 1.2) between the embedded and detected fragile watermarks, the MSE (Equation 1.2) between the embedded and detected robust watermarks and the DRDM (Equations 1.5 and 1.6) between the cover and watermarked images. Although there are multi-objective versions of the PSO in the literature (Coello *et al.*, 2004), for a matter of simplicity a single objective PSO, with function aggregation, is employed in this case study. Since the objective of this work is a proof-of-concept, the Conventional Weighted Aggregation (CWA) is applied.

1.4.1 Framework

1.4.1.1 Baseline watermarking system

The bi-tonal watermarking system proposed by Muharemagic (Muharemagic, 2004) will be the baseline watermarking system. In the proposed intelligent watermarking approach, two watermarks, a robust and a fragile, are embedded, along with Error Correction Code. This system embeds multi-bit messages in the spatial domain of bi-tonal images. Shuffling (Wu and Liu, 2004) is employed to handle uneven embedding capacity. Perceptual modeling is performed with the use of SNDM, which has been specifically developed for the bi-tonal spatial domain representation and is more flexible than its counterparts (Wu and Liu, 2004).

The message coding is based on code division, which comprises partitioning the image into blocks of a same size and then, encoding one bit at each block by manipulating the number of black pixels in each block. Regarding watermarking of bi-tonal images, one of the most simple techniques is to force the number of black pixels in a block to be either even (to embed a ‘0’) or odd (to embed a ‘1’). This technique is known as odd-even embedding. The problem with

this type of technique is that if a single pixel is changed in a block (due to noise, compression or even an attack), the bit value changes as well. A strategy to deal with this issue is to define a fixed quantization step size Q and to force the number of black pixels in a block to be either $2kQ$ or $(2k+1)Q$ (for a given k) (Chen and Wornell, 2001; Eggers *et al.*, 2003). This technique is known as Uniform Quantization (UQ). A larger Q will allow more pixels being randomly shifted (e.g. in the case of an attack) without changing the embedded bit value. This adds robustness to the watermark at the cost of more visual artifacts. Detection is done by checking the enforced relationship.

When compared to other bi-tonal watermarking techniques, the advantage of UQ is that it has been specifically designed for the embedding of multi-bit messages and has proven success in the watermarking of bi-tonal images (Wu and Liu, 2004). Moreover, UQ is not tied to a specific application – there are numerous techniques in the literature that have been developed for specific applications like watermarking of handwritten text document images (text or character shifting), *fac-simile* and others (Chen *et al.*, 2001; Yang and Kot, Dec. 2006; Puhan and Ho, 2005).

The projection of pixels in this baseline watermarking system is based on the manipulation of a property of the host signal (Type-II). Here, the watermark signal (w_a) computed for a block B is projected into the image by changing w_a flippable pixels on that block (black pixels are flipped to white if w_a is positive while the opposite happens if w_a is negative).

A UQ method named Scalar Costa Scheme (SCS) proposed by Eggers *et al* (Eggers *et al.*, 2003) will be employed since it is general and flexible when compared to the alternative (Chen and Wornell (Chen and Wornell, 2001)).
In this method, in order to embed a bit m_i into a given element of the cover signal x_i (in this case, quantity of black pixels at block i), quantization of the cover signal must first be performed

$$q_i = SQ_Q\{x_i - Q(\frac{m_i}{D} + \kappa_i)\} - (x_i - Q(\frac{m_i}{D} + \kappa_i)) \tag{1.8}$$

where $SQ_Q\{\}$ is the scalar uniform quantization operation, Q is the quantization step size, D is the alphabet size (2 for binary encoding), and κ_i is a pseudo-random number in the $[0, 1)$ range, used for security purpose.

The watermark signal (w_a) is obtained by multiplying q by the embedding strength α

$$w_a = \alpha q \tag{1.9}$$

The inverse process is done during detection. Here, the number of black pixels in a given partition box is the received signal w_n, that may have been attacked, that is $w_n = x + w_a + v$ (where v is the noise signal). Then, the detected message (m_n) is extracted from a given block with the use of the uniform quantizer. Firstly, quantization is applied to the received signal

$$q_{ni} = SQ_Q\{w_{ni} - \kappa_i Q\} - (w_{ni} - \kappa_i Q) \tag{1.10}$$

It is necessary to use the same Q and κ_i used on embedding.

Then, the message bit is extracted from q_{ni}. In our case, since the encoding comprises the block-based use of UQ, it is necessary to know here the partitioning scheme and the quantization step size employed on embedding. Basically, if the value of q_{ni} is close to either Q or 0, it means the corresponding bit is $m_{ni} = 0$. If instead, the value of q_{ni} is close to $Q/2$, it means the corresponding bit is $m_{ni} = 1$ (see Figure 1.7).
This process is repeated for each partition block. In this case study, the embedded message is a logo image. Given the dimensions of this image, it is possible to reconstruct the logo with the use of the detected bitstream.

An interesting aspect of this watermarking technique is that it allows for the embedding of several watermarks, at different levels of robustness (where robustness, as mentioned before, is determined primarily by the Q parameter). This process is called multi-level embedding and it basically comprises embedding the watermarks sequentially, starting with the one with the biggest value for Q until the one with the smallest value.

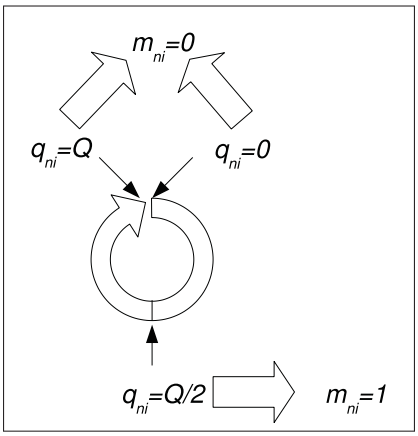


Figure 1.7 Detection decision.

1.4.1.2 Particle Swarm Optimization (PSO)

The use of PSO is justified by its quick convergence towards the global minimum in comparison with other popular methods – e.g. GA (Poli *et al.*, June 2007). The drawback is that the search performed by GA is more exhaustive. However, in the case study scenario, it is acceptable to have near-optimum solutions if they satisfy a predefined quality and robustness criteria. These two properties make PSO more suitable to the problem of fast optimization of watermarking systems.

In a seminal paper, Kennedy and Eberhart (Kennedy and Eberhart, 1995) proposed the Particle Swarm Optimization technique. The initial motivation behind their work was to graphically simulate the choreography of a bird flock. In PSO, a particle navigates through the search space based on two influences – the best position visited by itself (cognitive component) and the best position visited by its neighbour (social component). The neighbourhood of a particle can be restricted to a limited number of particles (L-Best topology) or the whole swarm (G-Best topology). The algorithm has passed through many different stages on its conceptual development. Parsopoulos and Vrahatis (Parsopoulos and Vrahatis, 2002), Kennedy (Kennedy, 2007) and Poli *et al* (Poli *et al.*, June 2007) provide a review of the improvements in PSO algorithm since its inception. The most popular implementation of PSO is known in the literature as the Canonical PSO. In this implementation, the velocity and position of a particle are updated at

each iteration according to the best locations visited by itself and by its best neighbour. Two different factors are employed to balance the influence of both attractors – the c_1 (cognitive) and c_2 (social) acceleration constants. To provide a fine grain search in the end of the optimization process, a inertia weight ($\omega_{inertia}$) and constriction factor (χ) were added. So the velocity and position of each particle in the Canonical PSO are calculated at each iteration as:

$$V_{id} = \chi \times (\omega_{inertia} \times V_{id} + c_1 \times r_1 \times (P_{id} - X_{id}) + c_2 \times r_2 \times (P_{gd} - X_{id})) \tag{1.11}$$

$$X_{id} = X_{id} + V_{id} \tag{1.12}$$

where V_i and X_i are the velocity and position of particle i, P_i is its best visited location, P_g is the best visited location for all of its neighbours. As mentioned before, the inertia weight controls the impact of previous history of velocities on the current one and the constriction controls the magnitude of the velocities. The most common approach in the literature is to fix χ. Regarding ω, the common approach is to set a large value at the beginning and then, gradually decrease it.

The method proposed in this case study employs the same notation as in (1.12), but uses only the $\omega_{inertia}$ (in the proposed method case, χ will be fixed to 1.0). During optimization this parameter is initialized with a large value, and then decreased, as seen in (Parsopoulos and Vrahatis, 2002).

As mentioned, the CWA approach is used in this case study for multi-objective optimization (MOO). Digital watermarking is in essence a multi-objective optimization problem. On its inception, PSO was able to handle only single objective problems. However, through a weighted sum of the fitness values, it is possible to aggregate several fitness functions (f_i) into a global one (F). This process is known as Weighted Aggregation (Parsopoulos and Vrahatis, 2002)

$$F = \sum_{i=1}^{N_{fitness}} \gamma_i f_i(x) \tag{1.13}$$

where γ_i is a non-negative weight, having $\sum_{i=1}^{N_{fitness}} \gamma_i = 1$ and $N_{fitness}$ is the number of fitness functions.

1.4.2 Simulation results

In this subsection the performance of the adaptive watermarking system based on the canonical PSO is evaluated for the optimization of the baseline watermarking system. This adaptive watermarking system is compared with the greedy technique proposed by Muharemagic (Muharemagic, 2004). The experiments were conducted in the CCITT database which was also used by Muharemagic (Muharemagic, 2004). This database is composed of 8 bi-tonal images (Figure 1.8). All images have the same dimension (2376×1728 pixels) and were scanned at 200 dpi.

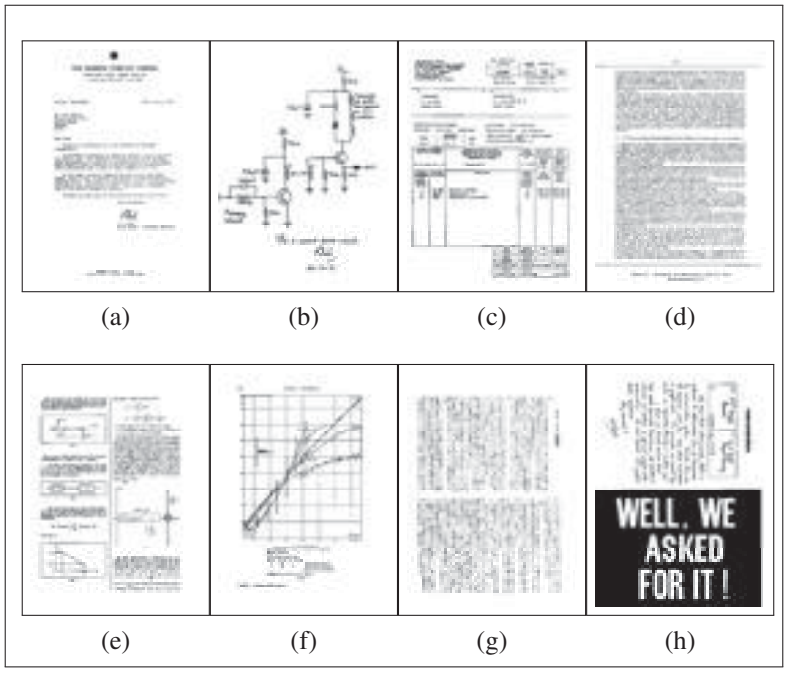


Figure 1.8 Samples from the CCITT database (left to right, top-down, CCITT1 to CCITT8).

The same 35×26 pixels OK and BIz binary logos (Figures 1.9a and b) were used as the fragile and robust watermarks, respectively. An eight-bit random number along with its CRC-4 code were appended to the logo in order to allow the search of the parameters during detection.

This resulted in a payload of 922 bits for each watermark (910 bits for the logo and 12 bits of self-verifiable data). For the robust watermark, $Q = 10$ and $\alpha = 0.7$ are employed while for the fragile watermark $Q = 2$ and $\alpha = 0.95$ are employed. The values of Q were chosen based on the literature (Muharemagic, 2004). For the α, a few different options were evaluated empirically, in order to find values leading to a similar DRDM as reported in (Muharemagic, 2004) for the given value of Q (the DRDM values reported in Table 1.4 are very similar to those reported by Muharemagic).

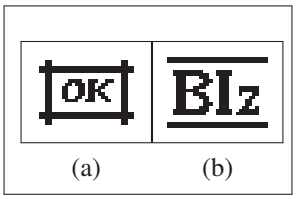


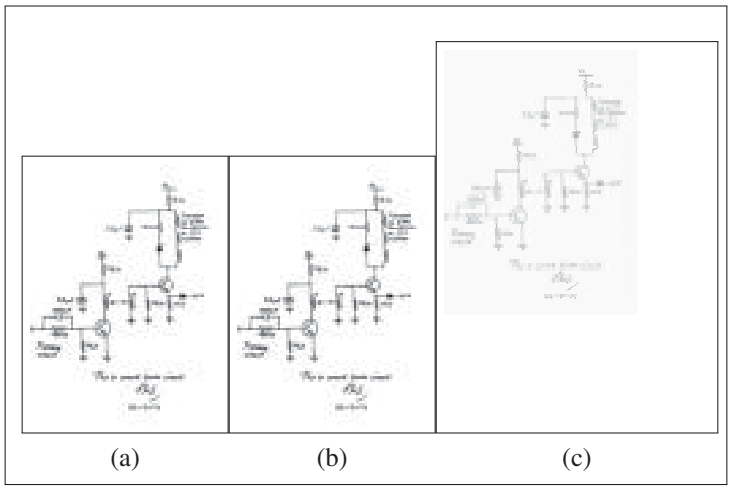
Figure 1.9 OK and BIz logos (Muharemagic, 2004).

It is interesting to observe that this baseline watermarking system provides high levels of quality and robustness for this proposed dataset and payload, even before optimization. For example, Figures 1.10 and 1.11 demonstrate the visual impact of embedding these two watermarks, using a 64×64 partition block size and a 3×3 SNDM window size into the CCITT2 image.

The robustness can be demonstrated by manipulating a given number of pixels in the watermarked image (as in (Muharemagic, 2004)). In Figure 1.12, the watermarked CCITT1 image was manually modified by 64, 128, 192 and 256 pixels respectively.

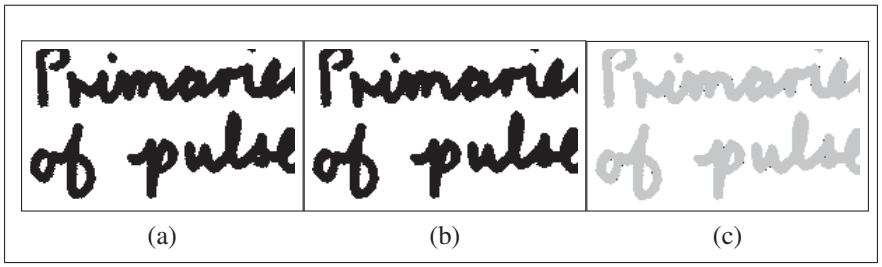
Figure 1.13 shows the detection of the BIz and OK logos in these scenarios. As expected, the robust watermark was more resistant against tampering. For the four attacks (64, 128, 192 and 256 pixel modifications in the watermarked image) only 0.2%, 0.8%, 0.8% and 1.3% of the pixels in the BIz logo, respectively, were corrupted against 6.4%, 11.3%, 16.4% and 20%, respectively, for the OK logo.

During experiments, five different options of block size were employed, that is $B = \{8 \times 8, 16 \times 16, 32 \times 32, 64 \times 64, 128 \times 128\}$. Regarding the shuffling key, a set containing 16 different randomly generated seeds was employed. Finally, three different SNDM window sizes were considered during optimization (3×3, 5×5 and 7×7).



(a) (b) (c)

Figure 1.10 Visual impact of multilevel embedding in the CCITT2 image. The BIz logo was embedded as a robust watermark ($Q = 10$ and $\alpha = 0.77$) while the OK logo was embedded as a fragile watermark ($Q = 2$ and $\alpha = 1$). (a) Original image. (b) Watermarked image. (c) Difference image.



(a) (b) (c)

Figure 1.11 Detail on visual impact of multilevel in the CCITT2 image. The BIz logo was embedded as a robust watermark ($Q = 10$ and $\alpha = 0.77$) while the OK logo was embedded as a fragile watermark ($Q = 2$ and $\alpha = 1$). (a) Original image. (b) Watermarked image. (c) Difference image.

Experiments were performed for each image and the parameters found were reported. The optimal watermark MSE results for the fragile and robust watermarks and the SNDM were also reported.

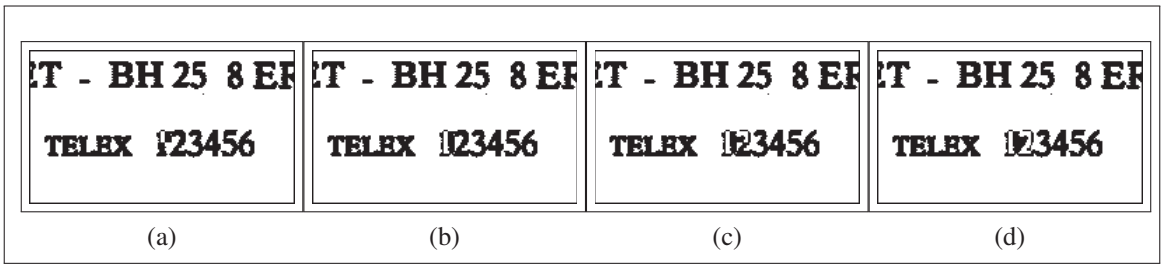


Figure 1.12 Flipping pixels on CCITT1 image. The BIz logo was embedded as a robust watermark ($Q = 10$ and $\alpha = 0.77$) while the OK logo was embedded as a fragile watermark ($Q = 2$ and $\alpha = 1$). Then, four different modifications were applied to image (a) Modification of 64 pixels. (b) Modification of 128 pixels. (c) Modification of 192 pixels. (d) Modification of 256 pixels.

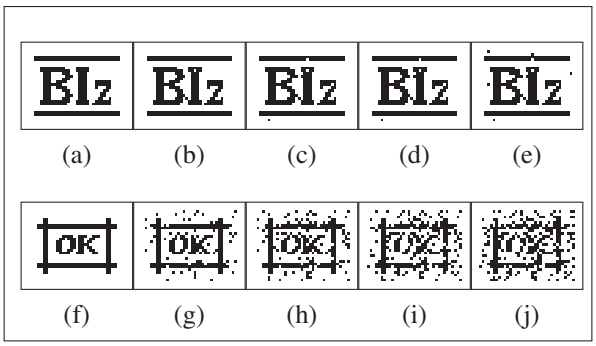


Figure 1.13 Detection of watermarks on watermarked/attacked CCITT1 image. A given number of pixels was modified in the watermarked image. Effect of modifying (a) No pixel, BIz watermark. (b) 64 pixels, BIz watermark. (c) 128 pixels, BIz watermark. (d) 192 pixels, BIz watermark. (e) 256 pixels, BIz watermark. (f) No pixel, OK watermark. (g) 64 pixels, OK watermark. (h)128 pixels, OK watermark. (i) 192 pixels, OK watermark. (j) 256 pixels, OK watermark.

1.4.2.1 Baseline adaptive system (Muharemagic, 2004)

As mentioned, the choice of block size is based on heuristic. Therefore, since the same payload was applied to all images, the block size chosen was always the same. The results can be seen in Table 1.4.

Table 1.4 Performance of the adaptive system proposed by Muharemagic in the CCITT database (Muharemagic, 2004).

Image	Block size	Key	SNDM window	MSE fragile	MSE robust	DRDM
CCITT1	64×64	5	5×5	15	29	0.0053
CCITT2	64×64	15	7×7	13	26	0.0067
CCITT3	64×64	8	3×3	19	30	0.0017
CCITT4	64×64	14	3×3	18	23	0.0002
CCITT5	64×64	5	3×3	17	28	0.0020
CCITT6	64×64	12	5×5	17	32	0.0050
CCITT7	64×64	1	3×3	14	30	0.0006
CCITT8	64×64	11	5×5	17	29	0.0034

1.4.2.2 Adaptive system based on PSO

The canonical PSO was employed to optimize the same three parameters (block size, shuffling key and SNDM window size). The swarm was composed of 20 particles, both cognitive and social constants were set to 2.05. The inertia weight was initialized with 1.2 and gradually decreased until 0.01. The number of iterations was set to 100. The maximum number of iterations without improvement in the global maximum was set to 10. The same aggregation weight of $\frac{1}{3}$ was employed for the three objective functions. The PSO topology chosen was the G-best (one single global best for the whole swarm). The results can be seen on Table 1.5.

Table 1.5 Performance of the canonical PSO version of the adaptive system proposed by Muharemagic in the CCITT database.

Image	Block size	Key	SNDM window	MSE fragile	MSE robust	DRDM
CCITT1	64×64	15	5×5	16	19	0.0056
CCITT2	32×32	5	7×7	17	23	0.0034
CCITT3	64×64	2	5×5	22	13	0.0017
CCITT4	64×64	1	3×3	22	15	0.0003
CCITT5	8×8	3	7×7	20	16	0.0005
CCITT6	32×32	9	7×7	18	16	0.0028
CCITT7	16×16	9	3×3	17	13	0.0003
CCITT8	32×32	2	7×7	16	25	0.0018

1.4.2.3 Discussion

On average, both the robustness of the robust watermark and the quality of the watermarked image were improved when compared to the Muharemagic adaptive system. It is interesting to observe that the robustness of the fragile watermark has degraded. This happened because in the adaptive method proposed by Muharemagic there is an "hierarchy" in optimization of the robustness of the watermarks. The first objective is to find parameters that improve the robustness of the fragile watermark. For PSO in contrast, the objective is to minimize a weighted sum of the three functions equally. Considering that the robustness of the robust watermark (which naturally requires more payload than the fragile mark) was increased with an improvement in the fidelity, it was expected that some of the channel capacity employed by the fragile mark would be transferred to the robust watermark. However there is no concern regarding this decrease in the robustness of the fragile mark since this type of watermark does not have to be robust anyway.

Another factor that led to the decrease in robustness of fragile watermark is that a single-objective version of PSO was employed. As mentioned before, such type of algorithm usually favours one objective in detriment of the others. A Pareto-based multi-objective evolutionary algorithm such as the NSGA-II (Deb, 2001) or MOPSO (Coello *et al.*, 2004) should probably find a more balanced trade-off between the three objectives.

It is possible to observe that the images are from different classes and this reflected in the optimal solution found by the PSO. For this reason, full optimization must be performed for each image, a costly process.

1.5 Conclusion

In this chapter a brief introduction to digital watermarking was provided, followed by an extensive survey on intelligent watermarking. As observed, most of the efforts in this area are concentrated in the optimization of embedding parameters with the use of evolutionary computing. One of the drawbacks of these approaches is the computational burden of optimization.

In intelligent watermarking, evolutionary computing is by far the most employed approach in what regards the optimization of embedding parameters. There are many reasons for this popularity, the simplicity of EC algorithms, their adaptability, which allows the direct application to many different types of digital watermarking techniques. However, intelligent watermarking has some drawbacks. One of them concerns the communication of embedding parameters to the detector. One of the main advantages of digital watermarking over other security techniques is the self-contained protection it offers. Notwithstanding, most watermarking systems require the knowledge of some of the embedding parameters on detection. A common approach is to use a fixed set of parameters and communicate them to all the detectors through a secure channel. But in intelligent watermarking, these parameters must be optimized according to each particular image. This can limit the application of intelligent watermarking, mainly in situations where the overhead of a secure channel to communicate these parameters is not acceptable. Another approach is to use part of the payload to either embed a training sequence, in the form of an Error Correction Code (ECC) or as a second watermark.

Another drawback of the use of EC in the optimization of digital watermarking systems is its high computational cost. Depending on the complexity of the problem, an EC algorithm such as PSO or GA can require thousands of fitness function evaluations. In intelligent watermarking this means thousands of costly embedding, detection and image processing (attack) operations. This limits intelligent watermarking to small sets of images. For this reason, decreasing the computational burden of evolutionary optimization techniques is a key issue, which can make possible the industrial use of intelligent watermarking.

As digital watermarking task comprises embedding a signal into an image in accordance with robustness and quality constraints, it can be said that it is in essence a multi-objective optimization problem. In the literature, many research works have tried to address this multi-objective problem with the use of a weighted sum of the objective functions. However, it is a known problem in evolutionary optimization that this approach usually favours one objective in detriment of the others. The alternative is instead of conducting a search for a single global solution

to the problem, try to find a set of nondominated solutions, known as Pareto-optimal as in (Sal *et al.*, 2006; Díaz and Romay, 2005).

Supervised learning has also been used in the context of creating adaptive watermarking systems. Most classifiers will rely in some sort of optimization (e.g. gradient descent for MLP and quadratic programming for SVM). However, differently than in EC-based intelligent watermarking, optimization is employed in the task of finding optimal parameters for the classifiers and not for the watermarking systems. These classifiers are then employed in intelligent watermarking either replacing a watermarking process or in the evaluation of a watermarking property.

As a case study, PSO was employed to the task of optimizing a bi-tonal watermarking system. It was possible to observe in this simulation that although EC can be useful in the task of finding a near-optimum trade-off between robustness and quality, the performance is bounded by theoretical limitations of the watermarking system (embedding capacity, etc). The advantage of PSO over greedy search is that it allows a paralel search for optimal parameters (that is, adjusting more than one parameter at the same time). This can make possible the optimization of more complex parameters such as the embedding strength (α) and the quantization step size (Q). Adjusting these parameters against some removal attacks can be considered a future work. Another important issues to be addressed include: decreasing the computational burden of EC and employing a Pareto-based multi-objective optimization technique.

1.6 Discussion

In this chapter we presented some of the key issues in intelligent watermarking. We also demonstrated through a proof-of-concept simulation the main advantages and limitations regarding the use of EC for the automatic adjustment of embedding parameters. The baseline bi-tonal watermarking system employed through the rest of this thesis was presented in details.

However, one of the limitations of the techniques presented in this chapter is their elevated computational burden. In most cases, automatic adjustment of embedding parameters through

EC involves thousands of embedding and detection operations per image. For this reason, such type of approach is limited to small, proof of concept applications.

But in optimization of embedding parameters for streams of document images, the optimization problems associated with these images are expected to be similar which makes possible reusing knowledge about previous optimization tasks in order to decrease the cost of optimization. Based on this insight, in the next chapter we investigate some important properties of this stream of optimization problems formulation of intelligent watermarking and propose a strategy to curb the computational cost in such scenarios.

CHAPTER 2

HIGH THROUGHPUT INTELLIGENT WATERMARKING OF HOMOGENEOUS STREAMS OF BI-TONAL IMAGES

In this chapter, a novel intelligent watermarking technique based on Dynamic Particle Swarm Optimization (DPSO) is proposed. The main objective here is to formulate intelligent watermarking of bi-tonal image streams as a dynamic optimization problem and to devise a technique that allows detecting and measuring the severity of changes in such type of problem. This population-based technique allows to evolve a diversified set of solutions (i.e., embedding parameters) to an optimization problem, and solutions from previous optimizations are archived and re-considered prior to triggering new optimizations. In such case, costly optimization may be replaced by direct recall of quasi identical solutions. Simulations involving the intelligent watermarking of several long streams of homogeneous PDF document images resulted in a decrease of computational burden (number of fitness evaluations) of up to 97.2% with a negligible impact on accuracy. The content of this chapter was published at the 10^{th} International Conference on Intelligent Information Hiding and Multimedia Signal Processing (Vellasques *et al.*, 2010b) and Applied Soft Computing (Vellasques *et al.*, 2011).

2.1 Introduction

The digitalization, storage and transmission of document images plays a vital role in many sectors, including government, health care and banking. Modern scanning devices have produced massive quantities of digitized documents, a situation that poses serious privacy threats, because most of these documents contain sensitive fiscal, medical and financial information. The enforcement of the confidentiality, integrity, and authenticity of these images has therefore become a very active research topic.

Cryptography has traditionally been employed as a mean of enforcing these aspects for different types of data. However, as pointed by Cox *et al* (Cox *et al.*, 1996), in conventional cryptographic systems, once the data is decrypted there is no way to track its reproduction or transmission. Digital watermarking, which is the practice of imperceptibly altering an image in

order to embed a message about it (Cox *et al.*, 2002) is a complement to cryptography and, can be employed in order to enforce the integrity and authenticity of document images. Although digital watermarking has been successfully employed in the protection of many different types of media such as audio, video and images (Wu, 2001), this chapter will focus on the watermarking of long streams of bi-tonal document images with similar structure as bank cheques, for example. The three main properties of a digital watermarking system are the data payload or capacity (amount of information that can be embedded within an image), robustness (watermark resistance against intentional and unintentional image processing operations) and fidelity (similarity between original and watermarked images). A gain in one of these properties usually comes at the expense of a loss in others.

Each application has a different requirement with regards to payload, fidelity and robustness. A common approach is to employ Constant Embedding Rate (CER) to set the payload, and to find an optimal trade-off for the other two properties. A watermark that is robust enough to resist attacks is embedded (as long as these attacks do not affect the commercial value of the watermarked image), without introducing visual artifacts. Some watermarking systems allow embedding multiple watermarks with different robustness levels (Wu and Liu, 2004). A robust watermark is usually employed in the enforcement of authenticity since it can survive some attacks while a fragile one is easily destroyed by tampering and can be employed in the enforcement of integrity.

Finding the optimal trade-off between fidelity and robustness is a very challenging problem because the payload varies for different types of images. In intelligent watermarking (Vellasques *et al.*, 2010a), evolutionary computing (EC) techniques such as Genetic Algorithms (GA) (Holland, 1992) and Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) have been proposed to determine the optimal embedding parameters for each specific image. The basic principle is to evolve a population of potential embedding parameters through time using a mix of robustness and quality metrics as objective function (Areef *et al.*, 2005; Arsalan *et al.*, 2010; Chen and Lin, 2007; Ji *et al.*, 2006; Kumsawat *et al.*, 2005; Shieh *et al.*, 2004; Shih and Wu, 2004; Pan *et al.*, 2004; Wei *et al.*, 2006; Wu and Shih, 2006). Genetic programming has also

been proposed in a similar context (Khan and Mirza, 2007). Such EC-based approaches are not feasible in high data rate applications because of their high computational cost (Chen and Lin, 2007), mainly because the optimization of a single image may require hundreds of thousands of embedding and detection operations (Kumsawat *et al.*, 2005; Shieh *et al.*, 2004; Shih and Wu, 2004).

In this chapter, it is hypothesized that when optimizing a large number of images of a same nature, it is possible to employ the knowledge acquired in previous optimization tasks in order to decrease the cost associated with frequent re-optimizations (Blackwell and Bentley, 2002). Knowledge of previous intelligent watermarking tasks has already been employed as a manner of improving subsequent tasks (Khan *et al.*, 2008; Usman and Khan, 2010). In such scenarios, the inherent optimization problems would share some similarity and intelligent watermarking can be cast as a single, long-term, *dynamic* optimization problem (DOP), instead of multiple, isolated, static problems. In a DOP, the optimum changes with time. Nickabadi *et al* (Nickabadi *et al.*, 2008) observed that there are three different types of changes:

- Type I – Optimum location changes with time.
- Type II – Optimum fitness changes with time (but location remains fixed).
- Type III – Both, the location and fitness of the optimum change with time.

The authors also characterize a DOP according to change severity in both time (called *temporal severity*) and space (*spatial severity*). Yang and Yao (Yang and Yao, 2008) categorize environment changes in two groups: periodical, where changes occur in a fixed time interval and cyclical, where several fixed states occur repeatedly.

For an intelligent watermarking system, the moment an image transition occurs is known. Therefore, the temporal severity will be considered negligible, the problem is to be said pseudo-dynamic. More specifically, intelligent watermarking can be formulated as a specific type of DOP where a change is followed by a period of stasis (Farina *et al.*, 2004). In the envisioned scenario, an image transition should result in an environmental change. No change is expected to happen during the optimization of a single image. Thus, in the envisioned scenario, a stream

of document images will correspond to a stream of optimization problems rather than a single problem where the optimum (or optima) continuously changes with time.

In this formulation of intelligent watermarking as a DOP, changes of type I are not expected as hardly two images will result in exactly the same fitness. Two images with very similar structure should result in a very similar set of optimal embedding parameters. That is, the set of embedding parameters can be either exactly the same or very similar, with only a small variation on their fitness values. The hypothesis we pose is that a transition between such similar images should result in either a change of type II (for the first case) or in a non-severe change of type III (for the second case). However, we will treat both as changes of type II. For the former, the location is exactly the same and it can be said that both optimal solutions are equivalent. For the later, the variation is still within the area surveyed by the population of solutions and thus there might exist other equivalent solutions that can be employed interchangeably for both problem instances. Two images with different structure by another way should result in considerable difference in both, set of optimal embedding parameters and respective fitness values. For this reason, a transition between them would imply in a severe change of type III. In such scenario, intelligent watermarking can be considered as a special case of cyclical problem (Yang and Yao, 2008) as similar rather than static states reappear over time.

For the scenario considered in this chapter, embedding parameters will be optimized for a large stream of similar bi-tonal document images. For two similar images, $\mathbf{Co}_1$ and $\mathbf{Co}_2$, the change between the respective optimization problems would be of type II. The respective sets of optimal embedding parameters and fitness values are also expected to be similar. Since existing intelligent watermarking methods optimize embedding parameters for each image, computational time is wasted in optimizing parameters for a previously seen image. In such case, the population of solutions (sets of embedding parameters) obtained in the optimization of $\mathbf{Co}_1$ may have one or more solutions for $\mathbf{Co}_2$ that are comparable to those that would be obtained by performing complete re-optimization. Given two other images with considerably different structure, $\mathbf{Co}_3$ and $\mathbf{Co}_4$, the change would be of type III and re-optimization would be necessary as their respective optimal embedding parameters and fitness values are also expected to

be very different. However, existing change detection methods do not provide means of measuring the similarity between two optimization problems which would allow re-using solutions for changes of type II.

In this chapter, fast intelligent watermarking of streams of document images is formulated as a DOP and tackled with the use of a novel technique based on Dynamic PSO (DPSO) since canonical PSO cannot tackle some issues in a DOP like outdated memory, lack of a change detection mechanism and diversity loss (Blackwell, 2007; Carlisle and Dozier, 2002). In the proposed technique, solutions of previous problems are stored in a memory and recalled for similar problems. An adaptive technique to measure the similarity between optimization problems associated with two different images (change detection) is also proposed. This technique allows distinguishing between changes of types II and III. The main application of the proposed method is to tackle intelligent watermarking of long, homogeneous streams of document images. Both, this formulation of intelligent watermarking as a dynamic optimization problem and the adaptive change detection mechanism are unprecedented.

Proof-of-concept simulations are performed with the use of a general bi-tonal watermarking system based on odd-even embedding and quantization (Muharemagic, 2004; Wu and Liu, 2004). Two databases containing binarized pages of scientific documents were employed in these simulations. Simulation results demonstrate that this approach resulted in significant decrease in the computational cost of intelligent watermarking by avoiding costly optimization operations but with nearly the same accuracy of optimizing each image.

This chapter is organized as follow. Section 2.2 presents a survey of digital watermarking and provides a baseline system for bi-tonal images. A baseline system for intelligent watermarking of isolated bi-tonal images based on PSO is presented in Section 2.3. The fast intelligent watermarking system based on DPSO is proposed in Section 2.4. Finally, Section 2.5 provides experimental results and discussions.

2.2 Digital watermarking methods for bi-tonal images

Since the intended application is the watermarking of document images (which is mostly based in bi-tonal encoding), a baseline bi-tonal watermarking method will be presented. Bi-tonal (or binary) watermarking offers additional challenges when compared to greyscale and color watermarking since in a bi-tonal image, pixels can only have two values – black or white – thus any variation tend to be more perceptible than in color and greyscale images. There are numerous bi-tonal watermarking techniques in the literature (Awan *et al.*, 2006; Mei *et al.*, Jan. 2001; Muharemagic, 2004; Pan *et al.*, 2000; Ho *et al.*, 2004a; Tseng and Pan, 2001; Wu and Liu, 2004; Yang and Kot, Dec. 2006; Zhang and Qiu, 2005; Zhao and Koch, 1995). A survey of such techniques can be found in (Chen *et al.*, 2001). The main drawback of bi-tonal watermarking is that most techniques were conceived to deal with very specific applications like printed text, handwritten text, half-toned images or a certain class of watermarks (robust or fragile).

The bi-tonal method of Wu and Liu (Wu and Liu, 2004) is employed as the baseline watermarking method in this research since it is general and allows embedding multiple watermarks at the same image with different levels of robustness. This technique is based on odd/even embedding, where basically the image is partitioned into several blocks of equal size ($B \times B$ pixels) and pixels are flipped in order to set the number of black pixels to either an odd number (to embed a '0') or an even number (to embed a '1'). The number of pixels to flip is quantized (Chen and Wornell, 2001; Eggers *et al.*, 2003) as a manner of allowing robust watermarking. In this method, a bit m is embedded into the i^{th} block of the cover image **Co** by manipulating the number of black pixels on that block (N_P) with the use of quantization

$$w_a = Q_\Delta\{N_P - Q(\frac{m}{2} + r)\} - (N_P - Q(\frac{m}{2} + r)) \tag{2.1}$$

where $Q_\Delta\{\}$ is the scalar uniform quantization operation, Q is the quantization step size and r is a pseudo-random number in the $[0, 1)$ range. The new number of black pixels on block i (N'_P) is computed as

$$N'_P = N_P + w_a \tag{2.2}$$

Detection is performed by an inverse process. Image is partitioned using the same block size and a bit is detected from block i by verifying the number of black pixels on it (N''_P, which might be different than N'_P if image has been attacked)

$$w_n = Q_\Delta\{N''_P - rQ\} - (r - N''_P Q) \tag{2.3}$$

The detected bit m_n is set to 0 if the value of w_n is close to either 0 or Q. Otherwise (closer to $Q/2$), it is set to 1. This is depicted in Figure 2.1. Basically, the value of w_n will be in the $[0, Q]$ range and we have $|w_n| \leq |w_n - Q/2|$ when it is closer to 0 than to $Q/2$ and $|w_n - Q| \leq |w_n - Q/2|$ when it is closer to Q than to $Q/2$.

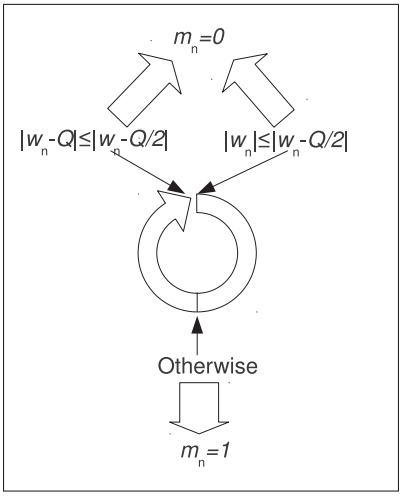


Figure 2.1 Detection decision.

Flipping pixels in uniform areas results in visual artifacts. Flippability analysis techniques (Muharemagic, 2004; Ho *et al.*, 2004a; Wu and Liu, 2004; Zhang and Qiu, 2005) tackle this problem by assigning a score to each pixel based on properties of its neighborhood. A window of size $W \times W$ is employed in this process. The Look-up Table (LUT) method proposed by Wu and Liu (Wu and Liu, 2004) uses a fixed 3×3 window in order to assign a score to a pixel based on the smoothness and connectivity of its neighborhood. A look-up table containing all the possible $2^{3\times3}$ patterns is built and the score for every pattern is calculated. For major window sizes, creating a look-up table becomes prohibitive.

Muharemagic (Muharemagic, 2004) proposes a more flexible technique named Structural Neighborhood Distortion Measure (SNDM). This method uses a reciprocal distance matrix **D** in order to compute the flippability of a pixel, based on its $W \times W$ neighborhood. The SNDM of a candidate pixel (i, j) of image **Co** is computed as follows:

$$\mathbf{SNDM}_{i,j} = \frac{\sum_{k=-\frac{W}{2}}^{\frac{W}{2}} \sum_{l=-\frac{W}{2}}^{\frac{W}{2}} (\mathbf{Co}(i,j) \oplus \mathbf{Co}(i+k, j+l)) \times \mathbf{D}_{k+\frac{W}{2}, l+\frac{W}{2}}}{\sum_{k=1}^{W} \sum_{l=1}^{W} \mathbf{D}_{k,l}} \tag{2.4}$$

where **D** is defined as:

$$\mathbf{D}_{i,j} = \begin{cases} 0, & \text{if } (i,j) = \frac{W}{2} \\ \frac{1}{\sqrt{(i-\frac{W}{2})^2 + (j-\frac{W}{2})^2}}, & \text{otherwise} \end{cases} \tag{2.5}$$

After that, pixels are shuffled using a pseudo-random sequence based on a seed S in order to distribute flippable pixels evenly across the image. Muharemagic (Muharemagic, 2004) observed that some seeds result in better (more uniform) shuffling than others for a given image. Thus, the use of a pool of shuffling seeds is preferred. After embedding, the image is de-shuffled. Detection consists of partitioning the image with the same block size used on embedding, shuffling all pixels (using the same key as well) and detecting the embedded bit on each block using the quantized detector. Flippability analysis is not necessary on detection as pixels do not need to be flipped. This watermarking process is explained in details in (Wu and Liu, 2004) while an explanation of the SNDM technique can be found in (Muharemagic, 2004).

2.3 Intelligent watermarking of isolated images using Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) (Poli *et al.*, June 2007) is an optimization technique inspired on the behavior of bird flocks. It relies on a population (swarm) of candidate solutions (particles). Each particle navigates in a multidimensional search space (or fitness landscape) guided by the best position visited by itself (cognitive component) and by its best neighbor (social component). A particle i has a position $\mathbf{x}_i$ and velocity $\mathbf{v}_i$ which are updated according

to:

$$\mathbf{v}_i = \chi \times (\mathbf{v}_i + c_1 \times r_1 \times (\mathbf{p}_i - \mathbf{x}_i) + c_2 \times r_2 \times (\mathbf{p}_g - \mathbf{x}_i)) \tag{2.6}$$

$$\mathbf{x}_i = \mathbf{x}_i + \mathbf{v}_i \tag{2.7}$$

where χ is a constriction factor, chosen to ensure convergence (Blackwell, 2005), c_1 and c_2 are respectively the cognitive and social acceleration constants (they determine the magnitude of the random forces in the direction of $\mathbf{p}_i$ and $\mathbf{p}_g$ (Poli *et al.*, June 2007)), r_1 and r_2 are two different random numbers in the interval $[0, 1]$, $\mathbf{p}_i$ is the best location visited by particle i and $\mathbf{p}_g$ is the best location visited by all neighbors of particle i. PSO parameters c_1 and c_2 are set to 2.05 while χ is set to 0.7298 as it has been demonstrated theoretically that these values guarantee convergence (Poli *et al.*, June 2007). The neighborhood of a particle can be restricted to a limited number of particles (L-Best topology) or the whole swarm (G-Best topology). The particle encoding employed in this system can be seen in Table 2.1. Basically, the block size has lower bound of 2×2 and upper bound of 62×62 pixels (maximum possible for the given watermark size, considering the dimension of the images in the database). The remaining bounds, ΔQ, SNDM window size and number of shuffling seeds were defined based on the literature (Muharemagic, 2004).

Table 2.1 Range of embedding parameter values considered for PSO algorithm in this chapter.

Embedding Parameter	**Particle Encoding**
Block Size (B): $\{2, 3, 4, ..., 62\}$	$x_{i,1} : \{1, 3, 4, ..., 61\}$
Difference between Q for the robust (Q_R) and fragile (Q_F) watermarks (ΔQ): $\{2, 4, 6, ..., 150\}$	$x_{i,2} : \{1, 2, .., 75\}$
SNDM window size (W): $\{3, 5, 7, 9\}$	$x_{i,3} : \{1, 2, 3, 4\}$
Shuffling seed index (S): $\{0, 1, 2, ..., 15\}$	$x_{i,4} : \{0, 1, 2, ..., 15\}$

Since one of the parameters in the intended application is a random shuffling seed (S) which leads to a multi-modal fitness landscape, L-Best topology will be employed in the proposed technique as it is known to outperform G-Best in such situation (Parsopoulos and Vrahatis, 2002). During initialization, each particle is set to communicate with its k-nearest neighbors

(neighborhood is based on Euclidean distance). During optimization, the link between particles is changed in a random manner if no improvement occurs after one generation as a mean of improving adaptability (Clerc, 2006). Regarding the neighborhood size, we propose setting k to 3 as it is common found in the literature (Kapp *et al.*, 2009).

The application of PSO in the optimization of embedding parameters is straightforward. In the envisioned application, a population of potential solutions is initialized randomly according to the bounds defined in Table 2.1. Then, at each generation, the fitness of each particle $\mathbf{x}_i$ is evaluated in the task of watermarking a given image and $\mathbf{x}_i$ is adjusted according to Equation 2.7. The fitness evaluation consists of embedding a robust and a fragile watermark – and is depicted in Figure 2.2 where **Co** is the cover image, $\mathbf{m}_R$ and $\mathbf{m}_F$ are the robust and fragile watermarks, respectively, **Cr** is the robust watermarked image, **Crf** is the image that has been watermarked with both, the robust and the fragile watermarks (multi-level watermarked image), **Crf′** is the multi-level watermarked/attacked image, $\mathbf{m}_{RAD}$ is the robust watermark that has been detected from the multi-level watermarked/attacked image, $DRDM$ is the Distance Reciprocal Distortion Measure, BCR^{-1} is the inverse of the Bit Correct Ratio (Areef *et al.*, 2005; Pan *et al.*, 2004) between $\mathbf{m}_R$ and $\mathbf{m}_{RAD}$, ω_1 is the weight assigned to BCR^{-1} and ω_2 is the weight assigned to $DRDM$.
The robust watermark is embedded first at a quantization step size Q_R and then, the fragile watermark is embedded at a quantization step size $Q_F < Q_R$. The robustness of the fragile watermark can be set to a fixed, small value (as it has to be destroyed in the event of an attack). For the robust watermark, the difference between both $\Delta Q = Q_R - Q_F$ will be optimized, with $Q_F = 2$. A secure channel is assumed to be available for the transmission of the optimal embedding parameters (Table 2.1).

Robustness is computed by embedding both watermarks, attacking the image, detecting the robust one and computing the inverse of the Bit Correct Ratio (BCR) (Areef *et al.*, 2005; Pan *et al.*, 2004) between the embedded and detected watermarks. As mentioned before, the fragile watermark does not need to be optimized and for this reason its robustness is not considered in the fitness evaluation. Quality is computed with the use of the Distance Reciprocal Distortion

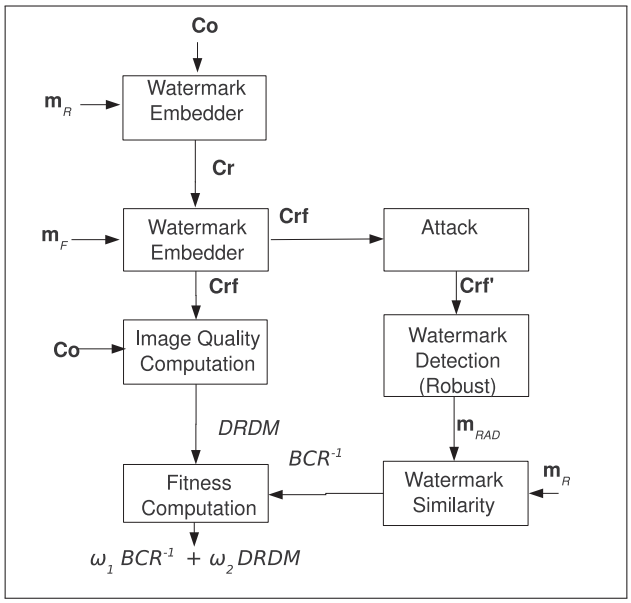


Figure 2.2 Fitness evaluation.

Measure (DRDM) (Muharemagic, 2004) which is also based on the use of a reciprocal distance matrix with size $W \times W$. This limits the number of objective functions to two (which must be minimized). Both metrics are combined with the use of weighted aggregation (Parsopoulos and Vrahatis, 2002):

$$F(\mathbf{x}_i) = \omega_1 BCR^{-1} + \omega_2 DRDM \tag{2.8}$$

where ω_1 is the weight assigned to the robustness metric, BCR^{-1} is the robustness metric, ω_2 is the weight associated with the quality metric and $DRDM$ is the quality metric. The weights are non-negative and $\omega_1 + \omega_2 = 1$.

More formally, a particle $\mathbf{x}_i$ represents a position in a 4-dimensional, discrete parameter space ($\mathbf{x}_i \in \mathbb{Z}^4$), with lower bound in $(1, 1, 1, 0)$ and upper bound in $(61, 75, 4, 15)$. This particle is mapped to a fitness function $F(\mathbf{x}_i)$ which consists of a weighted sum of the quality and robustness measurements obtained in a watermarking task involving the embedding parameters encoded by $\mathbf{x}_i$. The fitness landscape comprises the combination of both, parameter and fitness space.

2.4 Fast intelligent watermarking of image streams using Dynamic PSO

The proposed method assumes a long stream of bi-tonal document images ($\{\mathbf{Co}_1, ..., \mathbf{Co}_N\}$) and operates in two modes – a recall mode, where previously seen solutions are recalled from a memory and employed directly (avoiding re-optimization) and an optimization mode where the embedding parameters are optimized with the use of the L-Best PSO method described earlier until a certain stop criterion is met. Optimization will be halted whenever the global best has not improved for a given number of generations. The reason for choosing this criterion is that it is commonly found in the literature and it is not sensible to the number of generations chosen (Zielinski and Laur, 2007). There are two levels of memory. The first one is named Short Term Memory (STM), which in our notation is represented by $\mathfrak{M}_S$ and contains all the particles obtained in the optimization of a single image, that is, the whole swarm. This set of particles will be called a **probe**. The second one is named Long Term Memory (LTM), represented by $\mathfrak{M}$ and contains probes obtained in the optimization of different images. Since optimization will only be triggered when images have different structure, given two probes $\mathfrak{M}_1$ and $\mathfrak{M}_2$, the solutions found in $\mathfrak{M}_1$ should be very distinct from the solutions found in $\mathfrak{M}_2$.

For each image, an attempt to recall the STM is made. If this recall is not successful, an attempt to the LTM is made. Change detection is employed during a recall in order to measure the similarity between the fitness landscape of current image and the fitness landscape of the image for which that probe was obtained. When STM/LTM recall fails, the best solutions from the STM probe are injected into the swarm replacing its worst solutions (those which resulted in poorest combination of quality and robustness) and optimization is triggered. Thus, the STM provides a first level of recall and memory-based immigrants for the swarm (Wang, 2007). Regarding the amount of immigrant solutions, we propose injecting the best 70% particles, inspired by the results reported in (Kapp *et al.*, 2009), which employed the same amount of random rather than memory-based immigrant solutions.

This approach tackles diversity loss in a more precise manner than randomizing the entire swarm (Wang, 2007). The proposed method is illustrated in Figure 2.3. Here, starting with a first image ($\mathbf{Co}_1$), the swarm is initialized randomly (i) and optimization is performed until a

stop criterion is attained, resulting in an optimized swarm (ii). A probe is created with the particles of the swarm obtained, put in the LTM (I) and copied to the STM (iia). Then, for image $\mathbf{Co}_2$ the global best is successfully recalled from the STM. A recall is successful whenever the difference between the distributions of fitness values of both probes is smaller than a critical value D_α. For image $\mathbf{Co}_3$ an alternative solution is recalled from the STM. Then, for image $\mathbf{Co}_4$, the STM recall fails and since the LTM probe is identical to the STM probe, the best solutions in the STM probe are injected into the swarm, replacing its worst solutions (iii). Optimization is triggered and results in another optimized swarm (iv). A second probe is created and put into the LTM (II). The probe in the LTM with the highest number of successful recalls (I) is copied to the STM (iib). Images $\mathbf{Co}_5$, $\mathbf{Co}_6$ and $\mathbf{Co}_7$ result in unsuccessful STM recalls. However, another LTM probe (II) is successfully recalled in both cases and its successful recall counter becomes greater than that of the current STM probe which is then replaced by it (iva).

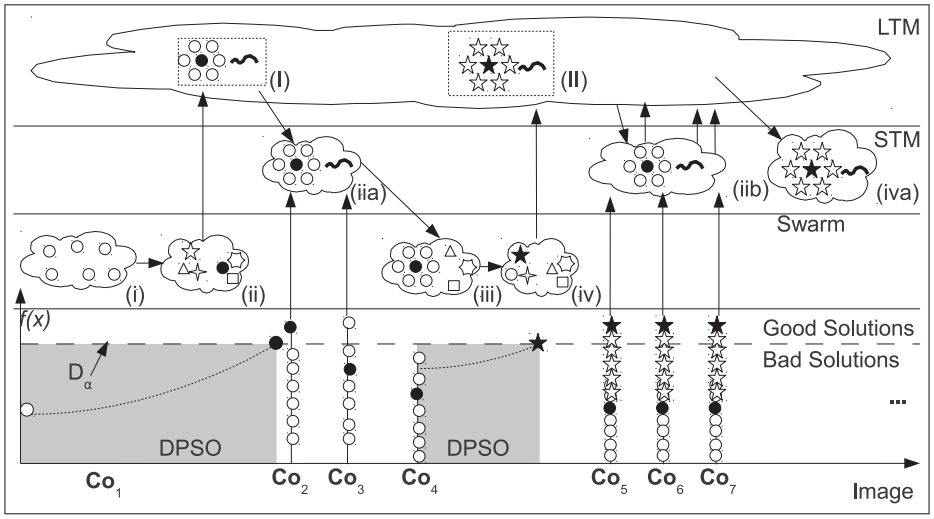


Figure 2.3 Overview of the proposed method.

During a transition between two images, the corresponding fitness landscape change can be either of type II (usually for images of a same nature) or III (images of different nature). It is possible to decrease computational burden by avoiding re-optimization for changes of type II – as for images $\mathbf{Co}_1$, $\mathbf{Co}_2$ and $\mathbf{Co}_3$ – since the optimum remains in the same location. Moreover, it is also possible to decrease computational burden of re-optimization for cases of type III by initializing the swarm with a few solutions from a previous optimization problem.

2.4.1 Change detection

The common strategy to perform change detection is to use fixed (sentry) particles (either from the swarm or from the memory (Branke, 1999; Yang and Yao, 2008)) and re-evaluate their fitness at each generation. However, such approach fails in detecting changes that occurred in restricted areas of the landscape as they are based on the assumption that if the optimum location of the fitness landscape changes, the fitness of any solution will also change (Carlisle and Dozier, 2002). The alternative is to choose one or more solutions randomly as sentries (Carlisle and Dozier, 2002). However, this approach does not allow measuring the severity of a change.

Wang *et al* (Wang *et al.*, 2007) try to tackle this problem by computing a running average of the fitness function for the best individuals over a certain number of generations, determining the severity based on a threshold. But this approach has two limitations in intelligent watermarking. The first is, it is not possible to put a threshold on variations of fitness value because of the issue regarding images with different capacity. The second is, since only the best solution is employed, it does not provide considerable information about the landscape.

Existing change detection methods use a limited number of sentries for a simple reason, they try to limit the number of fitness evaluations necessary in such process as in most of these cases, it needs to be performed at each generation. However, considering that in the envisioned scenario change detection will only be performed during image transitions, a larger number of sentries might be employed with little impact on the overall computational burden in cases where re-optimization is necessary and a significant decrease in cases where it is avoided. An intuitive approach is thus, use all particles as sentries. Considering that an appropriate diversity preserving mechanism is in place, such approach would provide more information about change in the fitness landscape than those based on single sentries. The L-Best topology employed in our technique maintains the diversity throughout the optimization process and should result in diverse enough probes. In the proposed change detection mechanism, the severity of a change between two images $\mathbf{Co}_i$ and $\mathbf{Co}_{i+1}$ is measured by evaluating the fitness

value (Figure 2.2) of a memory probe in both images and comparing the similarity between the respective distributions of fitness values with the use of a statistical test.

A slight difference between both distributions of fitness values might be due to change of type II or a non-severe change of type III. In such case, probe solutions could be employed directly, avoiding re-optimization. A severe difference between both distributions otherwise, can be due to a severe change of type III and re-optimization should be triggered. It is important to observe that the distinction between change types II and III is inferred indirectly as in a change of type III, the location of new optimum cannot be found with the use of sentries, requiring re-optimization for this end. However, in such situation, the severity of the variation for a given sentry is expected to be high, mainly if that sentry is positioned over a narrow peak. Since the distinction is based on the fitness distribution of sentry particles it is possible that two visually distinct images result in a similar distribution of fitness values. This means that although the images are different, their embedding capacity is equivalent and therefore, their optimal embedding parameters can be employed interchangeably.

Using a statistical test in the change detection process allows measuring the difference between the distributions of fitness values of a group of sentry particles in two different images. Such approach should provide much more information about a variation in the landscape than comparing fitness of isolated sentries. Figure 2.4 illustrates this process for a slight variation in the landscape which might be due to a type II change in an hypothetical situation where the change in the landscape (due to an image transition) was completely symmetrical, that is probe 4 has now the same fitness value as probe 1 in the previous landscape and so forth. In such case, both distributions would be identical but the best solution for the first image (number 1) would not be the best for the next image. However, another solution (number 4) would be equivalent to previous best solution.

Figure 2.5 illustrates the behavior of the change detection mechanism for a severe variation in the landscape, which might be due to a type III change. Here, both distributions differ significantly, no solution in probe could be employed directly in the second image.

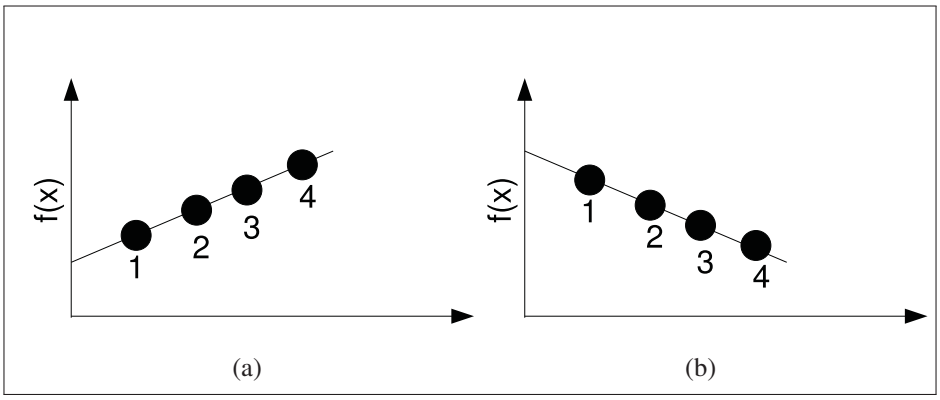


Figure 2.4 Illustration of a perfectly symmetrical type II change. (a) Fitness values of sentry particles for first image. (b) Fitness values of sentry particles for second image.

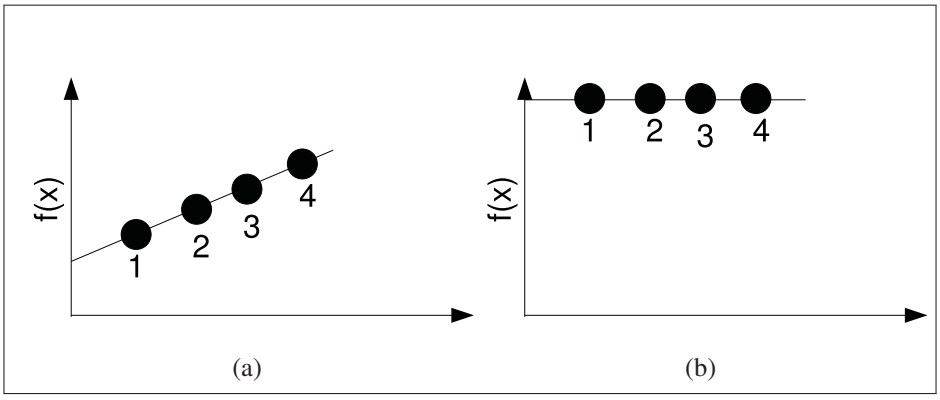


Figure 2.5 Illustration of a type III change. (a) Fitness values of sentry particles for first image. (b) Fitness values of sentry particles for second image.

Since the exact form of the fitness landscape is not known, no assumption can be made about the form of the distribution of fitness values of a probe. For this reason, the statistical test must be non-parametric. The Kolmogorov-Smirnov test (KS-test) (NIST/SEMATECH, 2010) is a non-parametric statistical test that can be employed to compare the distribution of two sets of unidimensional points. It is based on the empirical distribution function (ECDF). Given a probe $\mathfrak{M}_i$, with L sentry particles $\{\mathfrak{M}_{i,1}, ..., \mathfrak{M}_{i,L}\}$ ordered according to their respective fitness values $\{f(\mathfrak{M}_{i,1}, \mathbf{Co}_i), ..., f(\mathfrak{M}_{i,L}, \mathbf{Co}_i)\}$ obtained in the optimization of image $\mathbf{Co}_i$, the empirical distribution function (ECDF) is defined as a set of cumulative probabilities $\{E_1, ..., E_L\}$:

$$E_j = \frac{n_j}{L} \tag{2.9}$$

where n_j is the number of fitness values less than $f(\mathfrak{M}_{i,j}, \mathbf{Co}_i)$.

Given two sets of fitness values obtained in the evaluation of a same probe in two distinct images $\{f(\mathfrak{M}_{i,1}, \mathbf{Co}_i), ..., f(\mathfrak{M}_{i,L}, \mathbf{Co}_i)\}$ and $\{f(\mathfrak{M}_{i,1}, \mathbf{Co}_{i+1}), ..., f(\mathfrak{M}_{i,L}, \mathbf{Co}_{i+1})\}$, the KS statistic gives the maximum distance between their ECDFs. The null hypothesis is that both sets of fitness values were drawn from the same distribution and it must be rejected if their KS statistic is above the critical value for a given confidence level (D_α). For sets with more than 12 elements, the critical value can be computed as follows (Wessel):

$$D_\alpha = c_\alpha \sqrt{\frac{n_1 + n_2}{n_1 n_2}} \tag{2.10}$$

where n_1 is the number of elements in the first vector, n_2 is the number of elements in the second vector and c_α is the coefficient for confidence level α (Table 2.2).

Table 2.2 Values of c_α for confidence levels (two-sided) (Wessel).

	Values					
Confidence level (α)	0.1	0.05	0.025	0.001	0.005	0.001
Coefficient (c_α)	1.22	1.36	1.48	1.63	1.73	1.95

2.4.2 A memory-based intelligent watermarking method using DPSO

The proposed method is depicted in Algorithm 1.

Before optimization, STM and LTM will be empty (lines 2 and 3). For this reason, the swarm will be initialized randomly (line 4). Then, for each cover image ($\mathbf{Co}_i$), an attempt to recall the STM/LTM memory will be performed (line 7). If the recall fails, optimization is triggered and after that, the LTM memory ($\mathfrak{M}$) is updated with the swarm obtained in the end of the optimization process ($\mathfrak{S}_s$, lines 9 and 10).

After the first optimization, the STM will contain a single probe obtained at the end of the optimization of an image. Then, after at least two optimizations, the LTM will contain several probes, obtained at the end of the optimization of different images (more likely, images with

Algorithm 1 Algorithmic description of the proposed method.

Inputs:
$\mathbf{CO} = \{\mathbf{Co}_1, ..., \mathbf{Co}_N\}$ – set of cover images.
D_α – critical value for memory recall.

Definitions:
$\mathfrak{M}_S$ – Short Term Memory.
$\mathfrak{M}$ – Long Term Memory.
$\mathfrak{S}_s$ – set of solutions obtained in the optimization of $\mathbf{Co}_i$.
θ – recalled solution.
$Recall(\mathbf{Co}_i, D_\alpha)$ – recall STM/LTM memory (Algorithm 2).
$Update(\mathbf{Co}_i, \mathfrak{S}_s, \mathfrak{M})$ – update STM/LTM memory (Algorithm 3).

1: {Initialization}
2: $\mathfrak{M}_S \leftarrow \emptyset$
3: $\mathfrak{M} \leftarrow \emptyset$
4: Initialize swarm randomly (respecting bounds defined in Table 2.1).
5: {Computation}
6: **for** $i \in [1, N]$ **do**
7: $\quad \theta \leftarrow Recall(\mathbf{Co}_i, D_\alpha)$
8: $\quad$ **if** $\theta = \emptyset$ **then**
9: $\quad\quad$ Optimize $\mathbf{Co}_i$ using PSO and watermark it using best solution $\mathbf{p}_g$.
10: $\quad\quad$ $Update(\mathbf{Co}_i, \mathfrak{S}_s, \mathfrak{M})$
11: $\quad$ **end if**
12: **end for**

little similarity among them). In practice, in situations involving long sequences of images with similar structure, the STM should allow a faster recall than the LTM. In the same scenario, the LTM should provide means of recalling solutions for images that do not resemble the majority of the images in an homogeneous database. Moreover, it should provide means of adapting to new homogeneous sequences of images being fed into the system.

The memory recall is summarized in Algorithm 2.

During a STM recall, the probe ($\mathfrak{M}_S$) is re-evaluated for current image and a statistical test is employed to compare the similarity between both distributions of fitness values (line 3). If they are considered similar, the number of successful recalls of that probe ($Count_S$) is incremented (line 4) and the best solution is employed directly for current image, avoiding re-optimization (line 5). Otherwise, the LTM probes ($\{\mathfrak{M}_1, ..., \mathfrak{M}_L\}$) are sorted by their number of success-

Algorithm 2 Memory recall technique.

Inputs:
$\mathbf{Co}_i$ – cover image i.
D_α – critical value for KS-test.

Outputs:
θ – optimal solution.

Definitions:
$\mathfrak{M}_S$ – Short Term Memory (one probe).
$\mathfrak{M}$ – Long Term Memory (set of probes).
L – number of LTM probes.
$Count_i$ – number of successful recalls for probe i.
$f(\mathfrak{M}_j, \mathbf{Co}_i)$ – evaluate probe $\mathfrak{M}_j$ in image $\mathbf{Co}_i$.
$KS(\mathbf{A}, \mathbf{B})$ – Kolmogorov-Smirnov statistic between vectors **A** and **B**.

1: {Computation}
2: $\theta \leftarrow \emptyset$
3: /*STM Recall*/
4: **if** $KS(\mathfrak{M}_S, f(\mathfrak{M}_S, \mathbf{Co}_i)) \leq D_\alpha$ **then**
5: $Count_S \leftarrow Count_S + 1$
6: Set θ with best solution in $f(\mathfrak{M}_S, \mathbf{Co}_i)$.
7: **else**
8: /*LTM Recall*/
9: Sort $\mathfrak{M}$ by $Count$ (in reverse order).
10: **for** $j \in [1, L]$ **do**
11: **if** $KS(\mathfrak{M}_j, f(\mathfrak{M}_j, \mathbf{Co}_i)) \leq D_\alpha$ **then**
12: $Count_j \leftarrow Count_j + 1$
13: Set θ with best solution in $f(\mathfrak{M}_j, \mathbf{Co}_i)$.
14: Exit for.
15: **end if**
16: **end for**
17: $\mathfrak{M}_S \leftarrow max_{Count}(\mathfrak{M})$ /*Best probe is the first to be recalled and its best solutions are injected into the the swarm when re-optimization occurs.*/
18: **end if**

ful recalls ($Count_j$), in decreasing order (line 7) and the same procedure (fitness evaluation, followed by statistical test) is repeated for each probe until either a probe with similar fitness distribution is found or all probes have been tested (lines 9 – 13). After that, in both cases (successful or unsuccessful LTM recall), the probe with the highest number of successful recalls ($\max_{Count}(\mathfrak{M})$) is copied into the STM, replacing the previous one (line 15). If recall fails,

Algorithm 3 Memory update technique.

Inputs:
$\mathfrak{S}_s$ – set of solutions obtained in the optimization of $\mathbf{Co}_i$.
$\mathfrak{M}$ – Long Term Memory.

Definition:
$Count_L$ – success counter of new probe.

1: {Computation}
2: $Count_L \leftarrow 0$
3: Add $\mathfrak{S}_s$ to $\mathfrak{M}$.

the best STM solutions are injected into the swarm and re-optimization is triggered. There are two reasons for copying the LTM with highest number of successful recalls to the STM. The first is that for an homogeneous database, it should provide a better starting point in the event of re-optimization in comparison with re-randomization of the whole swarm. The second is that as the images in such scenario are expected to have a very similar structure, it makes sense trying to recall a probe that has been successfully recalled several times first.

This is a direct memory scheme (Yang, 2005) since the global best and respective swarm solutions (probe) are kept in the memory. An unlimited memory is assumed at this moment, thus there is no need to delete any solution from the LTM. Each probe contains a set of solutions, their respective fitness values and the number of successful recalls for that probe.

The memory update is summarized in Algorithm 3. In the memory update mechanism, a new probe ($\mathfrak{S}_s$) comprising solutions obtained in the optimization of embedding parameters for an image ($\mathbf{Co}_i$), their respective fitness values and a successful recalls counter (initialized at 0, line 2) is added to the LTM (line 3).

2.5 Experimental results

2.5.1 Methodology

In the following experiments, the swarm contains 20 particles, according to values found in the literature (Poli *et al.*, June 2007). Optimization stops whenever no improvement in global

best fitness occurs during 20 generations. This is considered quite conservative as literature suggests values between 5 and 20 (Zielinski and Laur, 2007). In order to be able to compare the behavior of the proposed technique in different scenarios, full optimization will be applied to all images and the resulting optimal swarms will be employed in the experiments involving the memory-based technique. Since the fragile watermark must be destroyed in the case of tampering, Q_F will be fixed at 2. In such case, flipping a single pixel in one block changes the value of the embedded bit at that block. Full optimization will occur for every image in the experiments with isolated images using the PSO system and for every time a change is detected in the experiments using the memory-based DPSO system.

Three different experiments will be performed (A, B and C). In experiment A, the performance of the approach based on using full optimization (PSO) for each image is compared with that of a non-optimized set of parameters found in the literature (Muharemagic, 2004): $\{B = 61,\ \Delta Q = 8,\ W = 3,\ S = 0\}$. In experiment B, the performance of the proposed memory-based DPSO method is compared with that of the full optimization method. Finally, in experiment C, the memory-based approach is applied in a smaller dataset and then, the probes obtained in that dataset are provided as a sort of *a priori* knowledge in the optimization of a separate, larger dataset. In the memory-based experiments, the KS statistic, α was set to 0.05, which corresponds to a coefficient $c_\alpha = 1.36$ (Table 2.2) and a critical value (D_α) of 0.43.

The two watermarks to be embedded are the 26×36 BancTec logo (Figure 2.6a) as robust watermark and a 36×26 Université du Québec logo (Figure 2.6b) as fragile watermark.

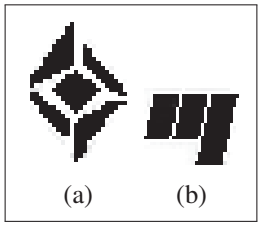


Figure 2.6 Bi-tonal logos used as watermarks. (a) 26×36 BancTec logo. (b) 36×26 Université du Québec logo.

Although most intelligent watermarking methods employ some kind of attack modeling (Vellasques *et al.*, 2010a) (i.e., apply an attack to watermarked images and measure the impact on robust watermark detection), the most simple scenario involves optimizing the parameters of the robust watermark against no attack. This is the approach to be followed in a first moment. Although it might seem trivial, it already requires choosing a set of parameters that makes the watermark robust enough to resist to the noise caused by the fragile watermark. Then, in a second moment, cropping attack (1% of image surface) will be employed. In both cases, the number of objective functions will be equal to two (which must be minimized) – visual distortion between cover and watermarked images (measured with the use of DRDM (Muharemagic, 2004)) and the inverse of the watermark detection rate (in this case, inverse Bit Correct Ratio or BCR^{-1}) – according to Equation 2.8. Since it was observed in the literature (Muharemagic, 2004) that absolute value of optimal DRDM is significantly smaller than that of the optimal BCR^{-1}, the DRDM will be scaled by a factor of 10^2 (which should put them in a same magnitude). Finally, an equal weight will be employed in the aggregation technique ($\omega_1 = \omega_2 = 0.5$) since an equal trade-off between robustness and imperceptibility is sought.

2.5.1.1 Database

The first database consists of a stream of 61 pages of issues 113(1) and 113(2) of the Computer Vision and Image Understanding (CVIU) Journal. The stream was divided in four blocks, where the first and third contain 15 pages of plain text, the second and fourth contain 15 and 16 pages of text and half-toned images, respectively. This is the Text/Image/Text/Image (TITI-61) database. Figure 2.7 shows some images from this database.

The second database contains 342 pages of 29 articles from CVIU 113(3) and 113(4) and will be named CVIU-113-3-4. These articles were converted to bi-tonal format with the use of ImageMagick [1] convert utility at 200 dpi. The resulting images have 1653×2206 pixels. These two databases were chosen mainly because the articles are publicly available in the Internet and other researchers can download the images and set the same database using the same protocol employed in this article. Moreover, the resulting image streams are considerably homogeneous

[1] http://www.imagemagick.org

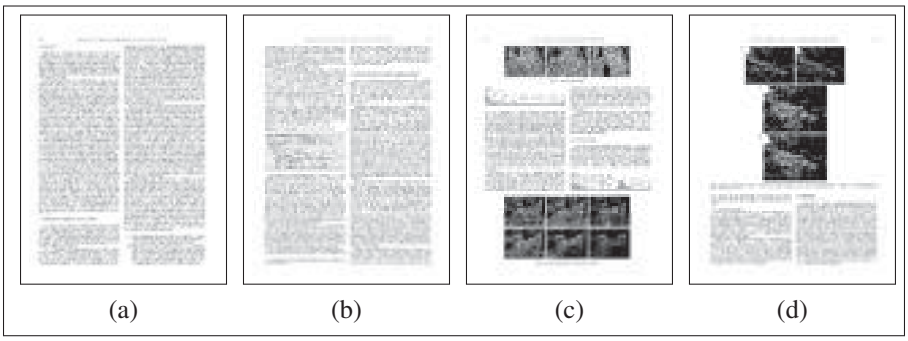


Figure 2.7 Database samples. Each image has a density of 200 dpi and 1653×2206 pixels. (a–b) Text. (c–d) Half-toned image.

and some of the samples contain color images. This allows employing the same protocol in the event of adapting the proposed method to the optimization of color and/or greyscale watermarking systems.

2.5.2 A – Optimization of isolated bi-tonal images using full PSO versus default embedding parameters

The main purpose of the experiments performed is to compare the performance of the static PSO method with that of default embedding parameters found in the literature. A comparison of the fitness values obtained by PSO-based system with that obtained by employing the default parameters suggested in (Muharemagic, 2004) shows that for most images, there was a decrease in fitness value. Figure 2.8a provides such comparison for the TITI-61 database, without the use of attacks. In this figure, $\Delta Fitness$ means the fitness obtained by the use of optimized parameters less the fitness obtained by the use of default parameters. The impact of optimization on robustness was negligible (Figure 2.8b). However, the use of optimization resulted in a significant improvement in the quality of the watermarked image (Figure 2.8d) with negligible impact on the fragile watermark (the corresponding BCR is $\geq 95\%$ for all cases as observed in Figure 2.8c).

But the main advantage of optimizing embedding parameters is when it comes to making the robust watermark resistant against an attack. Figure 2.9a shows $\Delta Fitness$ for the TITI-61 database, but with the use of cropping of 1%. In this particular case, such attack was employed during the optimization process (attack modeling). Regarding the default embedding param-

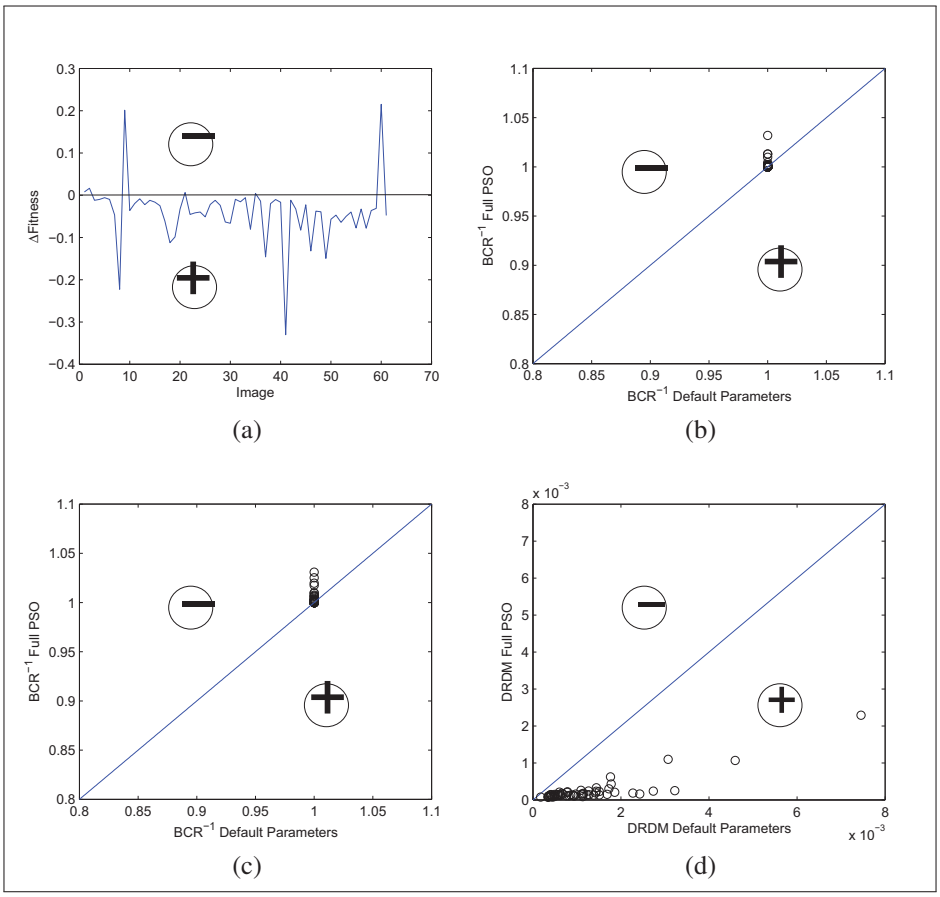


Figure 2.8 Comparison of performance between optimized and non-optimized embedding parameters (TITI-61, without attack). The region bellow the diagonal line ('+') represents an improvement in performance by the PSO-based method. (a) Difference between fitness values. (b) BCR^{-1} robust watermark. (c) BCR^{-1} fragile watermark. (d) $DRDM$.

eters, they are the same as employed in Figure 2.8. It is worth of notice that the optimized watermark is both more robust and less intrusive than the non-optimized robust watermark (Figures 2.9b and 2.9d) with little impact on the fragile watermark (the corresponding BCR is $\geq 90\%$ for most cases as observed in Figure 2.9c).

Figure 2.10 shows in details the difference in terms robustness between the non-optimized and the optimized watermark. The cover image (Figure 2.10a) is watermarked and then has 1% of its border cropped (Figure 2.10b). A zoomed in view of a portion of the optimized watermarked image shows that indeed the impact on quality is minimal (Figure 2.10c). For the non-optimized set of embedding parameters, both the robust and fragile watermarks were

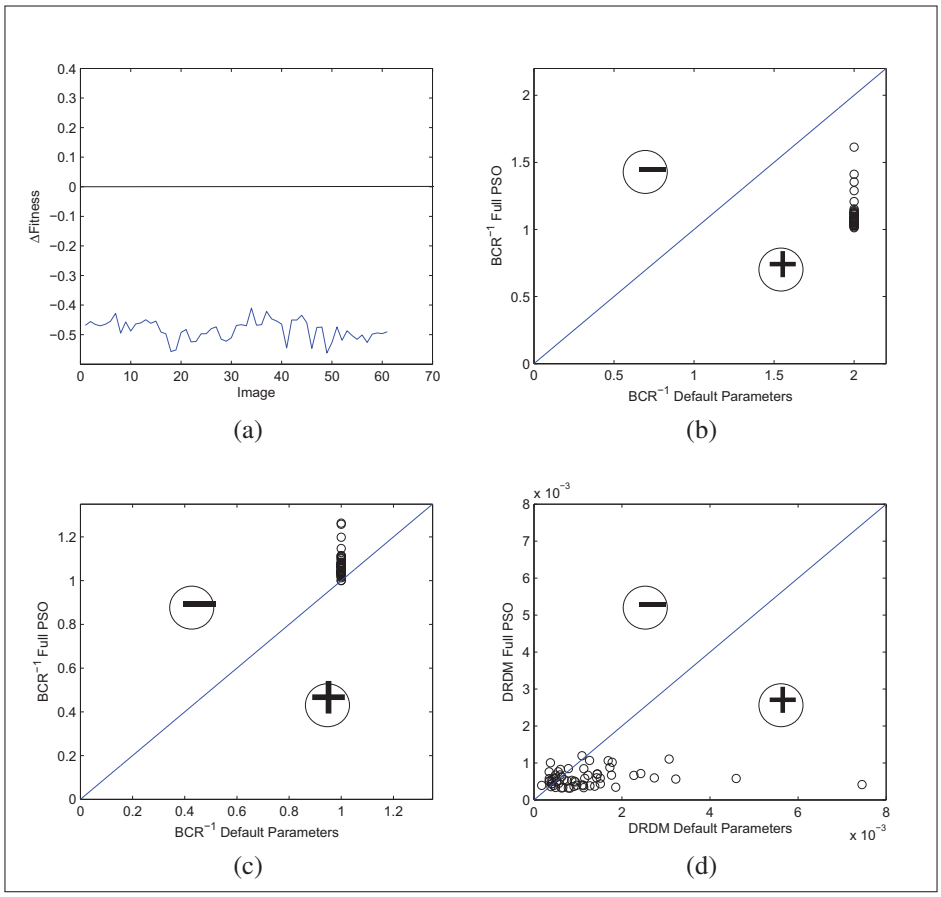


Figure 2.9 Comparison of performance between optimized and non-optimized embedding parameters (TITI-61, cropping of 1%). The region bellow the diagonal line ('+') represents an improvement in performance by the PSO-based method. (a) Difference between fitness values. (b) BCR^{-1} robust watermark (after attack). (c) BCR^{-1} fragile watermark (before attack). (d) $DRDM$.

completely removed (Figures 2.10d and 2.10e). However, for the optimized set of embedding parameters, the robust watermark resisted the attack (Figure 2.10f) while the fragile watermark was completely removed (Figure 2.10g). In this particular case, the set of optimal embedding parameters was $\{B = 9, \Delta Q = 16, W = 3, S = 4\}$ (it was $\{B = 28, \Delta Q = 4, W = 3, S = 11\}$ for the no attack modeling case).

This is the advantage of intelligent watermarking, it allows the optimization of embedding parameters for a specific attack (or set of attacks).

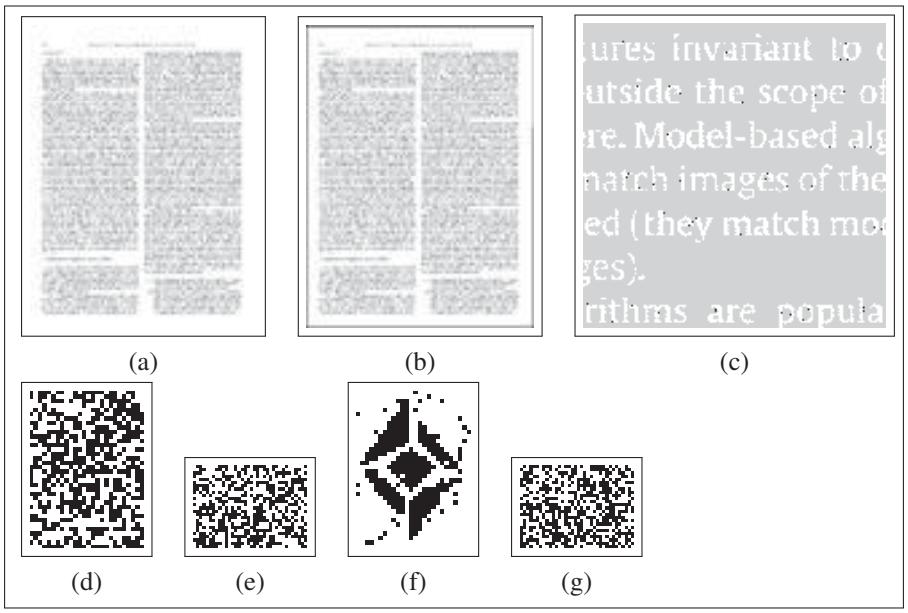


Figure 2.10 Effect of optimizing embedding parameters on quality. (a) Cover image. (b) Cropped watermarked image. (c) Difference between optimized watermarked (against cropping of 1%) and original images. (d) Detected non-optimized robust watermark. (e) Detected non-optimized fragile watermark. (f) Detected optimized robust watermark. (g) Detected optimized fragile watermark.

2.5.3 B – Optimization of streams of bi-tonal images using memory-based DPSO versus full PSO

The performance of the proposed method is compared with that of full optimization, in order to have a better understanding of the memory recall scheme (which is one of the main contributions of our method).

2.5.3.1 No attack

Figure 2.11 shows the difference in fitness performance between the proposed method and full optimization ($\Delta Fitness$) for the TITI-61 database, without the use of any attack. It can be observed that this difference is negligible. It required 2760 fitness evaluations to optimize all 61 images with the proposed method against 51460 fitness evaluations with full optimization (a gain of 94.6%). The Mean Squared Error (MSE) between the fitness values obtained by full optimization and by the proposed method is 5.4×10^{-6}. Full optimization was employed for

image 1, resulting in a first probe which was put in the STM/LTM. Probe 1 was recalled from STM for images 2–7, 9–33, 35, 36, 38–40, 42, 43, 45–61. Re-optimization has occurred for image 8, resulting in probe 2, which was put into the LTM. Probe 2 was recalled from LTM for images 34, 37, 41 and 44.

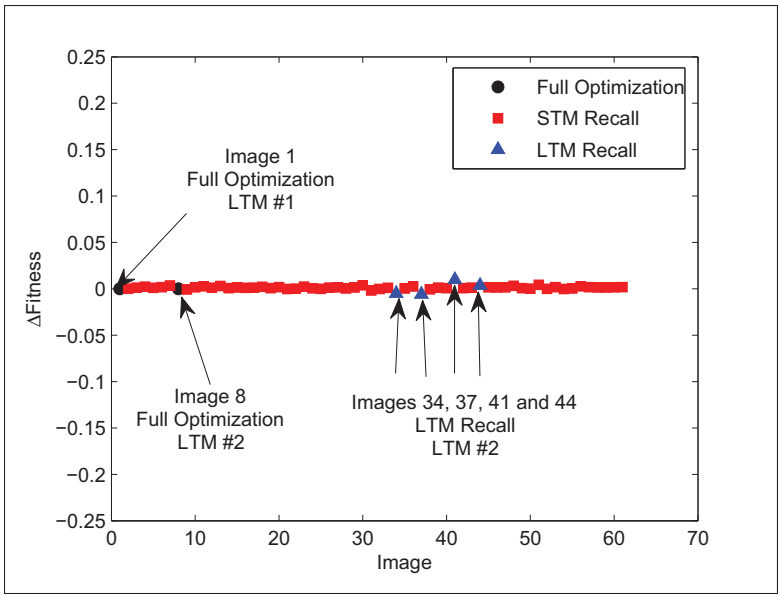


Figure 2.11 Fitness performance of proposed IW algorithm for the 61 images of the TITI-61 database (without attack).

Regarding the two metrics that compose the fitness, the main variations were due to quality (DRDM). However, as it can be observed in Figure 2.12, it was still quite similar to that of full optimization (MSE of 4.2×10^{-9}).

An interesting property of the memory scheme is that a probe also provides alternative solutions (other than the global best) during a recall. This can be observed in the histogram of recall of probe solutions (Figure 2.13). It is possible to observe that for probe 1, the global best resulted in the best fitness 13 times while other probe solutions – 5, 6, 11 and 15 – resulted in the best fitness 24, 1, 7 and 10 times, respectively. For the other probes, the global best was recalled three times while another solution (14) was recalled once. What is worth of notice is that all STM recalls (probe 1) for images from the Text category had either solutions 1, 5 or 11 as the best one (being the number of recalls 12, 4 and 7 respectively). And all STM recalls for images

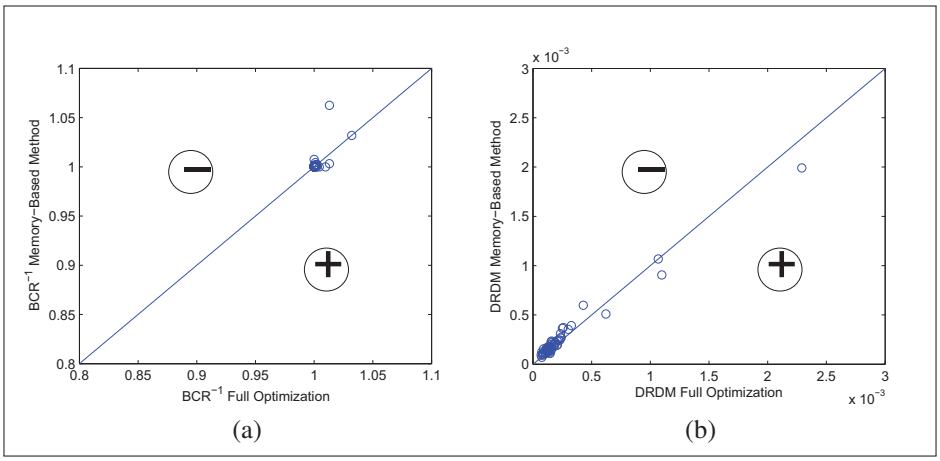


Figure 2.12 Comparison of watermarking performance between Full PSO and proposed method (TITI-61 database, without attack). The region bellow the diagonal line ('+') represents an improvement in performance by the memory-based method. (a) BCR^{-1}. (b) $DRDM$.

from the Image/Text category had either solutions 4 or 15 as the best one (with 20 and 10 recalls each, respectively). Thus, the same probe provided specialized solutions for different classes of images.

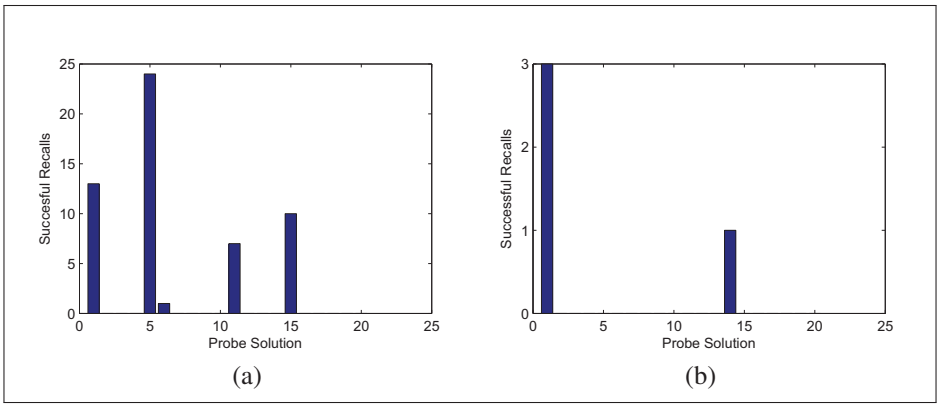


Figure 2.13 Histogram of recall of probes 1 and 2 solutions (TITI-61 database, no attack). (a) Number of recalls of probe 1 solutions. (b) Number of recalls of probe 2 solutions.

Another important observation is that all the STM recalls were made from the probe created by optimizing image 1 (which contains plain text and can be seen in Figure 2.7a). Then, probe 2 was created by optimizing the parameters for text image but which contains a significant

blank space (Figure 2.14a) and was recalled for other images with a significant amount of blank spaces (Figures 2.14b–e). Thus, the main benefit of the proposed long term memory mechanism is to provide ready-to-use solutions for images with similar embedding capacity (mainly in cases involving images that are considerably different from the majority of the stream).

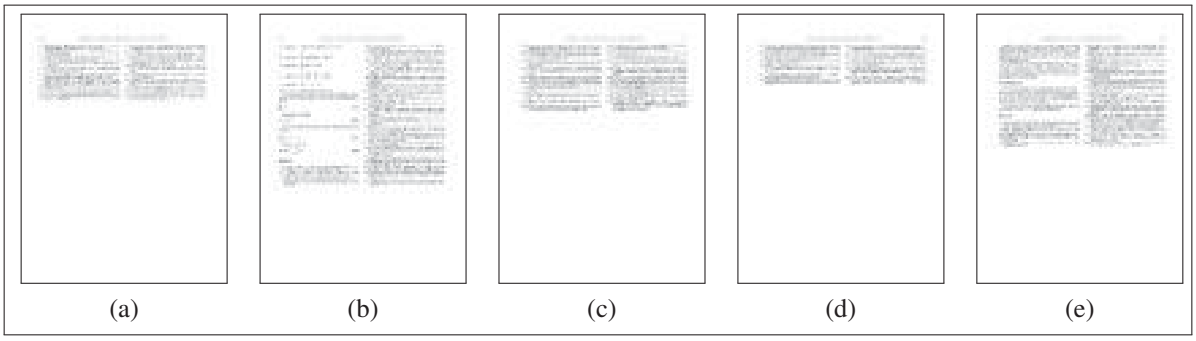


Figure 2.14 Images that resulted in either re-optimization or LTM recall (probe 2). (a) Image 8. (b) Image 34. (c) Image 37. (d) Image 41. (e) Image 44.

The behavior of the probes is analyzed more carefully for three specific cases. The first is a case of a successful STM recall (Figure 2.15). Here, probe 1, which is based on image 1 is re-evaluated on image 2. It is possible to observe that both cumulative distributions are very similar, thus this is a change of type II and re-optimization was not considered necessary. Another interesting observation is that both cumulative distributions of fitness cover a significant range of fitness values, which allows comparing the similarity of both fitness landscapes more precisely than using isolated solutions. What is worth of notice in this case is that the best solution for image 2 was considered sub-optimal in image 1. That is, the probe provided an alternate solution in this case.

Figure 2.16 shows a case of unsuccessful STM recall (Figure 2.16a) followed by a successful LTM recall (Figure 2.16b). In the first case, probe 1, which is based on image 1 is re-evaluated on image 37. Here, it is possible to observe that the distributions of fitness values in Figure 2.16a are considerably different, which corresponds to a change of type III. What is worth of notice here is that images 1 (Figure 2.7a) and 37 (Figure 2.14c) are also quite different. However, probe 2, which is based on image 8 resulted in a very similar cumulative distribution

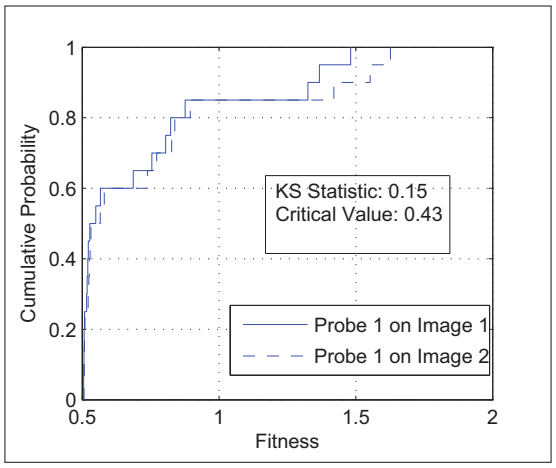


Figure 2.15 Case of successful recall. Cumulative distribution of probe 1 on images 1 and 2.

of fitness value when re-evaluated on image 37, which corresponds to a change of type II (both images have a significant amount of blank spaces as it can be observed on Figures 2.14a and 2.14c).

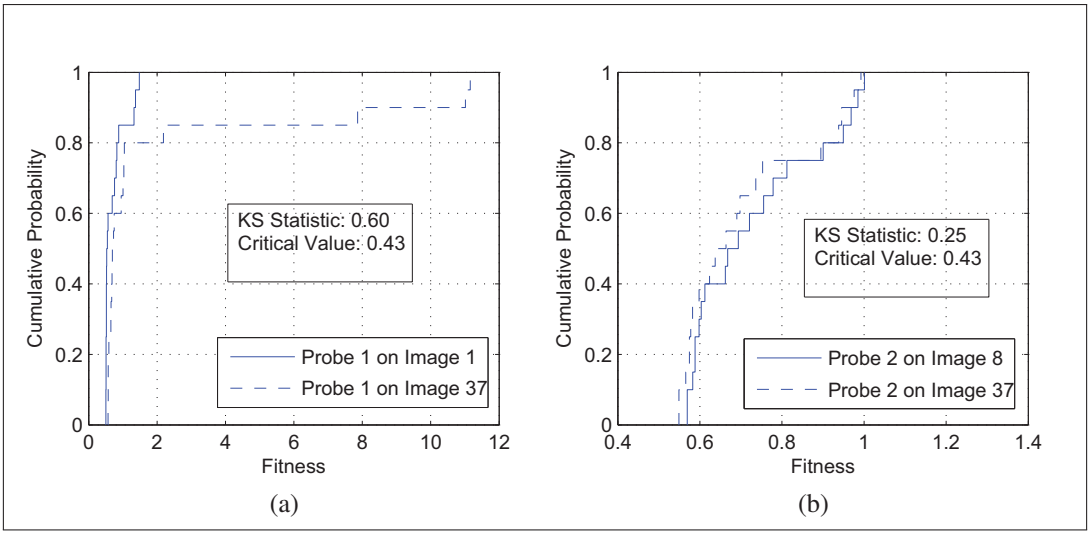


Figure 2.16 A case of unsuccessful STM recall followed by a successful LTM recall. (a) Cumulative distribution of probe 1 on images 1 and 37 (unsuccessful STM recall). (b) Cumulative distribution of probe 2 on images 8 and 37 (successful LTM recall).

The same experiment was performed in the CVIU-113-3-4 database. Regarding the fitness, the performance was quite the same (Figure 2.17).

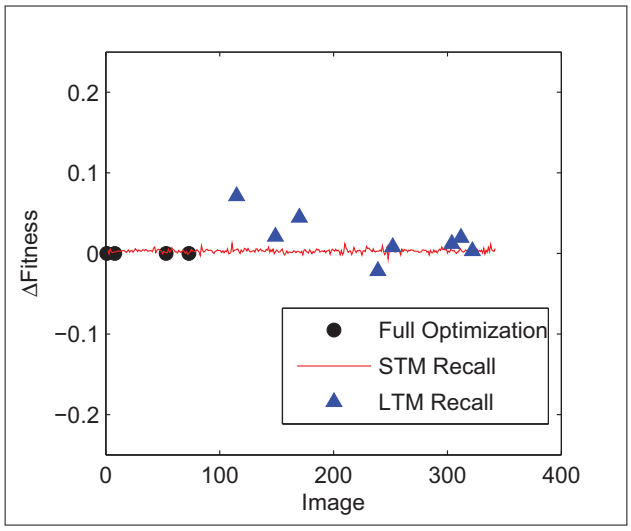


Figure 2.17 Fitness performance of proposed intelligent watermarking algorithm for the CVIU-113-3-4 database.

The Mean Squared Error (MSE) between the two sets of 342 fitness values obtained by full optimization and the proposed method is 3.9×10^{-5}. The decrease in fitness evaluations was more significant (96.4%, which corresponds to 10720 fitness evaluations in the proposed method against 301580 in full optimization).

It is possible to observe that the behavior of the metrics that compose the fitness was very similar to what was observed for the TITI-61 database (Figure 2.18).

Regarding the distribution of recalls per probe, for probe 1, the global best solution resulted in the best fitness evaluation for 315 recalls while another solution (20) was the best for 15 recalls. The 3 recalls of probe 2 were distributed among solutions 1 (global best), 11 and 20. The 5 recalls of probe 3 had solution 1 (global best) as the best one.

2.5.3.2 Attack modeling – cropping of 1% of image surface

The same experiments were performed using attack modeling (cropping of 1% of watermarked image area). The difference between fitness values can be seen in Figure 2.19.

Full optimization occurred twice (images 1 and 8). Probe 1 was employed in all STM recalls. As in the no attack case, the probe provided alternative solutions in numerous recalls (Figure

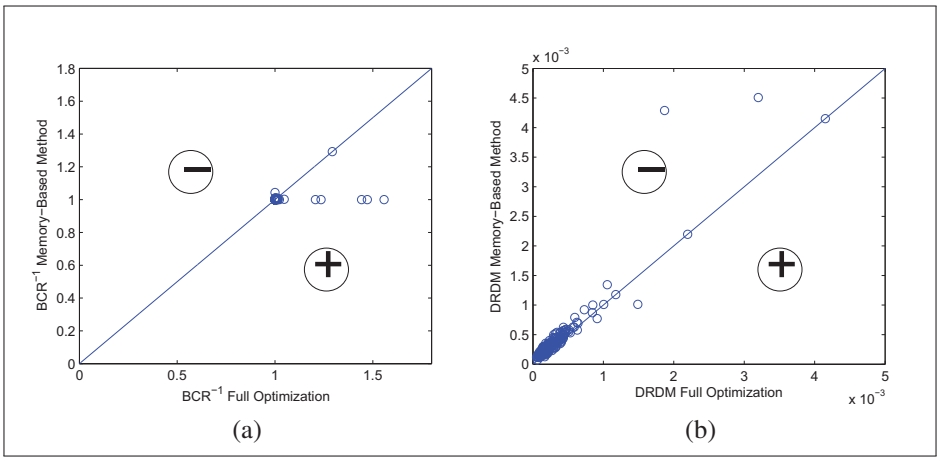


Figure 2.18 Comparison of watermarking performance between Full PSO and proposed method (CVIU-113-3-4 database, without attack). The region bellow the diagonal line ('+') represents an improvement in performance by the memory-based method. (a) BCR^{-1} (b) $DRDM$.

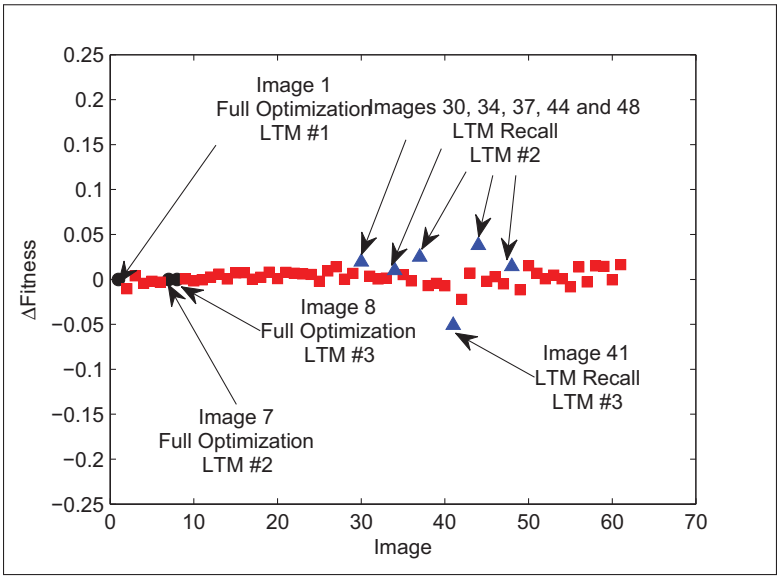


Figure 2.19 Fitness performance of proposed IW algorithm for the 61 images of the TITI-61 database with cropping attack.

2.20). For probe 2, solution 1 (global best) resulted in the best fitness five times while solution 2 was the best once.

It required 3960 fitness evaluations to optimize the 61 images against 55580 in full optimization mode (a gain of 92.9%). The MSE between both sets of fitness values was 1.4×10^{-4}. For the

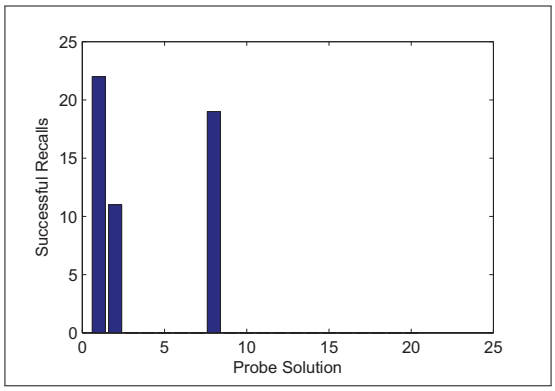


Figure 2.20 Histogram of recall of probe 1 solutions (TITI-61 database with cropping attack).

CVIU-113-3-4 database, although full optimization occurred twice, the gain in computational burden was a little bit higher (8740 fitness evaluations for the proposed method against 298100 for full optimization or 97.1%) for a MSE of 1.6×10^{-3}.

2.5.4 C – Optimization of streams of bi-tonal images using memory-based DPSO (learning mode) versus full PSO

In the first experiment involving learning, the probes obtained in the experiment with the TITI-61 database (no attack) were employed as a starting point for the CVIU-113-3-4 database (learning mode). Re-optimization was triggered twice and for this reason the number of fitness evaluations did not drop significantly when compared with the no-learning case. It required 10560 fitness evaluations to optimize all the 342 images (a gain of 96.5% when compared with full optimization). The MSE was considerably smaller (2.3×10^{-5}) than without learning.

In the second one, solutions from TITI-61 (cropping of 1%) were employed as a starting point for the CVIU-113-3-4 database resulted in a slight improve in computational burden performance (a gain of 97.2% when compared with full optimization) as no full optimization was required. There was also a significant gain in precision (MSE of 2.1×10^{-4}).

Finally, to illustrate the case-based learning capability of the proposed method, the images in the TITI-61 database (no attack) had their order shuffled and the same experiment was repeated

for the same database, but using the memory from previous experiment as a starting point. This resulted in zero full optimization and exactly the same MSE.

2.5.5 Discussion

It was observed through the experiments involving the full PSO version of the proposed method and the default parameters that the optimization of embedding parameters is justified, mainly in situations involving adapting these parameters to a certain type of attack. In these experiments, it can be said that the trade-off between the watermark robustness and image quality are tailored to the specific need of a given scenario. That is, in situations where the only "attack" to be expected is the embedding of a second (fragile) watermark, the use of optimization resulted in an increase in the quality of the watermarked image when compared to default parameters. In situations involving an intentional attack, it was possible to obtain a watermark that is at the same time more robust and less intrusive. In both cases, it resulted in little or no impact in the robustness of the fragile watermark.

In the experiments involving the memory-based approach, it was possible to observe that the proposed technique allowed a watermarking performance comparable to that of full optimization but for a fraction of the computational burden. Moreover, the memory provides a preliminary knowledge about a given intelligent watermarking task (learning capability).

The results are summarized in Table 2.3.

2.6 Conclusion

Since digital watermarking involves a trade-off between watermark robustness and image fidelity, numerous research papers have proposed the use of EC in order to find a setting of embedding parameters that result in an optimal trade-off for each specific image. However, this is a very costly process for high data rate applications as existing intelligent watermarking techniques rely on full optimization of embedding parameters for each image. This limits such approach to small proof-of-concept applications. In this chapter, fast intelligent watermarking

Table 2.3 Simulation results. Decrease of fitness evaluations is computed as $1 - (F_{Evals,M}/F_{Evals,F})$ where $F_{Evals,M}$ and $F_{Evals,F}$ are respectively, the number of fitness evaluations for the proposed approach and full optimization.

Attack	**Database**	**Learning**	**MSE Full PSO vs. DPSO Fitness**	**Decrease in fitness evaluations**
No attack	TITI-61	No	5.4×10^{-6}	94.6%
No attack	CVIU-113-3-4	No	3.9×10^{-5}	96.4%
No attack	CVIU-113-3-4	Yes	2.3×10^{-5}	96.5%
Cropping 1%	TITI-61	No	1.4×10^{-4}	92.9%
Cropping 1%	CVIU-113-3-4	No	1.6×10^{-3}	97.1
Cropping 1%	CVIU-113-3-4	Yes	2.1×10^{-4}	97.2%

of streams of document images is formulated as a dynamic optimization problem and a novel intelligent watermarking technique based on Dynamic Particle Swarm Optimization (DPSO) is proposed. With this technique, solutions (i.e., embedding parameters) from previous optimizations are archived and re-considered prior to triggering new optimizations. In such case, costly optimization were replaced by recall of previously computed solutions stored in memory. A practical application of the proposed technique would be the intelligent watermarking of massive amounts (e.g. tens of thousands per day) of different classes of documents like bank cheques, invoices and so on. The experimental results indicate that as long as different classes of images result in significant variation in the inherent fitness landscape of those images, the proposed technique should cope with those changes by triggering re-optimization. Moreover, in case of cyclical change, the memory should avoid costly re-optimization operations.

The proposed approach based on dynamic optimization is compared to the standard approach found in the literature which consists of applying full optimization to each image. To our knowledge, there is no approach based on dynamic optimization in the literature in order to make a comparison with the approach proposed in this chapter. In general, the accuracy of the memory-based method is similar to that of a method based on full optimization but for a fraction of the computational cost. In situations involving homogeneous databases of document images, the use of the proposed memory-based DPSO resulted in gains of up to 97.2% in computational burden. The main reason is that for transitions involving images already optimized

(which corresponds to a change of type II), the proposed method allowed recalling solutions from an archive directly, without need of re-optimization. It was also observed that the proposed adaptive change detection mechanism is robust enough to cope with minor variations in the fitness landscape between images with a similar structure (type II change) but is not discriminant enough to detect changes in the landscape between images with different structure (type III change). In addition, the proposed memory scheme provides a case-based reasoning capability to intelligent watermarking. A library of probes is incrementally built, which allows replacing costly full optimization operations by memory recalls. Such approach could be further improved by using statistical learning. This approach (as other intelligent watermarking approaches) assumes that a secure channel is available in order to make the optimal embedding parameters known at the detector.

In a future work, the performance of the proposed method will be analyzed in a more heterogeneous database. The performance is expected to be similar in a larger homogeneous database. However, an heterogeneous database should pose an additional challenge to the proposed technique. Since the main objective of this chapter was formulating intelligent watermarking of an homogeneous stream of images, only one type of attack was employed in the objective function (cropping of 1%). This attack was chosen because it was observed in proof of concept experiments that merely cropping 1% of image surface resulted in severe loss for the robust watermark when default embedding parameters found in the literature were employed. Adding other attacks should make the problem more heterogeneous and will be addressed in a future work. Having different types of attacks for different images in the database should also make the problem more heterogeneous and thus, more challenging. The use of machine learning in order to create a probe based on properties of the fitness landscape will also be addressed. Comparison with other DPSO approaches was not considered in this chapter because only one technique based on avoiding optimization for similar, cyclic problems was found in the literature (Kapp *et al.*, 2009) but this technique employs a change detection technique specific to the scenario addressed in that paper (pattern recognition). There are other promising EC techniques which could also be addressed in a future work like optimizing the heuristic parameters of PSO (Parsopoulos and Vrahatis, 2002) and using Genetic Programming (GP) in order to

develop a PSO algorithm tuned for a specific problem (Banks *et al.*, 2008). Finally, this framework should apply to different types of images and watermarking system since no property of the given system is employed during optimization other than robustness and quality (which are common to any watermarking system). The performance remains to be tested.

2.7 Discussion

In this chapter we demonstrate that the optimization of embedding parameters for homogeneous streams of document images can be formulated as a dynamic optimization problem. More specifically, we observe that in such scenario, a stream of document images will correspond to a stream of recurrent/cyclic problems.

We proposed a memory-based DPSO technique that allows decreasing the computational burden for such case by replacing costly re-optimization operations with recalls to a memory of ready-to-use solutions. We also proposed a technique that allows measuring the severity of changes in such problem stream (change detection).

It is important to recall that the main objective of this research is to find means of decreasing the computational cost of intelligent watermarking for streams of document images and a key element in tackling this issue is preserving a precise and compact representation of previously seen optimization problems in a memory. One of the limitation of storing static solutions in the memory is that these solutions tend to be biased to the problems for which they were obtained. Put differently, a memory of static solutions does not generalize well. This becomes an important issue when we are trying to deal with heterogeneous streams of document images. In such case, a biased memory will be too sensible to variations in the stream, leading to more unnecessary re-optimizations.

In the next chapter we investigate the use of Gaussian Mixture Models (GMMs) in order to devise a memory that generalizes better to variations in the problem stream. We also propose memory management mechanisms that allows this memory to adapt better to such variations.

CHAPTER 3

FAST INTELLIGENT WATERMARKING OF HETEROGENEOUS IMAGE STREAMS THROUGH MIXTURE MODELING OF PSO POPULATIONS

In this chapter we propose a Dynamic Particle Swarm Optimization (DPSO) technique which relies on a memory of Gaussian mixture models (GMMs) of solutions in the optimization space. This technique improves adaptability of the technique proposed in Chapter II for scenarios involving heterogeneous streams of document images. A compact density representation of previously-found DPSO solutions is created through GMM in the optimization space, and stored in memory. Solutions are re-sampled from this memory, re-evaluated for new images and have their distribution of fitness values compared with that stored in the memory. When the distributions are similar, memory solutions are employed in a straightforward manner, avoiding costly re-optimization operations. A specialized memory management mechanism allows to maintain and adapt GMM distributions over time, as the image stream changes. This memory of GMMs allows an accurate representation of the topology of a stream of optimization problems. Consequently, new cases of optimization can be matched against previous cases more precisely (when compared with a memory of static solutions), leading to considerable decrease in computational burden. Simulation results on heterogeneous streams of images indicate that compared to full re-optimization for each document image, the proposed approach allows to decrease the computational requirement linked to EC by up to 97.7% with little impact on the accuracy for detecting watermarks. Comparable results were obtained for homogeneous streams of document images. The content of this chapter was published at the Genetic and Evolutionary Computation Conference (GECCO) 2012 (Vellasques *et al.*, 2012b) and accepted for publication in Applied Soft Computing (Vellasques *et al.*, 2012a).

3.1 Introduction

Enforcing the security of digital images has become a critical issue over the last decade. Advances in communications and computing allow easy transmission and manipulation of digital images which limits the efficiency of traditional security methods like cryptography since when

the image has been decrypted there is no mean of enforcing its integrity and authenticity. Digital watermarking (Cox *et al.*, 2002) allows an additional level of security by embedding image related information in a covert manner through a manipulation of pixel values. The embedding process is subject to a trade-off between the robustness against intentional and unintentional image processing operations (attacks) and the imperceptibility of the embedded watermark (image quality) (Cox *et al.*, 1996). The embedding of multiple watermarks with different levels of robustness (Wu and Liu, 2004) allows enforcing image authenticity and integrity at the same time, which is a crucial issue in applications involving document images.

The trade-off between robustness and quality can be adjusted through manipulation of embedding parameters. In intelligent watermarking (IW), Evolutionary Computing (EC) algorithms such as Genetic Algorithms (GA) (Holland, 1992), Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) are employed in order to automatically find the embedding parameters that result in an optimal trade-off for a given image (Vellasques *et al.*, 2010a). A population of candidate embedding parameters is evolved through time using a combination of robustness and quality metrics as objective function (Areef *et al.*, 2005; Arsalan *et al.*, 2010, 2012; Chen and Lin, 2007; Ji *et al.*, 2006; Khan and Mirza, 2007; Khan *et al.*, 2008; Kumsawat *et al.*, 2005; Shieh *et al.*, 2004; Shih and Wu, 2004; Pan *et al.*, 2004; Usman and Khan, 2010; Wei *et al.*, 2006; Wu and Shih, 2006). But this process is not feasible in a large scale scenario due to the high computational cost of EC (Chen and Lin, 2007).

In (Vellasques *et al.*, 2010b, 2011), the IW of **homogeneous** streams of bi-tonal document images was formulated as a special case of *dynamic* optimization problem (DOP[1]), where a stream of images corresponds to a stream of optimization problems (states) and some states may occur repeatedly (Yang and Yao, 2008). Then, selected solutions found at the end of optimization were stored in an archive and recalled for similar problems. One limitation with such approach is that it assumes an homogeneous stream of document images, which is not always the case with real world applications. Selected solutions do provide an accurate representation of such stream of optimization problems, which makes it unfit for applications involving **heterogeneous** streams of document images.

[1] In a DOP the optima change over time and might be followed by a period of stasis (Farina *et al.*, 2004).

In this chapter, a novel IW technique is proposed for the fast intelligent watermarking of **heterogeneous** streams of document images. A memory consisting of Gaussian Mixture Models (GMMs) of all solutions in the optimization space (optimization history) plus their respective global bests is incrementally built, and for every image, solutions are sampled from this memory and re-evaluated for the new image. If both distributions of fitness values are similar, memory solutions are employed directly. Otherwise, the respective optimization problem is considered to be novel and a costlier DPSO operation is performed. After that, the memory is updated with the GMM of the optimization history of the new problem. Such approach results in a more precise representation of the topology of the stream of optimization problems. For this reason, it allows better recalling previously seen problems and is preferred in a scenario involving heterogeneous streams of document images. The research problem addressed in this chapter is how to use knowledge of past optimization problems in order to obtain a precise representation of a stream of optimization problems. The hypothesis on which this approach is based is that through time, density estimates of solutions found during optimization provide a compact but yet precise representation of the optimization problems presented to the intelligent watermarking system up to that point. The two main research questions addressed in this chapter are (1) how to build a compact representation of a stream of optimization problems in an incremental manner and (2) how to employ such representation in order to detect new cases of optimization.

The idea of using density estimates of solutions in the optimization space is not new. Estimation of Density Algorithms (EDA) (Pelikan *et al.*, 2002) rely on iteratively estimating density of **genotypic** data of **high evaluating** solutions. Differently than in EDA, our approach relies on both, **genotypic** and **phenotypic** data of **all** solutions from the optimization history in order to build a more general representation of the optimization problem. Moreover, in our approach the model is employed in order to match new problems with previously seen problems and to provide ready-to-use solutions. The research presented in this chapter follows the research presented in previous chapter. However, in the previous research we formulated IW of homogeneous streams of document images as the optimization of a stream of recurring problems and proposed a DPSO technique based on a memory of static solution. It was observed that such

memory lacked precision to tackle IW of heterogeneous streams of document images which led to a degradation in computational burden of that approach in such scenario. In this chapter, we focused on obtaining a precise representation of the underlying optimization problems in order to allow a better match between new and previous cases of optimization. Memory precision is an important element in our initial formulation of intelligent watermarking and has been neglected in previous chapter. Therefore, this strategy of incrementally building a compact yet precise model of a stream of optimization problems is the main contribution of this research and is to the best of our knowledge, novel.

The proposed approach is evaluated in the optimization of the embedding parameters of a multi-level (robust/fragile) bi-tonal watermarking system (Wu and Liu, 2004; Muharemagic, 2004) for both heterogeneous and homogeneous image streams, with and without cropping and salt & pepper (which are removal attacks (Voloshynovskiy *et al.*, 2001)). The standard approach in the bi-tonal watermarking literature is to test watermark robustness against tampering attacks like cropping, manual removal/modification of connected components like characters (Awan *et al.*, 2006; Ho *et al.*, 2004b; Lu *et al.*, 2002; Muharemagic, 2004; Pan *et al.*, 2000; Wu and Liu, 2004; Yang and Kot, Dec. 2006). Other removal attacks like Stirmark (Petitcolas *et al.*, 1998), image enhancement, JPEG compression, noise filtering either require grey-scale images or knowledge about the features present in the bi-tonal image (Marchand-Maillet and Sharaiha, 2000) and were not considered in our research. Resistance against geometric attacks can be easily tackled with the use of reference marks (Wu and Liu, 2004) and is also outside the scope of this chapter. Experimental results demonstrate that the proposed approach has a good memorization capability but at the same time, is flexible enough to adapt to variations in the stream of optimization problems.

Our optimization problem formulation of intelligent watermarking is presented in Section 3.2. A brief literature review of related techniques is presented in Section 3.3. The new approach proposed in this chapter, based on Gaussian Mixture Modeling for density estimation of solutions in the optimization space, and on adaptive memory management mechanisms is described in Section 3.4. Finally, Section 3.5 provides simulation results and discussion.

3.2 Optimization problem formulation of intelligent watermarking

The problem addressed in this article is the optimization of embedding parameters of a bi-tonal watermarking system, aimed at a high throughput adaptive watermarking of heterogeneous streams of document images. In this formulation, a stream of images is seen as a stream of optimization problems. Two possible actions can occur when an image from that stream is to be watermarked: (1) an existing solution (set of embedding parameters) is recalled from the memory; (2) optimization is triggered in order to find a new solution. If optimization is triggered, a population (swarm) of candidate solutions (particles) is evolved through several generations using Dynamic PSO (DPSO). At each generation, each solution has its fitness evaluated in a given watermarking task. The fitness function of the proposed technique is depicted in Figure 3.1.

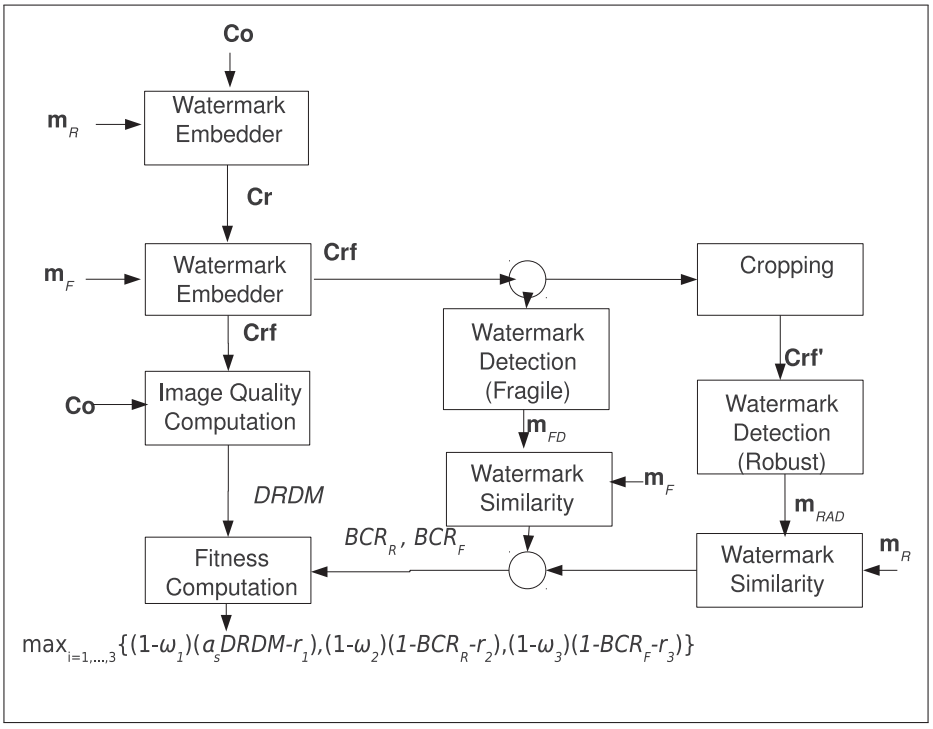


Figure 3.1 Fitness evaluation module.

The PSO algorithm employed on full optimization is the same described in (Vellasques *et al.*, 2011). The fitness function was slightly modified. Firstly, the Conventional Weighted Aggregation (CWA) mechanism was replaced by Chebyshev Weighted Aggregation which 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 fitnesses are considered good (Collette and Siarry, 2008). Secondly, the robustness of the fragile watermark was added to the aggregated function in order to minimize interference of the robust watermark as observed in (Vellasques *et al.*, 2011). Thirdly, BCR^{-1} was replaced by $1 - BCR$. Therefore, the fitness function will be defined as:

$$F(\mathbf{x}) = max_{i=1,..,3}\{(1-\omega_1)(\alpha_s DRDM - r_1), (1-\omega_2)(1-BCR_R - r_2), (1-\omega_3)(1-BCR_F - r_3)\} \tag{3.1}$$

where α_s is the scaling factor of the quality measurement $DRDM$ (Distance Reciprocal Distortion Measure (Lu *et al.*, 2004)), BCR_R (Bit Correct Ratio (Areef *et al.*, 2005; Pan *et al.*, 2004) between embedded and detected watermark) is the robustness measurement of the robust watermark, BCR_F is the robustness measurement of the fragile watermark, ω_i is the weight of the i^{th} objective with $\omega_i = \frac{1}{3}, \forall_i$, r_i is the reference point of objective i. The fitness function is depicted in Figure 3.1 where **Co** is the cover image, $\mathbf{m}_R$ and $\mathbf{m}_F$ are the robust and fragile watermarks, respectively, **Cr** is the robust watermarked image, **Crf** is the image that has been watermarked with both, the robust and the fragile watermarks (multi-level watermarked image), **Crf′** is the multi-level watermarked/attacked image, $\mathbf{m}_{RAD}$ is the robust watermark that has been detected from the multi-level watermarked/attacked image, $\mathbf{m}_{FD}$ is the fragile watermark that has been detected from the multi-level watermarked image.

The bi-tonal method of Wu and Liu (Wu and Liu, 2004) (relying on the pixel flippability analysis technique of Muharemagic (Muharemagic, 2004)) is employed as the baseline watermarking method in exactly the same manner as in (Vellasques *et al.*, 2011). This method allows the embedding of multiple watermarks in a same image with different levels of robustness where robustness is defined by a quantization step size parameter Q.

The particle encoding employed in this system can be seen in Table 3.1. Basically, the block size has lower bound of 2×2 and upper bound of $B_B \times B_B$ with $B_B = \max_B\{B^2 \times \max\{|\mathbf{m}_R|, |\mathbf{m}_F|\} \leq |\mathbf{Co}|\}$ pixels where B is the block width in pixels,

$|\mathbf{m}_R|$, $|\mathbf{m}_F|$ and $|\mathbf{Co}|$ is the size of the robust watermark, fragile watermark and cover images, respectively. The remaining bounds, ΔQ, SNDM (Structural Neighborhood Distortion Measure (Muharemagic, 2004)) window size and number of shuffling seeds were defined based on the literature (Muharemagic, 2004). Finally, $x_{i,j}$ is the j^{th} parameter encoded in the i^{th} particle.

Table 3.1 Range of embedding parameter values considered for PSO algorithm in this chapter.

Embedding Parameter	**Particle Encoding**
Block Size (B): $\{2, 3, 4, ..., B_B\}$	$x_{i,1} : \{1, 3, 4, ..., B_B - 1\}$
Difference between Q for the robust (Q_R) and fragile (Q_F) watermarks (ΔQ): $\{2, 4, 6, ..., 150\}$	$x_{i,2} : \{1, 2, .., 75\}$
SNDM window size (W): $\{3, 5, 7, 9\}$	$x_{i,3} : \{1, 2, 3, 4\}$
Shuffling seed index (S): $\{0, 1, 2, ..., 15\}$	$x_{i,4} : \{0, 1, 2, ..., 15\}$

3.3 Related work

3.3.1 Dynamic particle swarm optimization (DPSO) of recurrent problems

Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) relies on heuristics found on bird flocks and fish schooling in order to tackle the optimization of non-linear, noisy optimization problems. The underlying principle is that a population (swarm) of candidate solutions (particles) can tackle such type of optimization problem in a parallel manner with each particle performing its search guided by the best position found by itself and its best neighbor. The canonical PSO cannot tackle dynamic optimization when the optima changes due to issues like outdated memory, lack of a change detection mechanism and diversity loss (Blackwell, 2007; Carlisle and Dozier, 2002). One possible strategy to tackle this problem is to restart optimization whenever a change has been identified. However, the computational burden of such approach is prohibitive, specially in practical applications. But numerous practical applications, including intelligent watermarking of stream of document images, involve recurrent problems, that reappear through time, in a cyclical manner. It has been demonstrated in the literature that the best strategy to tackle such time of problem is to keep a memory of previous solutions to be recalled for future similar problems, in an approach named memory-based op-

timization (Yang and Yao, 2008). It has also been demonstrated that depending on the level of similarity between previous and new problems, it is possible to employ the solutions directly in the new problem, without any need of re-optimization (Vellasques *et al.*, 2011).

According to Yang and Yao (Yang and Yao, 2008), solutions can be stored in a memory either by an implicit or an explicit memory mechanism. In an implicit memory mechanism, redundant genotype representation (i.e. diploidy-based GA) is employed in order to preserve knowledge about the environment for future similar problems. In an explicit mechanism, precise representation of solutions is employed but an extra storage space is necessary to preserve these solutions for future similar problems. There are three major concerns in memory-based optimization systems that rely on an explicit mechanism: (1) what to store in the memory; (2) how to organize and update the memory; (3) how to retrieve solutions from the memory. Regarding what to store, there are two known approaches: direct memory scheme, where good solutions are stored and reused when the environment changes; associative memory scheme, where what is stored is information that associates good solutions with their environment (in most cases, a density estimate of the parameter space). The memory organization, by its way, can be based on a local mechanism (individual oriented) or on a global mechanism (population oriented). Regarding the memory update, since most real world applications assume limited memory, the basic approach is to select a solution stored in the memory to be removed (a review of removal strategies can be found in (Branke, 1999)) or updated by the newest solution.

An external memory requires an appropriate memory retrieval mechanism. There are two main memory retrieval strategies (Wang *et al.*, 2007) – memory-based resetting and memory-based immigrants. In the first strategy, when a change is detected (change detection is usually achieved by re-evaluating memory solutions on the new environment), all solutions in the memory are re-evaluated and the best one is chosen as the new global best solution if it is better than the old one. In the memory-based immigrants strategy, all the solutions in the memory are re-evaluated and injected into the population.

The approach proposed in this chapter is based on an associative memory. Since it has been already demonstrated in the literature that an associative memory allows associating previous

solutions with corresponding new cases of optimization, we evolve this idea a little further and employ the associative memory as a mean of modeling an stream of optimization problems. That is, more than associating solutions with new cases of optimization, the proposed approach allows classifying new cases of optimization based on previously learned problems.

3.3.2 Pattern classification

Pattern classification (Duda *et al.*, 2000) deals with assigning category labels to new patterns based on previously learned pattern/label assignments. Novelty detection (or one-class classification (Tax and Duin, 2004)) comprises the identification of patterns that were not available during a training (learning) phase. The main objective of a novelty detection system is to detect whether a new pattern is part of the data that the classifier was trained on or not (Markou and Singh, 2003a). A novelty detection system can be either off-line (Japkowicz *et al.*, 1995) (when the model is created once and not updated at all) or on-line (Ma and Perkins, 2003) (when the model is updated as new data arrives). In the proposed scenario, a cyclic DOP also requires detecting if a new problem corresponds to a previous (training) problem. And as in novelty detection, the complete representation of a problem is not available due to computational constraints. That is, a memory must provide means of storing and recalling optimization problem concepts in an incremental manner rather than simply associating stored solutions with new problems (as in the memory-based optimization approaches found in the literature).

Markou and Singh (Markou and Singh, 2003a) pointed the main issues related to novelty detection. Five of these issues are crucial in the envisioned scenario. The first is the principle of robustness and trade-off which means that the novelty detection approach must maximize the exclusion of novel patterns while minimizing the exclusion of known patterns. The second is the principle of parameter minimization which means that a novelty detection method must minimize the number of user-set parameters (mainly when we consider that in the envisioned application the data modeling technique must be closely integrated with the DPSO approach with minimal human intervention). The third is the principle of generalization which implies that the system should be able to generalize without confusing generalized information as novel. The fourth is the principle of adaptability which means that knowledge of novel sam-

ples must be integrated into the model. The fifth is the principle of computational complexity, which means that the computational complexity of a novelty detection should be as less as possible (also a very important issue in the given application, specially regarding detection, which should not be more expensive than re-optimizing).

It can be said that in the proposed application, the fourth and fifth principles are closely related. Retraining the model from scratch when novel optimization problem is detected would require storing all patterns (optimization history) seen so far, resulting in an ever increasing memory cost. Therefore, in the given scenario the model must be updated using only solutions from the new problem which can be seen as an incremental learning strategy. As defined by Jain *et al* (Jain *et al.*, 2006), in incremental learning, the learner has access only to a limited number of examples (patterns). In each step, an hypothesis can be built upon these examples and a former hypothesis in a way that (1) none of the intermediate hypotheses a learner explicates contradicts the data processed so far and (2) each intermediate hypothesis is maintained as long as it is consistent with the data seen. Gennari *et al* (Gennari *et al.*, 1989) studied the use of incremental learning in building hierarchical models of concepts (concept formation). They observed that initial non-representative data may lead a learning system astray. The use of GMM in such case is very common (Wu *et al.*, 2005; Yamanishi *et al.*, 2000) specially because it allows adaptability at a low computational cost when compared with other approaches such as neural networks (Markou and Singh, 2003b).

From a memory-based optimization point of view, a new concept must (1) represent novelty when compared with existing concepts; (2) provide a precise manner of probing the fitness landscape. The basic memory unit in the proposed approach is a **probe** and it contains a density estimate of solutions plus the global best solution, both created after the optimization of a single image. When a new probe is created after a round of optimization, it should only be inserted if there is no similar probe in the memory. Otherwise it should be merged with the most similar probe in order to enforce (1). That is, a good memory management mechanism should keep the dissimilarity between new probes and probes in the memory consistently high. Put differently, inserts should occur when a new probe provides new information about the

stream of optimization problems. Figure 3.2 illustrates the two possible scenarios concerning a memory update.

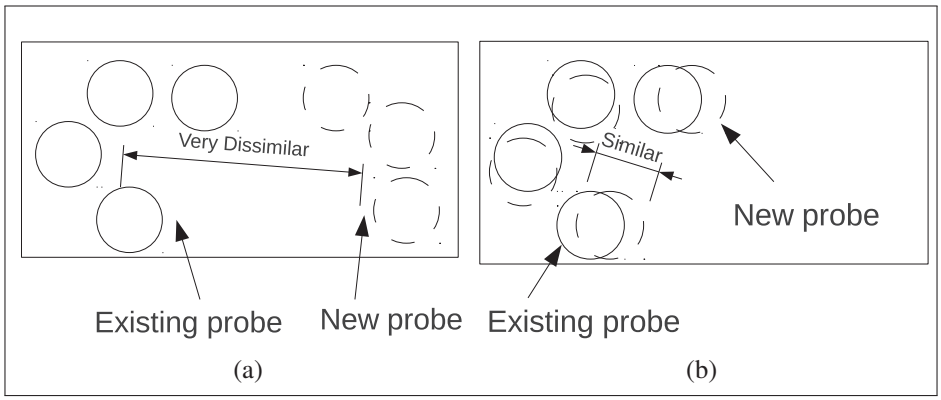


Figure 3.2 Two possible scenarios involving memory update (existing probe is represented by solid circle while new probe is represented by dashed circle). (a) New probe is not similar to existing probe (new concept). (b) New probe is similar to existing probe (existing concept).

By enforcing (1), memory redundancy is expected to be mitigated since the insert of new probes is constrained by a dissimilarity measure. In such case, memory elements are expected to resemble more Figure 3.2a than Figure 3.2b. That is, the memory is expected to be more diverse. This leads to a better usage of computational resources since the number of memory elements (probes) necessary to represent a given concept is minimized. Moreover, since the main objective of memory in the proposed system is to provide means of sampling the fitness landscape of unseen optimization problems, this increase in memory diversity should lead to an increased coverage of the sampled space (greater sampling diversity), enforcing (2). This means that during the optimization of a stream of images, as images are fed into the system, the amount of new information should decrease gradually as memorization takes place. Consequently the number of re-optimizations should gradually decrease after this memorization phase is complete. This allows for example, creating a memory on a laboratory environment (training mode) and then deliver this memory in a production environment.

3.4 Fast intelligent watermarking using Gaussian modeling of PSO populations

Figure 3.3 depicts a new memory-based IW system that integrates density estimation in order to minimize memory size. Given an image $\boldsymbol{Co}_i$ picked from a stream of $|\boldsymbol{Co}|$ images (see **1** in Figure 3.3), an attempt to recall the Short Term Memory (STM) – represented as $\mathfrak{M}_S$ and comprising a mixture model of solutions Θ_S obtained during the optimization of a single image $\boldsymbol{Co_S}$ and the global best solution for that image $\boldsymbol{p}_{g,S}$ – is performed first (see **2** in Figure 3.3). During a STM recall, a **set of solutions** (defined as $\boldsymbol{X}_{S,S}$) and their respective **fitness values** are sampled from Θ_S (including the global best, $\boldsymbol{p}_{g,S}$ stored in the STM). It is important to note that apart from $\boldsymbol{p}_{g,S}$, the position ($\boldsymbol{X}_{S,S}$) and fitness values ($\boldsymbol{F}(\boldsymbol{X}_{S,S}, \boldsymbol{Co}_S)$) of sentry solutions are an approximation of the positions and fitness values obtained during the optimization of $\boldsymbol{Co}_S$. The sentry solutions are re-evaluated for $\boldsymbol{Co}_i$ resulting in another set of fitness values $\boldsymbol{F}(\boldsymbol{X}_{S,S}, \boldsymbol{Co}_i)$. The Kolmogorov-Smirnov (KS) statistical test (NIST/SEMATECH, 2010) is employed in order to measure the similarity between the distribution of $\boldsymbol{F}(\boldsymbol{X}_{S,S}, \boldsymbol{Co}_S)$ and $\boldsymbol{F}(\boldsymbol{X}_{S,S}, \boldsymbol{Co}_i)$. If $KS(\boldsymbol{F}(\boldsymbol{X}_{S,S}, \boldsymbol{Co}_S), \boldsymbol{F}(\boldsymbol{X}_{S,S}, \boldsymbol{Co}_i))$ is smaller than a critical value D_α for a confidence level α, the watermarking parameters corresponding to the solution which resulted in the smallest $\boldsymbol{F}(\boldsymbol{X}_{S,S}, \boldsymbol{Co}_i)$ are employed right away for $\boldsymbol{Co}_i$, avoiding a costly optimization operation.

Otherwise (see **3** in Figure 3.3), the same process is repeated for each mixture model Θ_j and global best $\boldsymbol{p}_{g,j}$ in the Long Term Memory (LTM) – represented as $\mathfrak{M}$ and comprising $|\mathfrak{M}|$ mixture models of solutions ($\{\Theta_1, ..., \Theta_{|\mathfrak{M}|}\}$) obtained during the optimization of several different images and their respective global best solutions ($\{\boldsymbol{p}_{g,1}, ..., \boldsymbol{p}_{g,|\mathfrak{M}|}\}$) – being the LTM probes sorted in reverse order of their number of successful recalls.

If a LTM probe $\mathfrak{M}_j$ results in a successful recall, the watermarking parameters corresponding to the solution which resulted in the smallest fitness value in $\boldsymbol{Co}_i$ are employed right away for that image. If no probe in the LTM resulted in successful recall, the Dynamic PSO (DPSO) technique described in (Vellasques *et al.*, 2011) is employed in order to optimize the embedding parameters for $\boldsymbol{Co}_i$ (see **4** in Figure 3.3). A certain number of solutions re-sampled from the STM plus its respective global best are injected into the swarm, providing a starting point

for optimization. After that, in the memory update (see **5** in Figure 3.3), the optimization history (**position** and **fitness** of all solutions during all iterations) is employed in order to estimate a mixture model (Θ) of the fitness landscape. This mixture model plus the global best solution ($\boldsymbol{p}_g$) obtained during optimization will form a probe to be added to the STM replacing previous probe. This probe is also either merged or inserted into the LTM based on the similarity between its mixture model and the mixture models of LTM probes. In the case of an insert, an older probe might be deleted to give room for the new one if memory limit has been reached.

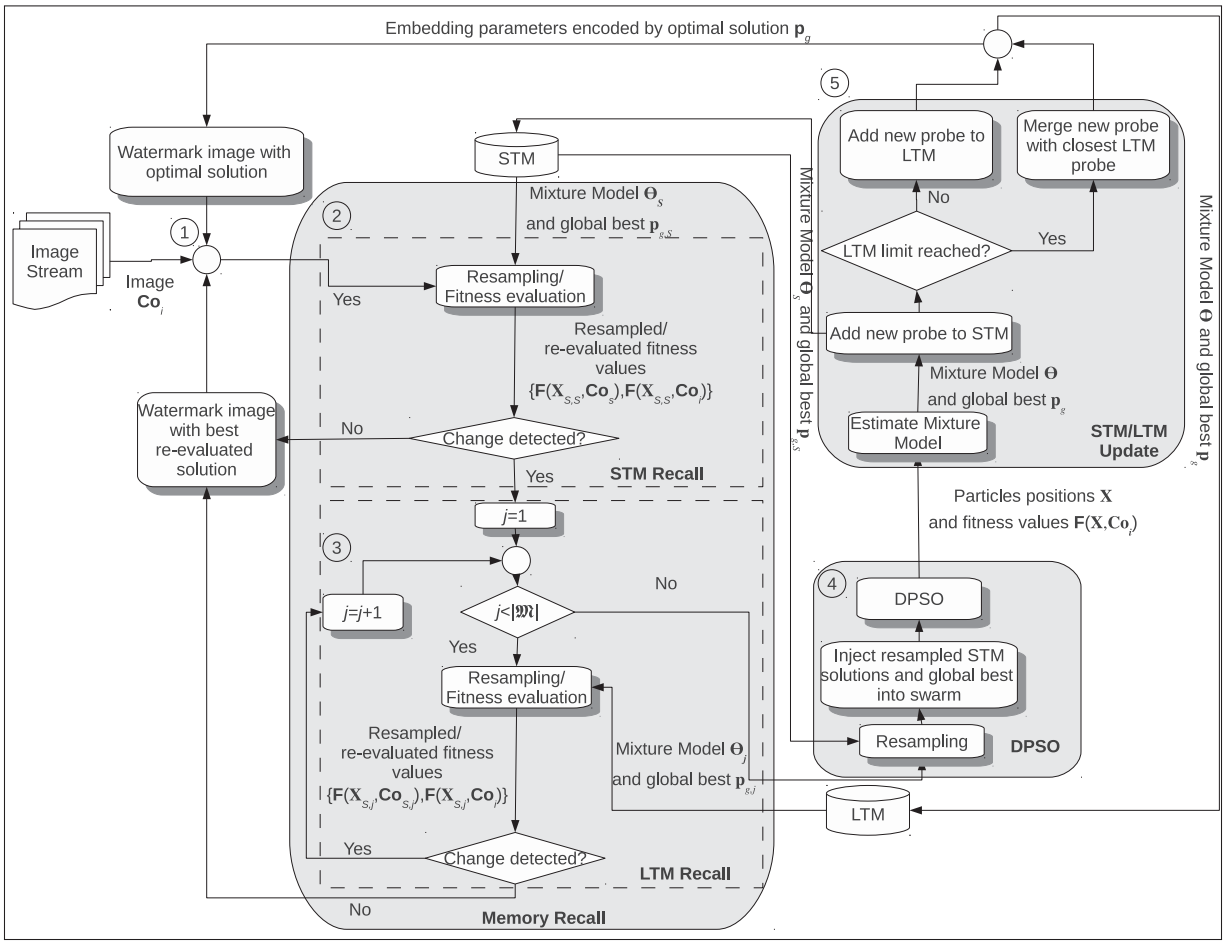


Figure 3.3 Flowchart diagram representing the proposed method for fast intelligent watermarking of heterogeneous bi-tonal image streams using Gaussian mixture modeling of PSO populations (anchor points are employed in order to guide the reader).

The first level of memory allows for a fast recall in situations where a block of similar images (e. g. pages of a same document) appears. The second level allows for recall of solutions

in situations where the fitness landscape associated with the image being watermarked is not similar to that of the last optimized image but still is similar to that of an image that had been processed before. Re-sampling of GMMs is expected to result in more diverse solutions which can cover a more significant region of the fitness landscape than would be possible with static solutions as the later tend to be concentrated in narrow regions of the fitness landscape (in the surroundings of previous optima). The rest of this section describes how the memory management approach addresses the three major concerns in memory-based optimization systems: (1) what to store in the memory; (2) how to organize and update the memory; (3) how to retrieve solutions from the memory. The memory update and retrieval algorithms are explained with details later in this section.

3.4.1 What to store?

In the proposed approach, a model of an optimization problem (which provides a more compact and precise representation than selected individual solutions) is estimated through unsupervised learning techniques (Jain *et al.*, 1999) based on the positions and fitness values of solutions in the optimization space. Because of the stream of optimization problems formulation of dynamic optimization, the distribution of these solutions is expected to be multi-modal. In such case, a finite mixture model is a powerful tool for estimating the distribution of these solutions. A mixture model consists of a linear combination of a limited (finite) number of models

$$p(\boldsymbol{x}|\Theta) = \sum_{j=1}^{K} \alpha_j p(\boldsymbol{x}|\theta_j) \tag{3.2}$$

where $p(\boldsymbol{x}|\Theta)$ is the probability density function (pdf) of a continuous random vector $\boldsymbol{x}$ given a mixture model Θ, K is the number of mixtures, α_j and θ_j are the mixing weights and parameters of the j^{th} model (with $0 < \alpha_j \leq 1$ and $\sum_{j=1}^{K} \alpha_j = 1$). The mixture model parameters $\Theta = \{(\alpha_1, \theta_1), ..., (\alpha_K, \theta_K)\}$ are estimated using observed training data. The common approach is to employ a Gaussian distribution to represent each element ($\theta_j = \{\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}$) where $\boldsymbol{\mu}_j$ is the mean vector and $\boldsymbol{\Sigma}_j$ is the covariance matrix. A mixture containing Gaussian elements is known as a Gaussian Mixture Model (GMM).

The approach proposed in this chapter builds a mixture model comprising both, the **parameter** and **fitness** space. Since it was observed that local best data results in density estimates that are over-fit to a specific problem, the approach employs current particle position instead of local best data. We propose employing particle positions and fitness values rather than local best positions and fitness values in order to estimate the model as they provide a more general model of a given optimization problem. Every time re-optimization is triggered, historical particle position data (all generations of an optimization task) will be employed as a training dataset. Since the problem itself is dynamic, during an update, the LTM needs to adapt to new changes in the data but as well be capable of "forgetting" or pruning unnecessary information.

3.4.2 How to organize and update?

In the proposed memory scheme there are two levels of update – STM and LTM. After re-optimization, position and fitness data of all particles for all iterations is employed in order to estimate a mixture model Θ (Eq. 3.2) of the fitness landscape. This model plus the global best will comprise a new probe to be added to the STM and LTM. The standard approach in the literature to estimate mixture parameters is to employ Expectation-Maximization (EM). In EM, Θ is estimated by gradually applying the E-step followed by the M-step until convergence is met. Convergence is attained when the log likelihood has stabilized over some dataset. A limitation regarding the use of standard EM in practical applications is the initialization of mixture components (Figueiredo and Jain, 2000). The main problem is that EM is unable to move components across low likelihood regions. EM is also unable to escape from situations where two or more components are similar, sharing the same data points. Another limitation is defining the appropriate number of components in a mixture. Usually when there are much more components than the necessary and the covariance matrices are unconstrained, some of the α_j's may approach zero and the corresponding covariance matrix may become arbitrarily close to singular.

Figueiredo and Jain (Figueiredo and Jain, 2000) initialize the mixture with a large number of components, where each component is centered at a randomly picked data point. As the parameters are updated (1) components lacking enough data points to estimate their covariance

matrices have their corresponding α's set to zero (component annihilation); (2) the number of components is gradually decreased until a lower boundary is achieved and then, the number that resulted in the best performance is chosen. They also proposed the following (log-likelihood) convergence criterion based on the Minimum Message Length (MML) which avoids local minima when two or more components are similar:

$$\mathfrak{L}(\Theta, \boldsymbol{x}) = \frac{N}{2} \sum_{\alpha_j > 0} \log(\frac{n\alpha_j}{12}) + \frac{k_{nz}}{2} \log \frac{n}{12} + \frac{k_{nz}(N+1)}{2} - \log p(\boldsymbol{x}|\Theta) \tag{3.3}$$

where k_{nz} is the number of components with $\alpha_j > 0$, n is the number of data points and N is the number of parameters (variables) in a given mixture (which is a function of d, the number of dimensions of X):

$$N = d + d(d+1)/2 \tag{3.4}$$

Then, the E-step and M-step are applied iteratively. In the E-step, the posterior probability is computed (Blekas and Lagaris, 2007):

$$w_{ij}^{(t)} = \frac{\alpha_j p(\boldsymbol{x}_i|\theta_j)}{\sum_{k=1}^{K} \alpha_k p(\boldsymbol{x}_i|\theta_k)} \tag{3.5}$$

In the M-step the model parameters are updated. The following α update annihilates components lacking enough data points:

$$\alpha_j^{(t+1)} = \frac{max\{0, (\sum_{i=1}^{n} w_{i,j}) - \frac{N}{2}\}}{\sum_{k=1}^{K} max\{0, (\sum_{i=1}^{n} w_{i,k}) - \frac{N}{2}\}} \tag{3.6}$$

The remaining mixture parameters are updated as:

$$\boldsymbol{\mu}_j^{(t+1)} = \frac{\sum_{i=1}^{n} w_{i,j}^{(t)} \boldsymbol{x}_i}{w_{i,j}^{(t)}} \tag{3.7}$$

$$\boldsymbol{\Sigma}_j^{(t+1)} = \frac{\sum_{i=1}^{n} w_{i,j}^{(t)} (\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(t+1)})(\boldsymbol{x}_i - \boldsymbol{\mu}_j^{(t+1)})^T}{w_{i,j}^{(t)}} \tag{3.8}$$

where d is the number of dimensions of $\boldsymbol{x}$.

3.4.2.1 Memory management operators – insert, merge and delete

In the given scenario, a memory update mechanism must address two fundamental issues of memory management. The first is what to do when a new probe is created. More specifically in which conditions should a new probe be merged with an existing probe and in which conditions should it be plainly inserted? The second is, in such situation, what to do when the memory is full? Should the new probe be merged with an existing probe even though they are not similar? Should an existing probe be deleted to make room for the new probe?

In order to mitigate these issues, we propose a selective memory update mechanism. In this mechanism, when the memory is due to be updated with a new probe, the $C2$ distance metric (Sfikas *et al.*, 2005) (which provides a good trade-off between computational burden and precision) will determine if the new probe will be either added to the LTM (insert operation) or merged with an existing probe. The distance between two mixtures Θ and Θ' (or $C2(\Theta, \Theta')$) is defined as:

$$\Phi_{i,j} = (\boldsymbol{\Sigma}_i^{-1} + \boldsymbol{\Sigma}_j^{\prime -1})^{-1} \tag{3.9}$$

$$\eta_{i,j} = \boldsymbol{\mu}_i^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j') + \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}_j^{\prime -1} (\boldsymbol{\mu}_j' - \boldsymbol{\mu}_i') \tag{3.10}$$

$$C2(\Theta, \Theta') = -\log \left[\frac{2 \sum_{i,j} \alpha_i \alpha_j' \sqrt{\frac{|\Phi_{i,j}|}{e^{\eta_{i,j}} |\boldsymbol{\Sigma}_i||\boldsymbol{\Sigma}_j'|}}}{\sum_{i,j} \alpha_i \alpha_j \sqrt{\frac{|\Phi_{i,j}|}{e^{\eta_{i,j}} |\boldsymbol{\Sigma}_i||\boldsymbol{\Sigma}_j|}} + \sum_{i,j} \alpha_i' \alpha_j' \sqrt{\frac{|\Phi_{i,j}|}{e^{\eta_{i,j}} |\boldsymbol{\Sigma}_i'||\boldsymbol{\Sigma}_j'|}}} \right] \tag{3.11}$$

If the distance is smaller than a given threshold, the new probe is merged with the closest probe in LTM. Otherwise an insert operation is performed. In such case, whenever the memory is full the probe with smallest number of successful recalls is deleted in order to give room for the new probe. Instead of using a fixed threshold we propose using an adaptive threshold, computed based on the minimum distance between new probes and probes on the LTM for the T previous updates (μ_δ^t). An insert occurs if $C2 - \mu_\delta^t$ is greater than the standard deviation for the same time-frame (σ_δ^t). Otherwise a merge operation is performed.

In what regards merging two mixtures, the basic approach consists of considering both mixtures as one ($p(x|\Theta) \cup p(x|\Theta')$) and then merge their components iteratively. A survey of techniques to merge components in a mixture of Gaussians can be found in (Hennig, 2010). Basically there are two main families of techniques: modality-based and those based on misclassification probability. In modality-based clustering, the components are assumed to be unimodal and then merging is performed until all mixture components are unimodal but any further merging would result in a component that is no longer unimodal. In misclassification probability approach, the notion of a cluster is not based on gaps between the densities but on how well two components (despite not being clearly separated) classify a sample generated from one of them. Split of mixture components (Blekas and Lagaris, 2007; Ueda *et al.*, 2000) can also be employed in order to avoid situations where a single component is fit over multi-modal data. However, it has been demonstrated in (Hennig, 2010) that a series of distance-based merge operations is already enough in tackling multi-modality of mixture components.

We propose the use of Hennig (Hennig, 2010) technique which is based on misclassification probability and resorts to the use of a Bhattacharyya distance. Differently than other techniques based on misclassification probability, Hennig's approach does not require the use of historical data. The Bhattacharyya distance is defined as:

$$\bar{\boldsymbol{\Sigma}} = \frac{1}{2}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2) \tag{3.12}$$

$$d_B(\Theta_1, \Theta_2) = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \bar{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) + \frac{1}{2}\log\left(\frac{|\frac{1}{2}(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)|}{\sqrt{|\boldsymbol{\Sigma}_1||\boldsymbol{\Sigma}_2|}}\right) \tag{3.13}$$

This method works as follows. Given a tuning constant $d^* < 1$, compute the Bhattacharyya distance between all pairs of components (d_B). If $e^{-d_B} < d^*$ for all components stop merging and let the mixture as is. Otherwise, merge the two components with maximum distance and repeat the whole process. The merged component parameters $\{\alpha_M, \boldsymbol{\mu}_M, \boldsymbol{\Sigma}_M\} = \{\alpha_1, \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1\} + \{\alpha_2, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2\}$ are defined as (Ueda *et al.*, 2000; Blekas and Lagaris, 2007):

$$\alpha_M = \alpha_1 + \alpha_2 \tag{3.14}$$

$$\boldsymbol{\mu}_M = \frac{\alpha_1 \boldsymbol{\mu}_1 + \alpha_2 \boldsymbol{\mu}_2}{\alpha_1 + \alpha_2} \tag{3.15}$$

$$\boldsymbol{\Sigma}_M = \frac{\alpha_1 \boldsymbol{\Sigma}_1 + \alpha_2 \boldsymbol{\Sigma}_2}{\alpha_1 + \alpha_2} \tag{3.16}$$

We propose merging the two components with minimum distance instead as it should result in smaller (more incremental) variations in the mixture components.

After the merge, if the number of mixture components is still higher than a given limit, unmerged components from the older mixture are deleted (purge). We propose the following purge approach: (1) compute Bhattacharyya distance between new/merged and old unmerged components; (2) delete the old unmerged component with the highest distance; (3) go to 1 until memory limit has been achieved.

The memory update mechanism is summarized in Algorithm 4. After optimization is over, the parameters of the new mixture (Θ_N) are estimated using position and fitness values of all particles found during the whole optimization process (step 1). This mixture along with the global best solution ($\boldsymbol{p}_g$) form a probe, to be added to the STM, replacing previous STM probe (step 2). After that, if the length of $\boldsymbol{\delta}$ (which contains the last n minimum $C2$ distances between new probes and probes in the LTM) is smaller than T (step 3), its mean and standard deviation ($\mu^t_{\boldsymbol{\delta}}$ and $\sigma^t_{\boldsymbol{\delta}}$) are set to user defined values ($\mu^0_{\boldsymbol{\delta}}$ and $\sigma^0_{\boldsymbol{\delta}}$, steps 4 and 5). Otherwise, they are computed based on $\boldsymbol{\delta}$ (steps 7 and 8). Then, the minimum $C2$ distance between new probe and probes in the LTM is added to $\boldsymbol{\delta}$ (steps 10 and 11). If the difference between the minimum $C2$ distance and $\mu^t_{\boldsymbol{\delta}}$ is greater than $\sigma^t_{\boldsymbol{\delta}}$ (step 12), the new probe is added to the LTM, noticing that the LTM probe with smallest number of recalls must be deleted if memory limit has been

reached (steps 13 to 16). Otherwise the new probe is merged with the most similar probe in the LTM and mixture elements are purged if mixture size limit has been reached (steps 18 and 19). Finally, if the limit of vector $\boldsymbol{\delta}$ has been reached, its first (oldest) element is deleted (steps 21 to 23).

3.4.3 How to retrieve solutions?

In the proposed memory retrieval technique, an attempt to recall the STM is first made. If it succeeds, the best solution is employed immediately as the embedding parameter for that image. Otherwise, recall of probes in the LTM is attempted. If no probe can be successfully recalled, STM provides solutions to be injected into the swarm for a new round of optimization.

Since the proposed technique relies on the use of a GMM of particle positions (rather than selected particles as in the case-based technique (Vellasques *et al.*, 2011)), recall requires sampling solutions from the GMM. Sampling N_s solutions from a mixture of Gaussians can be attained through a linear combination between a random vector and the eigen-decomposition of the covariance matrix, centered at the mean vector:

$$\boldsymbol{X}_s = \boldsymbol{\mu}_j + \boldsymbol{\Lambda}_j^{\frac{1}{2}} \boldsymbol{U}_j \boldsymbol{R}_s \tag{3.17}$$

where $\boldsymbol{X}_s$ is a sampled solution, s is the index of a solution sampled for the component j in the mixture ($\lfloor (N_s \alpha_j) + 0.5 \rfloor$ solutions are sampled per component), $\boldsymbol{\Lambda}_j$ and $\boldsymbol{U}_j$ are the eigen-decomposition of $\boldsymbol{\Sigma}_j$ ($\boldsymbol{\Sigma}_j = \boldsymbol{U}_j \boldsymbol{\Lambda}_j \boldsymbol{U}_j^{-1}$) and $\boldsymbol{R}_s$ is a vector with the same length as $\boldsymbol{\mu}_j$ whose elements are sampled from a normal distribution $N(0, \mathbf{I})$, being $\mathbf{I}$ the identity matrix.

The memory retrieval mechanism will basically bind the whole system together and is depicted in Algorithm 5. The best recalled solution $\boldsymbol{X}_o$ is initialized with null (step 1). After that, a given number of solutions are sampled from the STM mixture and best solution (steps 2 and 3). The fitness values of these sampled solutions are re-evaluated for the new image and if the KS statistic between these values and the sampled fitness values is smaller than a critical value (step 4), the best recalled solution is set with the solution that resulted in the smallest fitness value for the new image (step 5). Otherwise, the LTM probes are sorted in reverse order of their

Algorithm 4 Memory update mechanism.

Inputs:

k_{max} – maximum number of components with $\alpha_j > 0$.
$\mathfrak{M}_S$ – Short Term Memory.
$\mathfrak{M} = \{\mathfrak{M}_1, ..., \mathfrak{M}_{|\mathfrak{M}|}\}$ – Long Term Memory.
$\mathfrak{D}$ – optimization history (set of all particle positions and fitness values for new image).
$L_{\mathfrak{M}}$ – maximum number of probes in LTM.
$\boldsymbol{\delta}$ – last T minimum $C2$ distances between a new probe and probes in the LTM.
$|\boldsymbol{\delta}|$ – number of elements in $\boldsymbol{\delta}$.
T – maximum size of $\boldsymbol{\delta}$.
$\mu^0_{\boldsymbol{\delta}}$, $\sigma^0_{\boldsymbol{\delta}}$ – initial mean and standard deviation of $\boldsymbol{\delta}$.

Output:

Updated memory.

1: Estimate Θ_N using $\mathfrak{D}$ (Figueiredo and Jain, 2000).
2: Add Θ_N and $\boldsymbol{p}_g$ to $\mathfrak{M}_S$.
3: **if** $|\boldsymbol{\delta}| < T$ **then**
4: $\quad \mu^t_{\boldsymbol{\delta}} \leftarrow \mu^0_{\boldsymbol{\delta}}$
5: $\quad \sigma^t_{\boldsymbol{\delta}} \leftarrow \sigma^0_{\boldsymbol{\delta}}$
6: **else**
7: $\quad \mu^t_{\boldsymbol{\delta}} \leftarrow \frac{1}{|\boldsymbol{\delta}|} \sum_{i=1}^{|\boldsymbol{\delta}|} \delta_i$
8: $\quad \sigma^t_{\boldsymbol{\delta}} \leftarrow \sqrt{\frac{\sum_{i=1}^{n} (\delta_i - \mu^t_{\boldsymbol{\delta}})^2}{|\boldsymbol{\delta}|}}$
9: **end if**
10: $i^* \leftarrow argmin_i\{C2(\Theta_N, \Theta_i)\},\ \forall_{\Theta_i \in \mathfrak{M}}$
11: $\boldsymbol{\delta} \leftarrow \boldsymbol{\delta} \cup C2(\Theta_N, \Theta_{i^*})$
12: **if** $C2(\Theta_N, \Theta_{i^*}) - \mu^t_{\boldsymbol{\delta}} > \sigma^t_{\boldsymbol{\delta}}$ **then**
13: $\quad$ **if** $|\mathfrak{M}| = L_{\mathfrak{M}}$ **then**
14: $\quad\quad$ Remove LTM probe with smallest number of successful recalls.
15: $\quad$ **end if**
16: $\quad$ Add Θ_N and $\boldsymbol{p}_g$ to $\mathfrak{M}$
17: **else**
18: $\quad$ $Merge(\Theta_{i^*}, \Theta_N)$ (section 3.4.2.1)
19: $\quad$ Purge merged mixture in case number of elements exceed k_{max}.
20: **end if**
21: **if** $|\boldsymbol{\delta}| > T$ **then**
22: $\quad$ Remove δ_1.
23: **end if**

success counter (step 7) and the same process (re-sampling, followed by re-evaluation and KS test) is repeated for each probe in the LTM (steps 8 to 16). It is important to observe that in the event of a successful LTM recall, the success counter of that LTM probe is incremented (step

12) and the best recalled solution is set with the recalled solution that resulted in the smallest fitness for the new image (step 13). If the best recalled solution is null (step 18), the top STM re-sampled solutions are injected into the swarm and re-optimization is triggered (step 19). Otherwise, the embedding parameters encoded by the best recalled solution are employed in the watermarking of the new image (step 21).

Algorithm 5 Memory retrieval mechanism.

Inputs:
$\boldsymbol{Co}$ – cover image.
$\mathfrak{M}_S$ – Short Term Memory.
$\mathfrak{M} = \{\mathfrak{M}_1, ..., \mathfrak{M}_{|\mathfrak{M}|}\}$ – Long Term Memory.
N_i – amount of injected solutions (%).
D_α – critical value for KS-test.

Output:
Watermarked image (based on parameters encoded by optimal solution $\boldsymbol{X}_o$).

1: $\boldsymbol{X}_o \leftarrow$
2: $\boldsymbol{X}_{S,S} \leftarrow Sample(N_s, \mathfrak{M}_S)$
3: $\boldsymbol{X}_{S,S} \leftarrow \boldsymbol{X}_S \cup \boldsymbol{p}_{g,S}$
4: **if** $KS(\boldsymbol{F}(\boldsymbol{X}_{S,S}, \mathbf{Co}_S), \boldsymbol{F}(\boldsymbol{X}_{S,S}, \mathbf{Co})) \leq D_\alpha$ **then**
5: Set $\boldsymbol{X}_o$ with solution which resulted in smallest $\boldsymbol{F}(\boldsymbol{X}_{S,S}, \mathbf{Co})$.
6: **else**
7: Sort $\mathfrak{M}$ by $Count$ (in reverse order).
8: **for** $i \in [1, |\mathfrak{M}|]$ **do**
9: $\boldsymbol{X}_{S,i} \leftarrow Sample(N_s, \mathfrak{M}_i)$
10: $\boldsymbol{X}_{S,i} \leftarrow \boldsymbol{X}_{S,i} \cup \boldsymbol{p}_{g,i}$
11: **if** $KS(\boldsymbol{F}(\boldsymbol{X}_{S,i}, \mathbf{Co}_i), \boldsymbol{F}(\boldsymbol{X}_{S,i}, \mathbf{Co})) \leq D_\alpha$ **then**
12: $Count_i \leftarrow Count_i + 1$
13: Set $\boldsymbol{X}_o$ with solution which resulted in smallest $\boldsymbol{F}(\boldsymbol{X}_{S,i}, \mathbf{Co})$.
14: Exit for.
15: **end if**
16: **end for**
17: **end if**
18: **if** $\boldsymbol{X}_o =$ **then**
19: Inject the N_i best solutions in $\boldsymbol{X}_{S,S}$ into the swarm (replacing its N_i worst solutions), re-optimize and update memory (Algorithm 4).
20: **else**
21: Use $\boldsymbol{X}_o$ as optimal embedding parameter.
22: **end if**

The proposed memory management scheme (insert/update) is illustrated using five different bi-modal sets of 2D Gaussian points. For simplicity, all sets of points have the same covariance matrix and only their mean vectors vary. Each bi-modal set of points will simulate the behavior of particles positions during the optimization of a 2D problem. In this example the memory size is limited to three probes. Figure 3.4a shows the five bi-modal sets of points. From $t = 0$ to $t = 2$, memory update consists of insert operations (Figure 3.4b). Memory limit is reached at $t = 3$ leading to an insert followed by a delete (Figure 3.4c). At $t = 4$, one of the components appears close to a previously seen component and both components are merged (Figure 3.4d). It is worth noticing that in all cases, the knowledge about a new scenario is acquired without completely "forgetting" previous knowledge.

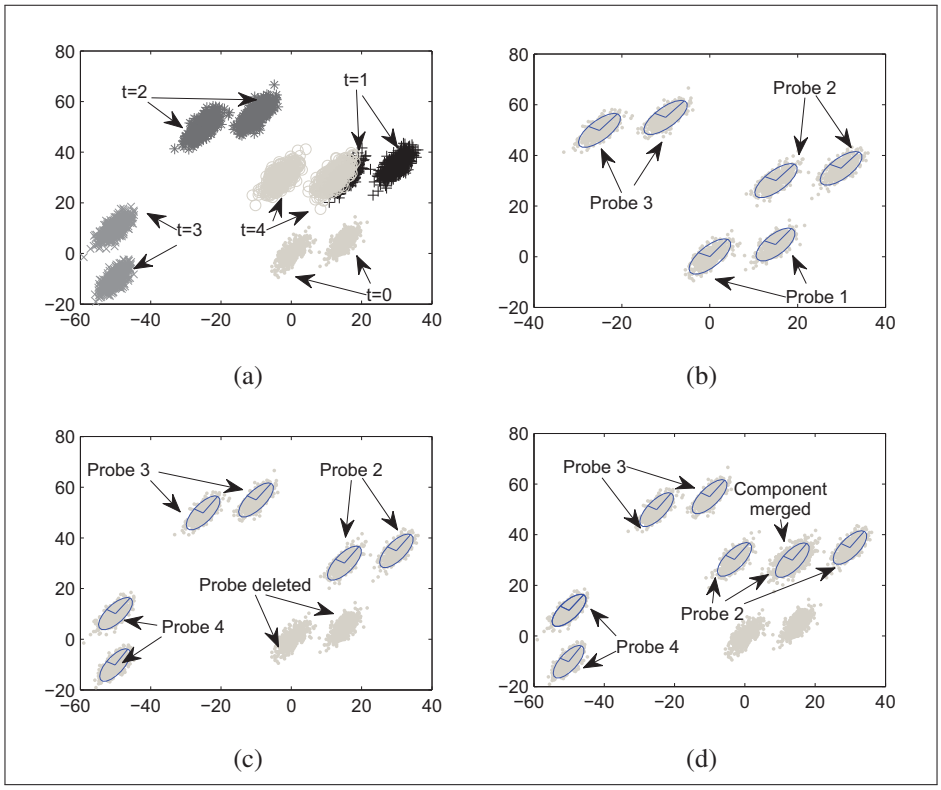


Figure 3.4 Illustration of memory update technique. (a) Bi-modal Gaussian points. (b) Three probes added between $t = 0$ and $t = 2$. (c) New probe at $t = 3$ is inserted while that of $t = 0$ is deleted. (d) Merging of probe obtained at $t = 4$ with that of $t = 1$. One of the components of the new probe was overlapped with another one of the old probe and both were merged.

3.5 Simulation results

3.5.1 Experimental protocol

3.5.1.1 Databases

The two watermarks to be employed in all experiments for all databases are same defined in (Vellasques *et al.*, 2011), namely, the 26×36 BancTec logo (Figure 3.5a) as robust watermark and the 36×26 Université du Québec logo (Figure 3.5b) as fragile watermark.

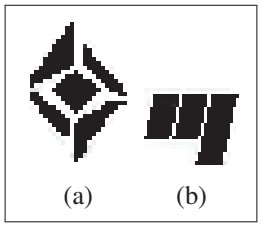


Figure 3.5 Bi-tonal logos used as watermarks. (a) 26×36 BancTec logo. (b) 36×26 Université du Québec logo.

Since the main objective of the proposed method is to tackle high throughput adaptive watermarking in heterogeneous streams of document images, the database of document images of the University of Oulu's MediaTeam (Sauvola and Kauniskangas, 1999) (OULU-1999) is employed in order to validate the performance of the proposed technique in such task (scenario A). This database is considerably heterogeneous, scanned at 300 dpi with 24-bit color encoding. Since this database is not bi-tonal, it was binarized using the same protocol as in (Vellasques *et al.*, 2011). However, it was observed that some of the images contained very large uniform regions (with only white pixels). These images lack the capacity necessary to embed the watermarks described above. Thus, a reject rule was applied: all images with less than 1872 flippable pixels were discarded (pixels with SNDM equal to 0). This is the minimum number of flippable pixels in order to embed the 936-bit robust watermark presented above with a quantization step size ($Q = 4$) which is the minimum level of robustness necessary for multi-level embedding. With this rule, 15 of the 512 images from the OULU-1999 database were excluded. The second objective of the proposed method is to allow learning the different categories of problems

found throughout the stream of optimization problems. To validate this, two separate sets of images – training and testing – are required. For this reason, the OULU-1999 database was split in two subsets. The training (memorization) subset contains 100 images chosen randomly from OULU-1999 and is named OULU-1999-TRAIN. The remaining 397 images compose the testing (generalization) subset which is named OULU-1999-TEST. Since the images on this database are from 19 different categories (Table 3.2), there is a lot of variation in the size and number of flippable pixels among these images.

Although the proposed technique was devised to tackle intelligent watermarking of heterogeneous image streams, in a real life scenario it needs to adapt to watermarking of homogeneous image streams as well. To validate this, the proposed technique will be also evaluated in two different (training and testing) homogeneous image streams, namely TITI-61 and CVIU-113-3-4 (Vellasques *et al.*, 2011) (scenario B). Finally, the performance on an unconstrained (homogeneous/heterogeneous) stream (scenario C) will be validated. For this purpose, the OULU-1999-TEST and CVIU-113-3-4 streams were concatenated and the images were shuffled in order to create a larger stream named SHUFFLE, to assess how does the proposed approach scales as the length of the stream grows. A larger learning stream was also created by concatenating TITI-61 and OULU-1999-TRAIN streams.

3.5.1.2 Methodology

The memory management mechanism should mitigate redundancy in the LTM. Therefore, a sensitivity analysis will be conducted in a first moment in order to find out how do the distance between probes and sampled particles diversity relate. The current method will be applied to the OULU-1999-TRAIN database but forcing re-optimization for each image and without using any memory management technique. The purpose of this experiment is to build a large memory (containing 100 probes) and then assess the distance between these probes in order to set an initial distance threshold for the proposed technique. As each probe is inserted in the LTM, the C2 distance (Sfikas *et al.*, 2005) between this probe and the probes already in the memory will be computed. Then 2000 solutions will be sampled uniformly from all probes and the normalized mean of the pairwise distance among individuals in the population D_{PW}^{N}

Table 3.2 OULU-1999 database structure.

Category	OULU-1999-TRAIN					OULU-1999-TEST				
	#	# Pixels				#	# Pixels			
		Regular		Flippable			Regular		Flippable	
		Min	Max	Min	Max		Min	Max	Min	Max
Addresslist	0	0	0	0	0	6	2.2×10^6	6.6×10^6	3.7×10^5	2×10^6
Advertisement	5	4.9×10^6	7.9×10^6	8.1×10^5	2.6×10^6	19	1.1×10^6	8.1×10^6	1.5×10^5	2.5×10^6
Article	51	1.8×10^6	7.9×10^6	2.5×10^5	3.0×10^6	180	2.0×10^6	15.7×10^6	2.4×10^5	3.0×10^6
Businesscards	1	6.2×10^5	6.2×10^5	9.8×10^4	9.8×10^4	10	5.3×10^5	1.1×10^6	7.8×10^4	3.4×10^5
Check	0	0	0	0	0	3	3.4×10^5	1.4×10^6	1.3×10^5	1.9×10^5
Color Segmentation	1	2.5×10^6	2.5×10^6	7.9×10^5	7.9×10^5	7	1.5×10^6	7.3×10^6	4.5×10^5	3.3×10^6
Correspondence	6	2.0×10^6	5.2×10^6	2.1×10^5	1.1×10^6	18	1.1×10^6	4.9×10^6	1.4×10^5	8.2×10^5
Dictionary	1	2.8×10^6	2.8×10^6	3.3×10^5	3.3×10^5	9	1.6×10^6	3.3×10^6	2.3×10^5	6.6×10^5
Form	9	7.3×10^5	5.5×10^6	1.2×10^5	1.1×10^6	14	4.5×10^5	3.9×10^6	7.6×10^4	7.5×10^5
Line Drawing	0	0	0	0	0	10	1.5×10^6	7.1×10^6	1.3×10^5	1.1×10^6
Manual	6	3.0×10^6	4.1×10^6	2.8×10^5	8.7×10^5	29	2.4×10^6	4.1×10^6	2.6×10^5	8.6×10^5
Math	4	3.2×10^6	3.9×10^6	2.0×10^5	3.1×10^5	13	3.2×10^6	3.9×10^6	1.8×10^5	3.8×10^5
Music	0	0	0	0	0	4	3.9×10^5	2.1×10^6	8.8×10^4	4.0×10^5
Newsletter	4	7.6×10^6	7.9×10^6	1.3×10^6	1.7×10^6	37	1.5×10^6	7.9×10^6	1.3×10^5	2.2×10^6
Outline	4	1.6×10^6	4.1×10^6	2.5×10^5	9.1×10^5	13	3.1×10^6	5.2×10^6	3.2×10^5	1.0×10^6
Phonebook	4	7.9×10^6	8.1×10^6	2.3×10^3	2.4×10^3	3	7.9×10^6	8.1×10^6	1.4×10^6	2.2×10^6
Program Listing	2	3.8×10^6	7.0×10^6	6.6×10^5	1.3×10^6	10	3.6×10^6	7.3×10^6	3.9×10^5	2.0×10^6
Street Map	0	0	0	0	0	5	1.8×10^6	1.1×10^7	3.5×10^5	6.2×10^6
Terrainmap	2	7.0×10^6	1.0×10^7	2.6×10^6	6.0×10^6	7	2.9×10^6	1.1×10^7	1.2×10^6	6.2×10^6
Total:	**100**					**397**				

(Corriveau *et al.*, 2012) will be computed for the sampled solutions:

$$D_{PW}^{N} = \frac{\frac{2}{|X|(|X|-1)} \sum_{i=2}^{|X|} \sum_{j=1}^{i-1} \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2}}{NMDF} \tag{3.18}$$

where $|X|$ is the population size, $x_{i,k}$ is the k^{th} parameter encoded by the i^{th} individual, d is the landscape dimensionality and $NMDF$ is the normalization (factor) with maximum diversity so far. This metric reflects quite well the population diversity.

Considering the number of probes in LTM is $|\mathfrak{M}|$, this involves sampling $2000/|\mathfrak{M}|$ from each probe. A plot of the minimum distance between the new probe and the probes already in the memory (min_{C2}) versus the diversity of the sampled population should show how does limiting the number of insert operations based on a distance threshold impacts sampling diversity.

We propose a novel metric based on the same principle of D_{PW}^{N} but tailored to measure the diversity of the LTM, namely the normalized pairwise distance between probes:

$$D_{PWM}^{N} = \frac{\frac{2}{|\mathfrak{M}|(|\mathfrak{M}|-1)} \sum_{i=2}^{|\mathfrak{M}|} \sum_{j=1}^{i-1} C2(\Theta_i, \Theta_j)}{NMDF_{C2}} \tag{3.19}$$

where $NMDF_{C2}$ is the the normalization (factor) with maximum diversity so far (applied to the $C2$ metric). This metric will show the amount of inter-probe diversity while D_{PW}^{N} will show the amount of intra-probe diversity.

The proposed management strategy should allow the memory to quickly adapt to an abrupt change in the stream of optimization problems. First we have to define what an abrupt change is. In this specific scenario an abrupt change is a change in the stream of optimization problems that requires re-optimization to be triggered. Since defining when re-optimization should be triggered is subjective, we propose the use of Kullback-Leibler (KL) (Pérez-Cruz, 2008) divergence measure between the cumulative sets of particles of two consequent optimization problems in order to precisely verify this variation. The KL divergence is a measure of information gain between two distributions. A cumulative set of particles at instant t (or $\boldsymbol{X}_{C,t}$) is the set of all particles seen in all generations of all problem instances up to t. The KL divergence

between cumulative sets of particles at instants t and $t-1$ is defined as $D_k(\boldsymbol{X}_{C,t-1}||\boldsymbol{X}_{C,t})$. The method proposed in (Pérez-Cruz, 2008) is non-parametric and depends on a k-nearest neighborhood estimate (that is, depends on a neighborhood size parameter). This parameter was set to 10 in our experiments as seen in (Pérez-Cruz, 2008).

The number of previous updates T employed to compute the adaptive threshold will be set to 10. The mean and standard deviation of the minimum distance obtained in the memory fill up experiments with no attack (which are 361.7 and 172.3, respectively) will be employed as an initial minimum distance threshold in the memory update. These values were obtained by simply measuring the minimum $C2$ distance during inserts for the memory fill up experiments (which resulted in 99 $C2$ values) and then, computing their mean and standard deviation.

In order to measure the impact in the computational cost we will analyze how does the number of fitness evaluations behave in different scenarios. One of the metrics that will be employed to this end is the average number of fitness evaluations per image ($AFPI$). A second metric to be employed is the cumulative number of fitness evaluations (F_{Evals}) which is the total number of fitness evaluations required to optimize the whole image stream. A third is the decrease in the number of fitness evaluations (DFE), computed as:

$$DFE = 1 - \frac{F_{Evals,M}}{F_{Evals,F}} \tag{3.20}$$

where $F_{Evals,M}$ is the cumulative number of fitness evaluations for the memory based approach and $F_{Evals,F}$ is the cumulative number of fitness evaluations for full optimization. For each experiment, the mean and standard variation of AFPI, the F_{Evals} and the DFE is presented.

The reference points for the Chebyshev Weighted Aggregation were set to $r_1 = r_2 = r_3 = 0.01$ based on sensitivity analysis using the OULU-1999-TRAIN dataset. The scaling factor of the DRDM (α_r) was set to 0.53 based on the largest DRDM value found for all fitness evaluations during the full optimization of all images of the OULU-1999-TRAIN dataset. These parameters have been used in the test streams to validate their generalization performance.

The confidence level (α) of the KS statistic will be set to 0.95, which corresponds to a coefficient $c_\alpha = 1.36$ and a critical value (D_α) of 0.43 in order to allow a comparison with the results reported in (Vellasques *et al.*, 2011). The LTM size is limited to 20 probes. All the simulations were performed first with no attack and then with cropping of 1%.

DPSO parameters are set as in (Vellasques *et al.*, 2011). Constants c_1 and c_2 are set to 2.05 while χ is set to 0.7298. Population size is set to 20 particles and optimization halts if the global best has not improved for 20 iterations. The neighborhood size of the L-Best topology is set to 3.

3.5.2 Overview

In terms of computational burden, the GMM-based approach outperformed the case-based approach for the heterogeneous streams and underperformed for some of the homogeneous streams (Table 3.3).

However, the watermarking performance of the GMM-based approach is equivalent to that of the case-based approach for the heterogeneous streams but at a smaller computational burden (Table 3.4). Moreover, there was a significant improvement in watermarking performance for the homogeneous streams (mainly due to the modified fitness function). It is important to observe that mainly for the cropping 1%, the worsening in computational cost is largely offset by the improvement in watermarking performance.

Figure 3.6 summarizes the computational and memory burden results.

3.5.3 Scenario A – optimization of heterogeneous streams of bi-tonal images using memory-based DPSO versus full PSO

3.5.3.1 LTM fill up

In the first experiment, performed on the OULU-1999-TRAIN stream, the memory limit was removed and re-optimization was forced on each image transition. This led to the creation of 100 probes. Figure 3.7 shows the normalized pairwise distance between probes (D^N_{PWM})

Table 3.3 Computational cost performance. $AFPI$ is the average number of fitness evaluations per image where the mean μ and standard deviation σ are presented as $\mu(\sigma)$. F_{Evals} is the cumulative number of fitness evaluations required to optimize the whole stream and DFE is the decrease in the number of fitness evaluations compared to full optimization. An asterisk (*) indicates results extracted from (Vellasques *et al.*, 2011).

Attack	Database	Learning	Full PSO		Case-based			GMM-based		
			$AFPI$	F_{Evals}	$AFPI$	F_{Evals}	DFE	$AFPI$	F_{Evals}	DFE
No attack	OULU-1999-TRAIN	No	925 (286)	92520	564 (630)	56380	39.1%	66 (194)	6580	92.9%
No attack	OULU-1999-TEST	No	1007 (341)	399840	270 (551)	107060	73.2%	59 (188)	23280	94.2%
No attack	OULU-1999-TEST	Yes	1007 (341)	399840	464 (842)	184180	53.9%	42 (133)	16700	95.8%
No attack	TITI-61	No	844 (226)*	51460*	46 (134)*	2760*	94.6%*	84 (224)	5140	92.6%
No attack	CVIU-113-3-4	No	882 (251)*	301580*	32 (103)*	10720*	96.4%*	76 (233)	26000	91.4%
No attack	CVIU-113-3-4	Yes	882 (251)*	301580*	31 (83)*	10560*	96.5%*	49 (157)	16600	95.4%
No attack	SHUFFLE	No	1026 (345)	758500	273 (571)	201640	73.4%	66 (189)	48840	93.6%
No attack	SHUFFLE	Yes	1026 (345)	758500	259 (613)	191240	74.8%	54 (179)	40220	94.7%
Cropping 1%	OULU-1999-TRAIN	No	887 (340)	88740	351 (455)	35100	60.5%	179 (363)	17860	79.9%
Cropping 1%	OULU-1999-TEST	No	860 (310)	341520	177 (351)	70300	79.4%	83 (212)	32920	90.4%
Cropping 1%	OULU-1999-TEST	Yes	860 (310)	341520	148 (301)	58940	82.7%	67 (205)	26760	92.2%
Cropping 1%	TITI-61	No	911 (237)*	55580*	66 (200)*	3960*	92.9%*	52 (178)	3200	94.8%
Cropping 1%	CVIU-113-3-4	No	872 (251)*	298100*	26 (36)*	8740*	97.1%*	50 (166)	16980	94.5%
Cropping 1%	CVIU-113-3-4	Yes	872 (251)*	298100*	25 (10)*	8480*	97.2%*	21 (4)	7120	97.7%
Cropping 1%	SHUFFLE	No	887 (320)	798100	151 (292)	111420	86%	67 (194)	49780	93.8%
Cropping 1%	SHUFFLE	Yes	887 (320)	798100	128 (252)	94780	88.1%	49 (136)	36300	95.5%

Table 3.4 Watermarking performance. Here, † is the $DRDM$, ‡ is the BCR robust, § is the BCR fragile. For all values, the mean μ and standard deviation σ per image are presented in the following form: $\mu(\sigma)$. $DRDM$ is presented with two decimal points and BCR is presented in percentage (%) with one decimal point. An asterisk (*) indicates results extracted from (Vellasques *et al.*, 2011).

Attack	**Database**	**Learning**	**Full PSO**			**Case-based**			**GMM-based**		
			†	‡	§	†	‡	§	†	‡	§
No attack	OULU-1999-TRAIN	No	0 (0)	100 (0)	100 (0)	0 (0)	100 (0)	100 (0)	0 (0)	100 (0)	100 (0)
No attack	OULU-1999-TEST	No	0 (0)	100 (0)	100 (0)	0 (0)	100 (0)	100 (0)	0 (0)	100 (0)	100 (0)
No attack	OULU-1999-TEST	Yes	0 (0)	100 (0)	100 (0)	0 (0)	100 (0)	100 (0)	0 (0)	99.9 (0.4)	99.9 (0.7)
No attack	TITI-61	No	0 (0)*	99.9 (0.5)*	99.7 (0.6)*	0 (0)*	99.8 (0.9)*	99.6 (1.2)*	0 (0)	100 (0)	100 (0)
No attack	CVIU-113-3-4	No	0 (0)*	99.5 (3.6)*	99.3 (3)*	0 (0)*	99.5 (3.3)*	99.6 (2.7)*	0 (0)	100 (0)	100 (0)
No attack	CVIU-113-3-4	Yes	0 (0)*	99.5 (3.6)*	99.3 (3)*	0 (0)*	99.4 (3.3)*	99.2 (2.8)*	0 (0)	100 (0)	100 (0)
No attack	SHUFFLE	No	0 (0)	100 (0)	100 (0)	0 (0)	100 (0)	100 (0.1)	0 (0)	100 (0)	100 (0)
No attack	SHUFFLE	Yes	0 (0)	100 (0)	100 (0)	0 (0)	100(0)	100 (0.1)	0 (0)	100 (0)	100 (0)
Cropping 1%	OULU-1999-TRAIN	No	0.03 (0.03)	98.4 (2.1)	99.7 (0.6)	0.03 (0.03)	97.9 (2.6)	99.6 (1)	0.03 (0.03)	97.1 (3.8)	99.4 (1)
Cropping 1%	OULU-1999-TEST	No	0.03 (0.04)	98.4 (2.2)	99.6 (0.6)	0.03 (0.03)	97.2 (3.6)	99 (1.6)	0.03 (0.03)	96.7 (4)	99.1 (1.5)
Cropping 1%	OULU-1999-TEST	Yes	0.03 (0.03)	98.4 (2.2)	99.6 (0.6)	0.03 (0.03)	97.5 (2.8)	99.3 (1.2)	0.03 (0.04)	97.5 (3.3)	99.4 (1.1)
Cropping 1%	TITI-61	No	0 (0)*	*92 (6.5)*	94 (4)*	0 (0)*	92.4 (6)*	94.8 (4.5)*	0.03 (0.03)	99 (1.8)	99.7 (0.04)
Cropping 1%	CVIU-113-3-4	No	0 (0)*	89.6 (7.1)*	92.5 (5.3)*	0 (0)*	86.6 (7.2)*	90 (5.9)*	0.04 (0.05)	98.3 (3)	99.5 (0.8)
Cropping 1%	CVIU-113-3-4	Yes	0 (0)*	89.6 (7.1)*	92.5 (5.3)*	0(0)*	90.5 (6.4)*	93.4 (5.1)*	0.04 (0.06)	98.1 (0.03)	99.4 (1)
Cropping 1%	SHUFFLE	No	0.03 (0.04)	98.6 (2.2)	99.6 (0.5)	0.03 (0.04)	97.9 (3)	99.3 (1.1)	0.03 (0.04)	97.1 (4.4)	98.9 (1.8)
Cropping 1%	SHUFFLE	Yes	0.03 (0.04)	98.6 (2.2)	99.6 (0.5)	0.03 (0.04)	98 (2.8)	99.4 (1)	0.03 (0.04)	97.1 (4.3)	99.1 (1.4)

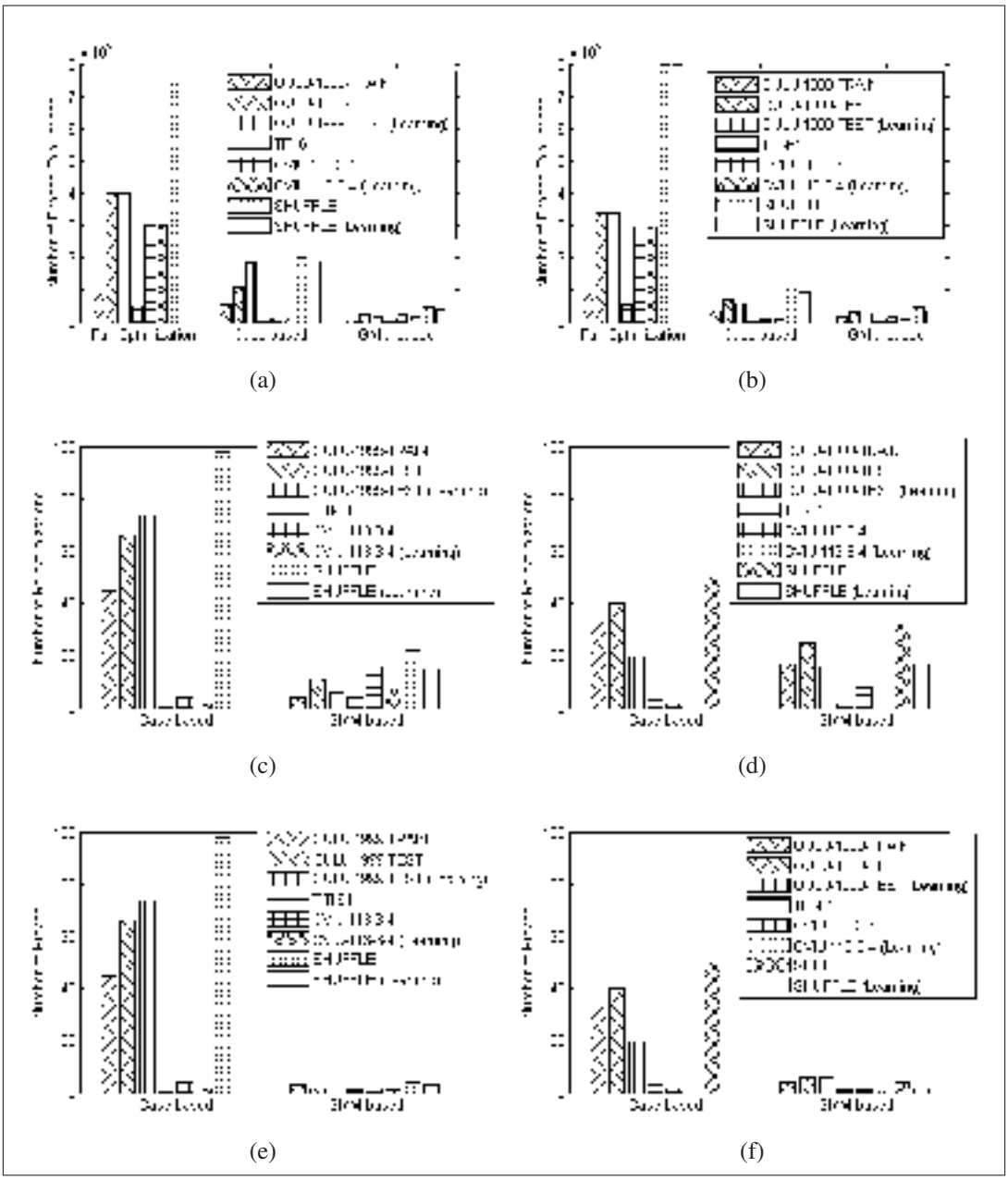


Figure 3.6 Comparison of computational and memory burden for the different approaches. (a) Number of fitness evaluations, no attack. (b) Number of fitness evaluations, cropping 1%. (c) Number of re-optimizations, no attack. (d) Number of re-optimizations, cropping 1%. (d) Number of probes, no attack. (e) Number of probes, cropping 1%.

for both, no attack and cropping 1%. It is possible to observe that in both cases, inter-probe diversity decreases steeply until image 11 for the cropping 1% case and image 12 for the no

attack case. After that, for the no attack case it rebounds sharply until image 20 and then becomes stable. For the cropping 1% it rebounds softly and becomes stable.

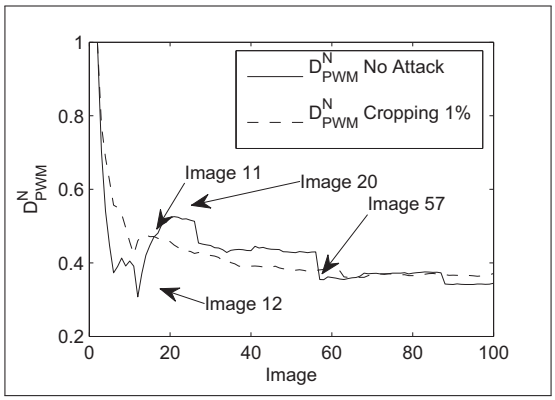


Figure 3.7 LTM diversity (OULU-1999-TRAIN).

It is interesting to observe that the sampling diversity has a similar behavior (Figure 3.8). If a probe brings new knowledge to the LTM, the sampling diversity should increase. However, it follows a downward trend as new probes are added indiscriminately which means that in most cases, the new probes do not imply in new knowledge about the fitness landscape (the sampled solutions are just probing already probed areas).

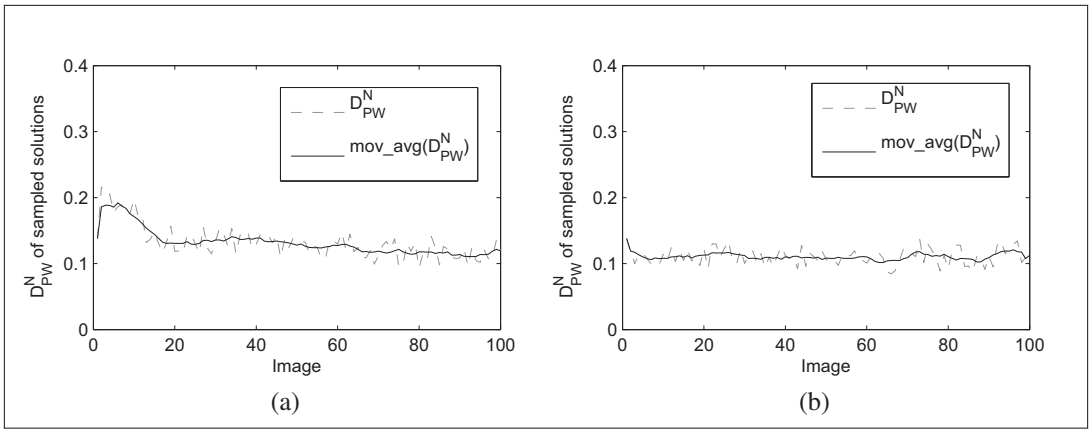


Figure 3.8 Diversity of 2000 solutions sampled uniformly for all probes (D^N_{PW}) including moving average with window size 10 ($mov\_avg(D^N_{PW})$) for OULU-1999-TRAIN stream. (a) No attack. (b) Cropping 1%.

In Figure 3.9 it is possible to observe that the minimum distance between new probes and probes already in the memory behaves in a similar manner. Although the minimum distance

itself is less stable than the LTM diversity, its moving average ($mov\_avg(min_{C2})$) follows a steep downward trend for the first 11-12 images and then becomes stable. It is worth noticing that a steep variation in the minimum distance is associated with a steep change in the LTM diversity. For example, for the no attack case, the D^N_{PWM} decreases steeply between images 1 and 12 and then increases gradually until image 20. Nearly at the same time-frame, $mov\_avg(min_{C2})$ follows a similar trend. It is slightly slower because of the window size chosen. A smaller window size would give less importance to the min_{C2} of previous probes and make it follow more rapidly the trend of D^N_{PWM}. The same phenomenon can be observed for the cropping 1% case.

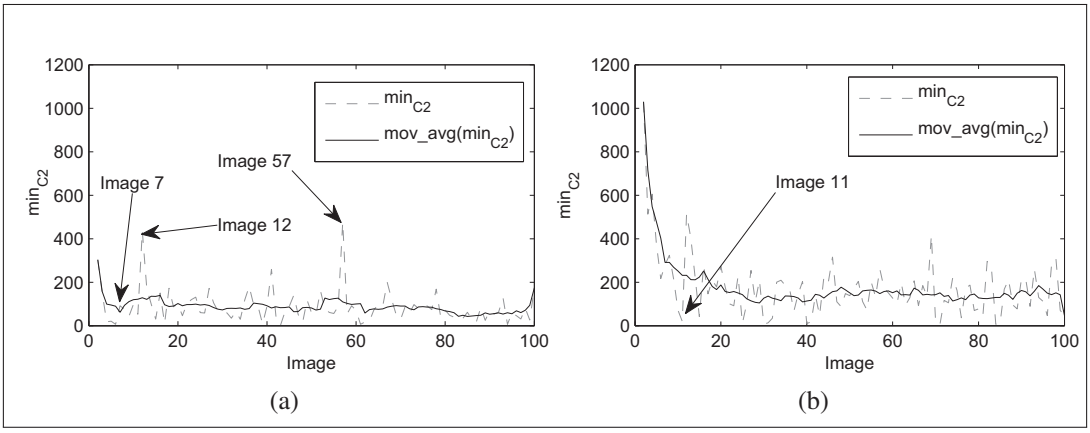


Figure 3.9 Minimum $C2$ distances between new probes and probes already in the memory (min_{C2}) for OULU-1999-TRAIN stream. Moving average of min_{C2} with window size 10 ($mov\_avg(min_{C2})$) is also depicted.
(a) No attack. (b) Cropping 1%.

The Kullback-Leibler (KL) divergence (Pérez-Cruz, 2008) between the cumulative sets of particles at instants t and $t-1$ (Figure 3.10) behaves similarly. It is possible to see here that from an information theoretical standpoint, the particles of a given optimization problem provide new information about the stream of optimization problems until around image 30 (for both no attack and cropping 1%). After that, except for small disturbances like for image 60 in the no attack case, swarm solutions do not bring new knowledge about the stream of optimization problems. Most importantly, the KL divergence follows a trend similar to that of the moving average of the minimum $C2$ distances seen in Figure 3.9. Therefore, the proposed strategy

of only performing an insert operation if distance between the new probe and probes already in the memory is above a certain threshold should maximize the amount of new information brought by each new probe.

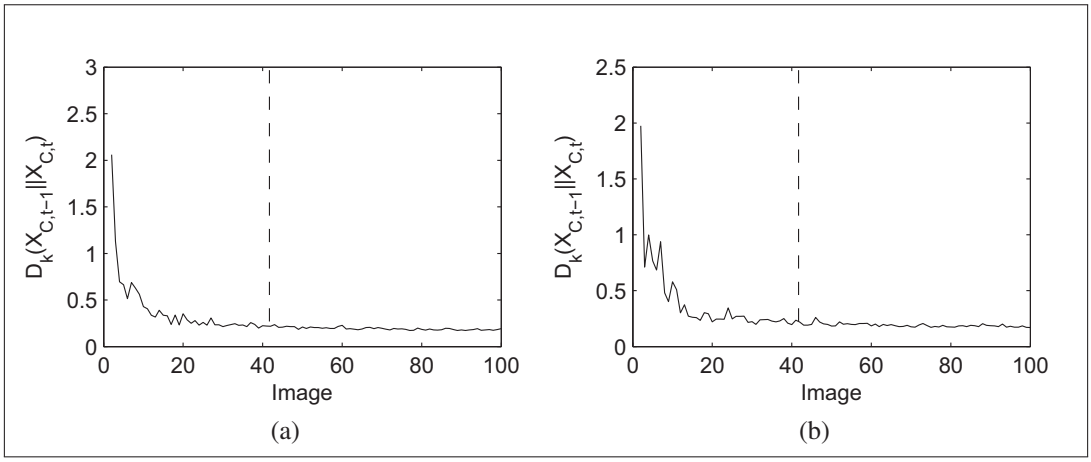


Figure 3.10 Kullback-Leibler divergence between cumulative sets of particles at at instants t and $t-1$. (a) No attack. (b) Cropping 1%.

3.5.3.2 Adaptive memory management

The GMM-based technique resulted in less re-optimizations when compared with the case-based approach for all experiments involving heterogeneous image streams which consequently led to a bigger decrease in the number of fitness evaluations when compared to full optimization. It is also important to mention that the use of a training sequence resulted in a further decrease in computational burden for the OULU-1999-TEST stream in both cases (with and without attack). Despite the decrease in computational burden, the watermarking performance of the GMM-based technique is comparable to that of the case-based technique. The reason is that the solutions sampled from the GMM are less biased to a particular optimization problem than the case-based solutions.

The same was observed for the cropping 1% case. The proposed GMM-based memory scheme resulted in considerably less re-optimizations than the case-based memory scheme for the three heterogeneous streams with an equivalent watermarking performance. For this reason, the number of fitness evaluations decreased significantly when compared to full optimization.

An analysis of LTM dynamics for the OULU-1999-TRAIN stream shows that the proposed memory management scheme resulted in a more diverse memory than that obtained in the memory fill-up experiment (Figure 3.11). What is interesting here is that for the no attack case, re-optimization was triggered 28 times. However, it resulted in an insert for only 5 of these cases. For the remaining 23 cases, a merge took part. A similar situation occurred for the cropping 1% case. Re-optimization was triggered 21 times but the number of inserts was 4 (with 17 merges).

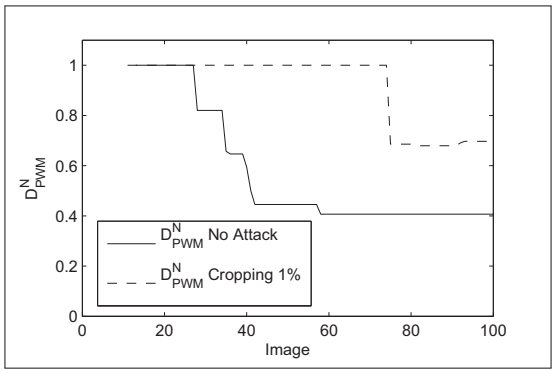


Figure 3.11 LTM diversity (OULU-1999-TRAIN, with memory management).

At the same time, the sampled solutions have more diversity than when insert is used indiscriminately (Figure 3.12). It is possible to observe also that the two plots in Figure 3.12 are more stable than those of Figure 3.8. This means that the sampling obtained by the use of the proposed memory scheme not only improves diversity but is also more consistent. This shows that this strategy of limiting insert operations to cases where the distance between new probes and probes in the memory is above an historic average helps to improve the diversity of the sampled solutions.

The plot of minimum C2 distance between new probes and probes in the memory (Figure 3.13) gives another perspective about the memory dynamics. In this plot, a min_{C2} of zero means that the memory was not updated (that is, re-optimization was not triggered). It is possible to observe that insert operations have in general a min_{C2} that is many times greater than that of merge operations. It becomes clear as well that in both cases, for the first 30 images, the update frequency is high, which means that learning (memorization) is taking place, and then updates become less frequent. When we go back to the KL divergence plot in Figure 3.10 it becomes

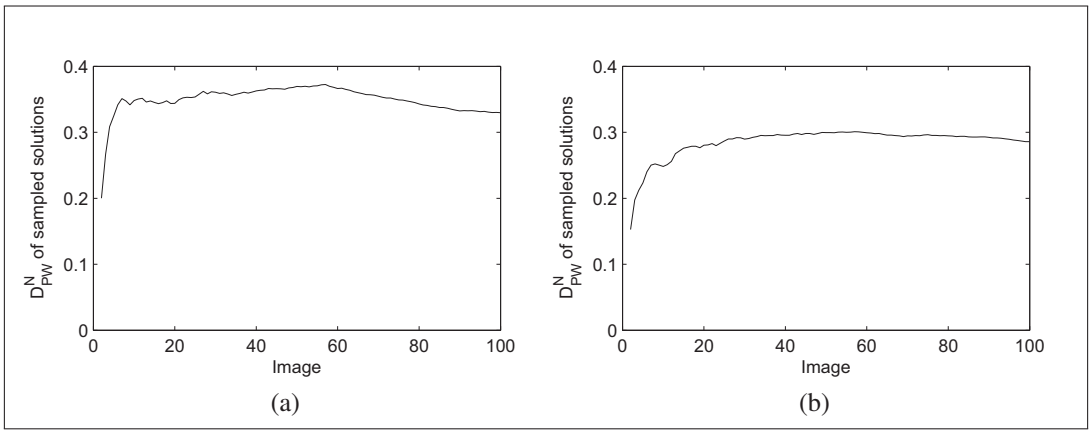


Figure 3.12 Diversity of 2000 solutions sampled uniformly for all probes (D^N_{PW}) for OULU-1999-TRAIN stream (with memory management). (a) No attack. (b) Cropping 1%.

clear that this memorization phase occurs when there is novelty in the stream of optimization problems.

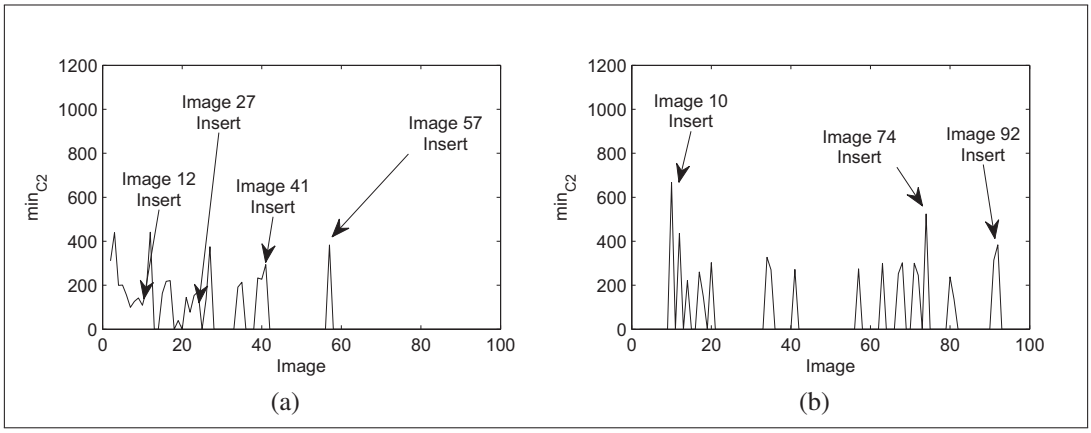


Figure 3.13 Minimum $C2$ distance between new probes and probes already in the memory (min_{C2}) for OULU-1999-TRAIN stream (with memory management). (a) No attack. (b) Cropping 1%.

3.5.3.3 Impact of choice of confidence level

In terms of memory size, the worst case scenario for the GMM-based technique results in a memory that is a fraction of the size obtained for the case-based approach (Figure 3.14).

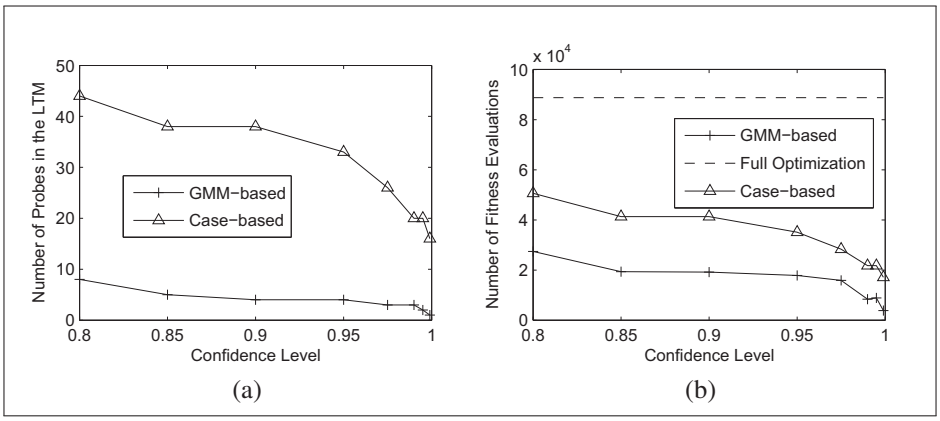


Figure 3.14 Number of LTM probes produced by the case-based and GMM-based techniques as a function of confidence level for the OULU-1999-TRAIN with cropping of 1%. (a) LTM size. (b) Number of fitness evaluations.

Figure 3.15 shows the cumulative number of fitness evaluations for the case-based and GMM-based approaches with a confidence level of 0.8 (OULU-1999-TEST with learning, no attack). It is possible to observe that between images 137 and 240 the computational cost for the case-based memory approach is higher than that of full optimization while for the GMM-based approach it is practically stable after a learning phase that lasts until image 80. This illustrates the main limitation of case-based memory management strategy and the main advantage of GMM-based memory. It is important to observe that this result was obtained in a considerably small database. In a real world scenario, involving thousands or even millions of images, an ever growing memory would pose a serious issue to the performance of the case-based intelligent watermarking system.

The main reason for improved performance when compared with the case-based approach is that probe solutions in the case-based memory scheme are less diverse than those of the GMM-based memory. That is, case-based solutions only cover the near optimal region and for this reason are very sensitive to small variations in fitness values caused by a change of type II (basically, these solutions are over-fit to the images that generated them). However, the solutions sampled from the GMM have a more general coverage of the fitness landscape, mainly because they are generated from a density estimate of all solutions found during optimization

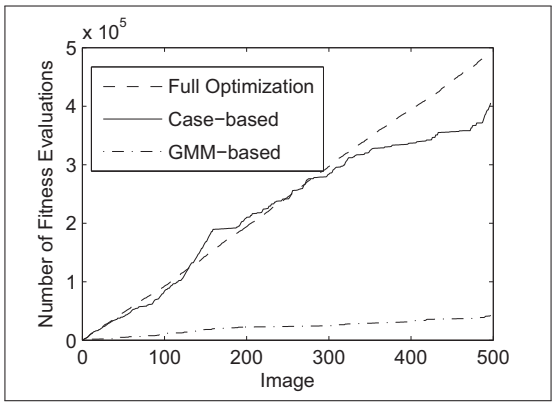


Figure 3.15 Cumulative number of fitness evaluations for the case-based, GMM-based memory scheme and full optimization for OULU-1999-TEST (Learning), no attack, confidence level of 0.8.

and consequently, perform better in avoiding unnecessary re-optimizations than the case-based approach.

3.5.3.4 Memorization performance

In the first memorization experiment we picked a probe that resulted in re-optimization followed by a merge for OULU-1999-TRAIN with cropping of 1% (the probe of image 38) and performed multiple attempts to recall the new and merged probes in three situations: (1) new probe before merge; (2) old probe before merge; (3) merged probe. The first simulation should give an idea of the true acceptance rate of the proposed technique while the second simulation should give an idea of its true reject rate. The third simulation by its way should give an idea of at what point, incorporating new knowledge will improve the recall rate of a previous probe (adaptability).

In scenario (1), the newly created probe was recalled in all cases, which means a true acceptance rate of 100% (obviously, for this sample size, or put differently, a false reject rate smaller than 1%). In scenario (2), the old probe was accepted only 30 times of the cases, which means a true reject rate of 70%. Finally, in scenario (3), the merged probe resulted in an accept rate of 73%. That is, the merged probe has a better performance for image 38 than the old unmerged

probe. At the same time, it is not as fit to the new image as the newly created (unmerged) probe which means it is less biased to a specific image.

In the second memorization experiment, the same stream (OULU-1999-TRAIN) with cropping of 1% was optimized twice, but using the memory of the first run as a starting point for the second run. The first run resulted in 17 re-optimizations while the second run resulted only in 10. This demonstrates that the proposed approach can memorize a stream of optimization problems quite well. Then, the merge operator was de-activated and the same experiment was repeated. This time the second run resulted in 3 re-optimizations. It can be said that such increase in the number of re-optimizations for the merge operator was the result of the smaller bias of that approach. That is, the merge operator, as observed in the first memorization experiments, results in probes that are less tuned to specific images (more general).

3.5.3.5 Other attacks

It is possible to observe in Table 3.5 that the computational cost proposed approach is not considerably affected by an increase in the attack level or by a different removal attack such as salt & pepper (S&P).

Regarding the watermarking performance (Table 3.6), the behavior was similar to the cases of no attack and cropping of 1%: a slight variation when compared to full optimization, but largely offset by gains in computational burden.

3.5.3.6 Adaptation performance

Memory adaptability is another important aspect in the given scenario. It is reasonable to consider that in the course of its normal operation, the set of attacks an intelligent watermarking system must deal with is expected to change and that the memory should be capable to adapt to such change. In such case, the system must avoid recalling solutions that result in poor watermarking performance. To validate this, we performed a memory adaptation experiment (Figure 3.16).

Table 3.5 Computational cost performance. $AFPI$ is the average number of fitness evaluations per image where the mean μ and standard deviation σ are presented as $\mu(\sigma)$. F_{Evals} is the cumulative number of fitness evaluations required to optimize the whole stream and DFE is the decrease in the number of fitness evaluations compared to full optimization.

Attack	Database	Learning	Full PSO		Case-based			GMM-based		
			$AFPI$	F_{Evals}	$AFPI$	F_{Evals}	DFE	$AFPI$	F_{Evals}	DFE
Cropping 2%	OULU-1999-TRAIN	No	860 (335)	86040	185 (382)	18520	78.5%	72 (187)	7240	91.6%
Cropping 2%	OULU-1999-TEST	No	828 (309)	328900	140 (342)	55740	83.1%	64 (179)	25560	92.2%
Cropping 2%	OULU-1999-TEST	Yes	828 (309)	328900	113 (290)	44940	86.3%	50 (150)	19800	94%
S&P 0.02	OULU-1999-TRAIN	No	893 (354)	89280	462 (507)	46220	48.2%	163 (360)	16320	81.7%
S&P 0.02	OULU-1999-TEST	No	978 (379)	388220	253 (433)	100580	74.1%	92 (281)	36360	90.6%
S&P 0.02	OULU-1999-TEST	Yes	978 (379)	388220	157 (321)	62200	84%	42 (133)	16560	95.7%

Table 3.6 Watermarking performance. Here, † is the $DRDM$, ‡ is the BCR robust, § is the BCR fragile. For all values, the mean μ and standard deviation σ per image are presented in the following form: $\mu(\sigma)$. $DRDM$ is presented with two decimal points and BCR is presented in percentage (%) with one decimal point.

Attack	Database	Learning	Full PSO			Case-based			GMM-based		
			†	‡	§	†	‡	§	†	‡	§
Cropping 2%	OULU-1999-TRAIN	No	0.04 (0.05)	98.2 (2.7)	99.9 (0.4)	0.04 (0.05)	98 (3.1)	99.9 (0.5)	0.04 (0.06)	97.1 (3.8)	99.8 (0.6)
Cropping 2%	OULU-1999-TEST	No	0.04 (0.04)	98 (3)	99.8 (0.7)	0.03 (0.04)	97 (4.5)	99.6 (1.4)	0.04 (0.04)	95.4 (5.7)	99.3 (2)
Cropping 2%	OULU-1999-TEST	Yes	0.04 (0.04)	98 (3)	99.8 (0.7)	0.04 (0.05)	97.1 (4.4)	99.6 (1.2)	0.04 (0.05)	94.7 (6.4)	99.1 (1.9)
S&P 0.02	OULU-1999-TRAIN	No	0.03 (0.03)	97.9 (2.6)	99.7 (0.5)	0.03 (0.03)	97.9 (3.1)	99.7 (0.5)	0.03 (0.03)	97.1 (4.3)	99.3 (1.3)
S&P 0.02	OULU-1999-TEST	No	0.03 (0.04)	98 (2.4)	99.6 (0.6)	0.02 (0.03)	97.2 (3.3)	98.9 (1.4)	0.03 (0.04)	97.2 (3.6)	99.4 (1)
S&P 0.02	OULU-1999-TEST	Yes	0.03 (0.04)	98 (2.4)	99.6 (0.6)	0.03 (0.04)	97.1 (3.6)	99.4 (1.1)	0.03 (0.04)	97.1 (0.04)	99.2 (1.2)

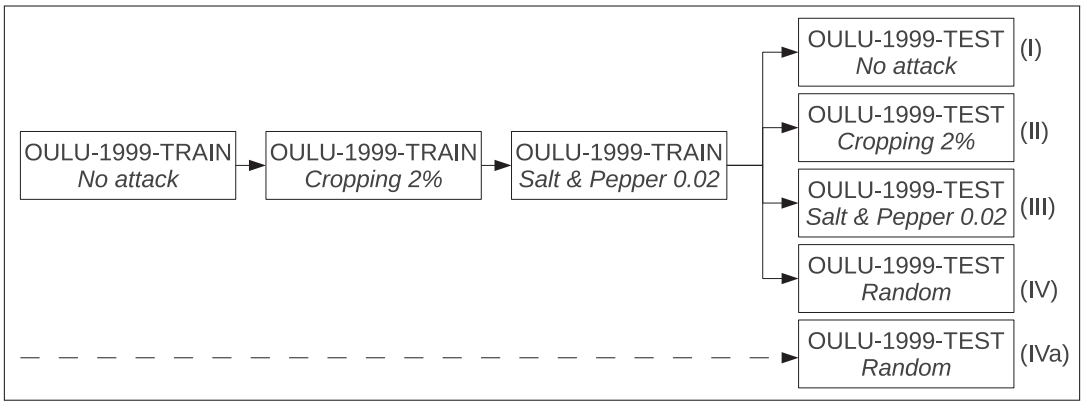


Figure 3.16 Memory adaptation experiment.

In this experiment, the GMM-based approach was first applied to the OULU-1999-TRAIN stream with no attack. Then, using the resulting memory as a starting point, the same approach was applied to the same stream but with cropping of 2%. Next, the same procedure was repeated (also using the previous memory as a starting point) but now with salt & pepper 0.02. Finally, the proposed approach was applied to the OULU-1999-TEST database in four different scenarios: using the memory of previous case as a starting point but now with (I) no attack; (II) cropping 2%; (III) salt & pepper 0.02; (IV) randomly chosen attacks (salt & pepper 0.02, no attack, cropping 2%) for each image; (IVa) not using previous memory (no learning) with random attacks. In all cases the confidence level was set to 0.8, as adaptation requires a more restrictive confidence level.

It is interesting to observe that the results obtained in the adaptation experiments (Table 3.7) are similar to previously presented results. The slight degradation in computational burden was mainly due to the more restrictive confidence level. For example, OULU-1999-TRAIN with no attack resulted in 92.9% decrease with confidence level 0.95 (Table 3.3) versus 84.8% with confidence level 0.8 (Table 3.7). However watermarking performance of both was very similar (Table 3.4). The same happened for the simulations involving cropping 2% and salt & pepper 0.02 (Tables 3.5 and 3.6). Regarding the OULU-1999-TEST stream, the computational performance of cases I, II, III and IV was close to that of no learning for the previous simulations (Tables 3.3 and 3.5) with an equivalent watermarking performance (Tables 3.4 and 3.6). It is worth noticing that in Table 3.7, for the random attacks, the use of a training sequence (IV)

resulted in a considerable decrease in computational burden when compared to no training (IVa). It is also worth noticing that the OULU-1999-TEST simulations with learning resulted few inserted probes when compared to OULU-1999-TRAIN simulations. This demonstrates that even in such a challenging scenario involving changes in the set of attacks, the proposed approach can learn how to adapt to such changes.

Table 3.7 Adaptation performance. DFE is the decrease in the number of fitness evaluations compared to full optimization, † is the $DRDM$, ‡ is the BCR robust, § is the BCR fragile. For all values, the mean μ and standard deviation σ per image are presented in the following form: $\mu(\sigma)$. $DRDM$ is presented with two decimal points and BCR is presented in percentage (%) with one decimal point.

Attack	Database	Re-optimizations	Inserted probes	DFE	†	‡	§
No attack	OULU-1999-TRAIN	13	3	84.8%	0 (0)	100 (0)	100 (0)
Cropping 2%	OULU-1999-TRAIN	13	3	84.3%	0.04 (0.05)	97 (3.6)	99.7 (1)
S&P 0.02	OULU-1999-TRAIN	12	1	79.4%	0.03 (0.04)	97.3 (3.6)	99.5 (1.2)
No attack (I)	OULU-1999-TEST	20	1	88.9%	0.01 (0.02)	99.9 (0.01)	99.9 (0.01)
Cropping 2% (II)	OULU-1999-TEST	15	2	91.4%	0.04 (0.05)	93.3 (0.06)	99.1 (0.02)
S&P 0.02 (III)	OULU-1999-TEST	29	5	87.4%	0.04 (0.04)	97.1 (3.7)	99.3 (1.1)
Random (IV)	OULU-1999-TEST	31	4	85.5%	0.03 (0.04)	97.3 (4.3)	99.4 (1.4)
Random (IVa)	OULU-1999-TEST	65	8	76.3%	0.03 (0.04)	97.6 (3.7)	99.6 (1)

3.5.4 Scenario B – optimization of homogeneous streams of bi-tonal images using memory-based DPSO versus full PSO

In general, for the homogeneous image streams, the computational burden performance of the GMM-based approach is slightly worse than what has been reported for the case-based approach in (Vellasques *et al.*, 2011) as it required more re-optimizations. Yet, adjusted for the size of the image streams, the number of re-optimizations for the GMM-based approach in this scenario is consistent with that obtained for the heterogeneous image streams while for the case-based approach, there is a huge discrepancy between the performances for the heterogeneous and homogeneous streams. That is, since a case-based probe is over-fit to a particular optimization problem, it tends to perform better than the GMM-based approach when the stream of optimization problems is homogeneous. In the GMM-based approach by its way, a probe is less biased to a specific optimization problem and can cope better with variations in a more heterogeneous image stream. The watermarking performance (mainly watermark

robustness) of the GMM-based approach is considerably better than that of the case-based approach.

3.5.5 Scenario C – optimization of unconstrained (homogeneous/heterogeneous) streams of bi-tonal images using memory-based DPSO versus full PSO

The behavior of the proposed technique when compared to case-based for scenario C was quite similar to that observed for scenario A. The proposed technique resulted in a decrease in computational burden at an equivalent watermarking performance. The use of a training sequence of images allowed a further decrease also with little impact on watermarking performance.

3.5.6 Discussion

The GMM-based approach was evaluated in three main scenarios – intelligent watermarking of homogeneous, heterogeneous image streams, and a mix of both, respectively. It is possible to observe through the simulation results that for the heterogeneous image streams, the proposed memory scheme results in less re-optimizations than the case-based scheme but at nearly the same watermarking performance. Both, the fidelity of the watermarked image and the detection rate of the robust and fragile watermarks are comparable to those of full optimization. The main reason is that by using particle history data, it is possible to sample a larger region of the fitness landscape but in a targeted manner. It can be said thus that the case-based mechanism is sensitive to the distribution of particles in the end of the optimization process. It was also observed that the proposed technique allows a significant decrease in computational burden when compared to full optimization in both, homogeneous and heterogeneous image streams. More specifically, the number of fitness evaluations per image was above 800 for the best scenario of Full Optimization which is unfeasible for practical applications as it involves more than 800 embedding and detection operations per image. This number was decreased to 67 in the worst case for the proposed approach with learning.

For the heterogeneous scenario, a memory fill up experiment was performed and it showed that as new images are fed into the system, the amount of novelty brought by these images decreases considerably for the first third of the image stream (OULU-1999-TRAIN) and then

stabilizes. Consequently, the lack of a proper memory management mechanism results in redundant probes which impair the computational performance of a unsuccessful recall (since all LTM probes need to be tested before re-optimization is triggered). At the same time, when insert operations are employed indiscriminately, the resulting memory becomes quite noneffective. Moreover, the probing capability of the memory is negatively affected as the diversity of sampling solutions decrease.

The adaptive memory management experiments involving heterogeneous streams showed that the proposed approach not only decreases the computational burden of intelligent watermarking (when compared to the case-based approach) but with practically no impact on watermarking performance. And more important than that, an analysis of memory dynamics showed that in the proposed mechanism, the memory space is used in a more effective manner as insert operations are employed sparingly. Moreover, it has been demonstrated that the frequency of memory update operations are in a par with the amount of novelty brought by the new problems. This is more in tune with the formulation of incremental learning seen in (Jain *et al.*, 2006) as with this combination of merge and insert operations (1) none of the inserted probes will contradict the data processed up to that point and (2) through the use of a merge operator each intermediate hypothesis is maintained as long as it is consistent with the data seen. That is, insert only occurs when the new problem represents new knowledge to the memory. These experiments also showed that by maintaining the distance between LTM probes high, it is possible to improve the diversity of sampled solutions which allows a better probing capability. Analysis of memory dynamics showed that the proposed memory management mechanism helps to avoid inserting probes that do not bring novelty to the LTM. For example, both the pairwise distance between probes and the minimum distance between new probes and probes in the memory are increased considerably when the memory management scheme is employed. This shows that the proposed scheme minimizes redundancy in the LTM. The sampling diversity was also increased which means that despite smaller memory and computational burden, the proposed memory management scheme resulted in probes that cover a significant area of the fitness landscape.

Memorization experiments demonstrated that the GMM memory can learn considerably well the stream of optimization problems. First because density estimate of solutions in the optimization space offer a reliable approximation of the fitness landscape and second because the merge operator results in less biased probes that generalize well to new problems, as observed in the experiments involving multiple recalls for a same image. These experiments also demonstrated that the probe is subject to a trade-off between memorization and generalization (bias/variance trade-off). This trade-off can be modified when necessary (e.g. in an application involving more dynamism in the stream of document images) by adjusting the confidence level of the change detection mechanism. And yet, memorization can be further improved (when necessary) by de-activating the merge operator (not recommended for heterogeneous streams).

It was possible to observe in experiments with higher cropping intensity and salt & pepper attack that the results observed for the cropping 1% and no attack are applicable to other types of removal attacks. The conclusion to be drawn here is that as long as robustness against a given attack can be attained through optimization of embedding parameters and considering that the stream of images contains recurrent (similar images), the proposed GMM-based approach is expected to result in a smaller computational burden compared to full optimization, with an equivalent watermarking performance. The reason is that the use of GMM results in a precise approximation of the stream of optimization problems. The limitation of the proposed approach is that its watermarking performance is bounded by the watermarking performance of full optimization. For example, in the baseline watermarking system, robustness against geometric attacks cannot be attained through manipulation of embedding parameters (instead, it is attained through the use of reference marks (Wu and Liu, 2004)). Therefore, the GMM-based approach also will not tackle robustness against such type of attack.

In the adaptation experiments, it was possible to observe that in applications involving high dynamism in the stream of problems (e.g. changing attacks), the proposed approach can adapt well, with a relatively small computational burden. The reason is that the memory of GMMs results in a more precise representation of the stream of optimization problems which allows a better change detection capability (as observed in the memorization experiments as well).

These experiments also allow us to draw some guidelines regarding the choice of confidence level. In situations involving high variability (like changing attacks), a more restrictive confidence level is to be preferred. Otherwise, a more relaxed confidence level is preferred (since it should result in less re-optimizations).

It was possible to observe that the GMM-based approach is not only less expensive than the case-based approach (for the heterogeneous streams) but the gains in computational burden are more consistent, that is, are quite similar across different scenarios. Another advantage of the GMM-based approach is that it has a smaller memory footprint than the case-based approach. Not only because the mixture model offers a more compact data representation but also because in the GMM-based approach, the number of probes is considerably smaller than for the case-based approach. It is important to mention that although the LTM size is limited for the GMM-based approach, such limit was not achieved for the chosen confidence level. It is worth mentioning that the decrease in the number of fitness evaluations is proportional to the number of probes, the number of re-sampled particles, the frequency of recall and the number of fitness evaluations required in full optimization. Since the number of fitness evaluations required in full optimization varies across the images in a stream the possible boost obtained by replacing full optimization by memory recall is image-dependent. It is also important noticing that for a limited memory size, the number of fitness evaluations in full optimization tends to be considerably larger than that of a successful recall. Therefore, the impact of a case of re-optimization in the number of fitness evaluations tends to be exacerbated in small databases.

In general these experiments show that by estimating mixture models of swarm solutions and keeping a memory of these models with the use an appropriate memory management strategy it is possible to build a general model of a stream of optimization problems in an intelligent watermarking application using a set of learning images and then decrease significantly the cost of intelligent watermarking with little impact on watermarking performance. This general model is more adaptive than that created by the case-based approach and is thus more appropriate for applications where the stream of images to be optimized is heterogeneous.

3.6 Conclusion

In this chapter an intelligent watermarking technique based on Dynamic Particle Swarm Optimization (DPSO) is proposed. The adaptive memory relies on sampled solutions from GMMs of previous optimization problems and their respective global best solutions in order to (1) compare how similar future optimization problems are to those previously seen and (2) provide alternative solutions in cases where the similarity between problems is small, avoiding re-optimization. Its memory management strategy aimed at tackling two main issues observed in previous experiments. The first was to avoid redundancy in the LTM while the second was to allow the memory to adapt quickly to new optimization problems.

Although the use of density models in evolutionary computing is not new, the use of models based on phenotypic and genotypic data of candidate solutions is novel. Moreover, while in the EDA literature most authors rely on high evaluation solutions in order to estimate these models, in the proposed approach we rely on all solutions in order to build a more comprehensive model of the fitness landscape. It was demonstrated empirically that this more comprehensive model allows a more precise match between previously seen and new optimization problems. Another contribution of the proposed technique was the inception of a management approach that allows the memory to incrementally learn new trends on the stream of optimization problems while limiting memory footprint.

Experimental results demonstrate that replacing memory solutions by density estimates of swarm solutions result not only in less memory burden but in a more precise probing mechanism which resulted in a decrease in the number of re-optimizations with little impact in watermarking performance. Since the proposed approach allows an incremental learning of optimization problems, the use of a learning stream of images allowed decreasing computational cost while improving precision altogether. In such case, a decrease of 97.7% in the number of fitness evaluations was obtained for heterogeneous image streams (when compared to full optimization) through the use of a learning stream of images. Such improvement in computational performance was higher than that of no learning. It was also possible to observe that the GMM memory allows a more precise representation of the fitness landscape. This results in

better probing of the fitness landscape (compared to a memory of static solutions) which helps to avoid false positive errors (recalling wrong probes which would decrease the watermarking performance). Such memory makes possible for example, changing the attack employed on the DPSO module, without any further need of human intervention in what regards memory management.

As a future work we propose a deeper study on each of the main modules of the proposed technique and a comparison study with alternative approaches for these modules. We also propose validating the GMM-based approach using a larger image stream.

3.7 Discussion

In this chapter we proposed a hybrid GMM/DPSO approach aimed at the intelligent watermarking of heterogeneous streams of document images. Such approach provides a more precise (but compact) representation of the fitness landscape. Moreover, we introduced a specialized memory management mechanism which allows the memory to adapt to variations in the stream of optimization problems. For this reason, the proposed technique resulted in a considerable decrease in terms of computational burden for heterogeneous streams of document images when compared to the approach of Chapter 2.

However, it is important to observe that the decrease in computational burden obtained by replacing re-optimizations with memory recall is constrained by the frequency of re-optimization. And in our stream of optimization problem formulation of intelligent watermarking, the frequency of re-optimization is application-dependent. Therefore, in a less constrained environment (e.g. changing attacks), as re-optimization becomes more frequent, the decrease in computational cost obtained by memory recall becomes less important since re-optimization is much more expensive. In the next chapter we propose using previously learned GMMs in order to replace costly fitness evaluations during re-optimization with Gaussian Mixture Regression (GMR) in a strategy named surrogate-based optimization. To this end, we investigate strategies to assign promising GMMs to new problems, perform regression on GMMs, update them on-line and control the quality of the predictions.

CHAPTER 4

DS-DPSO: A DUAL SURROGATE APPROACH FOR INTELLIGENT WATERMARKING OF BI-TONAL DOCUMENT IMAGE STREAMS

In this chapter we propose a dual surrogate approach which employs the memory of GMMs in regression mode in order to decrease the cost of re-optimization when novel problem instances occur. The goal of the proposed approach is to decrease the cost of re-optimization of the approach described in Chapter II in situations involving a high variability in the stream of optimization problems. In scenarios like changing sets of attacks, re-optimization tends to be triggered more often. Decreasing that specific cost becomes a relevant issue. In the proposed approach, GMMs are assigned to new problems and then, costly fitness evaluations are replaced with Gaussian Mixture Regression (GMR). Simulation results in scenarios involving high variation in the stream of problems (changing attacks) demonstrate that the proposed approach allows a decrease of up to 36% in the computational burden compared to the approach described in the previous chapter. The content of this chapter was submitted to Applied Soft Computing (Vellasques *et al.*, 2012c).

4.1 Introduction

The decreasing costs of data transmission and storage provided numerous opportunities for sharing multimedia documents like images. This has led to the creation of a digital economy with new services that are available 24 hours a day, 7 days a week, around the globe. Individuals and businesses depend more and more on sharing important documents which raises serious privacy concerns. Enforcing the security of document images is an important issue. Cryptography can solve part of this issue. However, specially with multimedia documents like images, the protection allowed by cryptography vanishes as the data has been decrypted. Digital watermarking (Cox *et al.*, 2002) which consists of embedding image-related secret data through the manipulation of pixel values in an imperceptible manner, allows another layer of protection. Most importantly, the protection mechanism provided by digital watermarking follows the image even when it is inadvertently distributed or tampered. Enforcing the security of bi-

tonal document images poses an additional challenge as bi-tonal images have lower embedding capacity and the manipulation of bi-tonal pixels is more prone to result in visual artifacts.

Digital watermarking has become an active area of research in recent years. Because of its nature, watermarking systems are subject to attacks by hackers (Voloshynovskiy *et al.*, 2001). Robustness against attacks always comes at the cost of degradation on imperceptibility (Cox *et al.*, 1996). Many watermarking techniques allow adjusting the trade-off between robustness and quality through the manipulation of embedding parameters. The optimal trade-off and the corresponding values vary from one image to another. To make matters worse, security requirements also vary across applications. Adjusting these parameters manually is infeasible in practical applications and evolutionary computing (EC) techniques such as Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995) and Genetic Algorithms (Holland, 1992) have been employed in order to find embedding parameters that optimize the trade-off between image quality and watermark robustness for each image and set of attacks (Vellasques *et al.*, 2010a). In EC, a population of candidate solutions is evolved through a certain number of generations, and guided by an objective function. In intelligent watermarking (IW), objective functions are usually a combination of image quality and watermark robustness. The fitness of each candidate solution is evaluated at each generation. Each fitness evaluation requires one or more embedding, detection and attack (image processing) operations which is prohibitively expensive in industrial applications.

Recent efforts to decrease the computational cost of IW techniques for streams of document images is promising. In the Dynamic PSO (DPSO) system proposed in (Vellasques *et al.*, 2011), IW of homogeneous streams of bi-tonal document images (or problems) was formulated as a special type of dynamic optimization problem (DOP[1]). In this special formulation of DOP, a stream of document images corresponds to a stream of recurring optimization problems. A change detection mechanism assigns case-based solutions of previously-seen problem instances (associated with previous document images) to new similar problem instances (associated with new images). This significantly reduced the number of costly re-optimization operations, allowing for a significant decrease in computational burden. In the DPSO system

[1] In a DOP, the optimum location and/or fitness value change over time.

proposed in (Vellasques *et al.*, 2012b), Gaussian mixture modeling (GMM) of optimization history was employed in order to represent a model previous optimization problems. This approach allowed for a significant decrease in the cost for IW of **heterogeneous** streams of document images compared to the case-based approach. In both approaches, when a new optimization problem is similar to a previously-solved one, solutions in memory corresponding to that previous problem should be recalled, avoiding a costly re-optimization process.

The basic assumption behind that approach is that after a learning phase, most new problem instances will result in recall rather than re-optimization operations. However, a significant variation in the stream of optimization problems such as that caused by a new attack, will result in an increase in the number of re-optimization operations. The time complexity of re-optimization is orders of magnitude higher than that of recall. Each attempt of recalling a memory element has a time complexity comparable to a single iteration in the optimization phase, and optimization generally requires generally 50 plus iterations. Decreasing this cost is an important issue. It has been demonstrated in literature that optimization strategies based on the use of an associative memory (Yang and Yao, 2008) outperform other dynamic optimization strategies in cyclic/recurrent problems. These techniques rely on storage of high performance solutions, as well as information about their fitness landscape using a density estimate. The most common approach to associative memory optimization is to inject memory solutions in the initial population, in a strategy named memory-based immigrants (Wang *et al.*, 2007).

One limitation of approaches based on associative memory is that for a case of re-optimization, the density estimates will only provide an initial set of candidate solutions. After that, these solutions are evolved with the use of EC and the knowledge of previous problems provided by that estimate is not explored during the optimization process whatsoever. It has been observed in the Estimation of Distribution Algorithms (EDA) literature that probabilistic models can be employed in order to guide the optimization process (Pelikan *et al.*, 2002). A second limitation is that although memory-based immigrants can reduce the number of generations needed for convergence, still, each generation involves re-evaluating the fitness of each solution.

In surrogate-based optimization, costly fitness evaluation operations are replaced by a regression model. Sampling, model update and optimization are applied in an iterative manner. The advantage of such approach is that most of the fitness evaluations required for optimizations are performed using a regression model at a fraction of the cost of an exact fitness evaluation. There are two schools of thought: a first one that sees a surrogate as an oracle that will replace the objective function (Queipo *et al.*, 2005) and a second one that sees a surrogate as a compact database employed in order to forecast good solutions during optimization, accelerating convergence (Parno *et al.*, 2011). Both provide different ways of addressing the trade-off between model precision and fitness evaluation cost. The first one favors decreasing fitness evaluation over precision and is preferred in situations where the model provides a precise representation of the fitness landscape and/or the computational cost of fitness evaluation is too high. The second one favors precision over decreasing fitness evaluations and is preferred in situations where the model does not provide a precise representation of the fitness landscape and/or the cost of optimization is not too high. Surrogate-based optimization involves a mixed use of exact and predicted fitness values which leads to a trade-off between increase in model precision and decrease in computational cost – a more precise model makes possible relying less on expensive exact fitness evaluations but improving model precision involves probing the exact fitness landscape which is computationally expensive.

It is important distinguishing between EDA and surrogate-based optimization. In each generation of EDA, solutions are sampled from a probabilistic model, re-evaluated and the best solutions are employed in order to update the model. In contrast, surrogate-based optimization builds a sampling plan in the parameter space. Then, numerical simulations are performed at the sampled locations, followed by model update and optimization. However, optimization is based on fitness values predicted by the model. This is the main advantage of surrogate-base optimization compared to EDA. In EDA the model guides the search process while in surrogate-based optimization, the model provides a mean of replacing expensive exact fitness evaluations with cheaper approximated fitness values obtained through regression.

In this chapter, a novel approach called Dual Surrogate Dynamic PSO (DS-DPSO) is proposed in which models of previous optimization are employed as surrogates in order to decrease the computational burden associated with full re-optimization for a hybrid GMM/DPSO system (proposed in (Vellasques *et al.*, 2012b)). This system performs four different levels of search for solutions with increasing computational burden and precision. As in previous research, levels 1 and 2 first attempt to recall ready-to-use solutions from a memory of GMMs. If embedding parameters require a significant adaptation, the optimization modules are activated in levels 3 and 4. Whenever re-optimization is triggered, an attempt to optimize the embedding parameters using the surrogate as an oracle is performed level 3. This allows for a substantial decrease in the number of fitness evaluations as the optimization process is performed mostly on a GMM regression model. If it fails, a new attempt is performed, this time using the surrogate as a database in order to accelerate convergence in level 4. This second optimization stage relies mostly on exact fitness evaluations (the surrogate is employed on a best-case basis) and, for this reason, provides a safeguard to the whole system for situations where the surrogate model recovered from memory at level 3 does not correspond to the new problem. The main advantage of this DS-DPSO strategy is that it tackles the precision/cost trade-off by relying on a memory of previous surrogates and employing two different surrogate-based optimization strategies in sequence – level 3 with smaller cost and smaller precision (but which should provide good results for situations where the model shares some similarity with the new problem), and level 4 with higher cost and precision which provides a safeguard to the whole system and allows building new surrogates (for cases of novel problem instances).

This research attempts to exploit a memory of GMMs (learned for a stream of reference training images) to decrease the computational cost of full re-optimization. In addition, incorporating knowledge about a new optimization problem into the model of a previous problem is considered in order to produce a model with a better fit to the new problem. It is assumed that whenever re-optimization is triggered, the best fit model should provide a good starting point for building a surrogate for the new problem. Regardless, it should decrease the computational burden of optimization by accelerating convergence. The proposed approach is validated empirically with the use of heterogeneous streams of bi-tonal document images. The University

of Oulu's MediaTeam (Sauvola and Kauniskangas, 1999) dataset and samples of the Computer Vision and Image Understanding (CVIU) Journal are employed for this purpose. For each simulation, watermarking performance and number of fitness evaluations are reported.

A formulation of optimization problems in digital watermarking is provided in Section 4.2. A literature review of surrogate-based optimization is presented in Section 4.3. The proposed method named DS-DPSO is presented in Section 4.4. The experimental methodology and simulation results are presented and discussed in Sections 4.5 and 4.6.

4.2 Particle swarm optimization of embedding parameters

In the given formulation of IW, a stream of document images corresponds to a stream of optimization problems, where some problems may reappear over time. A PSO-based system proposed by the authors in a previous research (Vellasques *et al.*, 2011) allows for the optimization of embedding parameters of multiple watermarks with different levels of robustness into a given bi-tonal image of a stream. During embedding, (1) the bi-tonal images is partitioned in blocks of equal size, (2) a flippability analysis is performed in order to evaluate the visual impact of flipping each pixel, and (3) the pixels are shuffled in order to distribute flippable pixels evenly across the image. Then, (4) each message bit is embedded on each block by manipulating the quantized number of black pixels on that block and finally, and (5) the image is de-shuffled. Detection involves partitioning the image, shuffling using the same key used on embedding, and counting the quantized number of black pixels on each block in order to detect each message bit. Four different parameters can be adjusted in order to modify the trade-off between watermark robustness and image quality for a given image during embedding, namely: quantization step size (Q), size of the window employed in the flippability analysis (W), block width (B) and shuffling key index (S). Readers are referred to (Vellasques *et al.*, 2011) for more information on the bi-tonal watermarking system.

In our formulation of the the optimization problem for digital watermarking, two watermarks – a fragile one with $Q_F = 2$ and a robust one with $Q_R = Q_F + \Delta Q$ – are embedded into the cover image **Co** where ΔQ is the difference between the robust and fragile quantization step

sizes. The fitness function comprises robustness and quality metrics, aggregated with the use of the Chebyshev technique (Collette and Siarry, 2008):

$$F(\mathbf{x}) = max_{i=1,..,3}\{(1-\omega_1)(\alpha_s DRDM - r_1), (1-\omega_2)(1-BCR_R - r_2), (1-\omega_3)(1-BCR_F - r_3)\} \tag{4.1}$$

where α_s is the scaling factor of the quality measurement $DRDM$ (Distance Reciprocal Distortion Measure (Lu *et al.*, 2004)), BCR_R (Bit Correct Ratio (Areef *et al.*, 2005; Pan *et al.*, 2004) between embedded and detected watermark) is the robustness measurement of the robust watermark, BCR_F is the robustness measurement of the fragile watermark, ω_i is the weight of the i^{th} objective with $\omega_i = \frac{1}{3}, \forall\ i$, r_i is the reference point of objective i. It is important to note that (unlike the BCR_F and $DRDM$) BCR_R is computed after an attack has been applied. The fitness function is depicted in Figure 4.1 where **Co** is the cover image, $\mathbf{m}_R$ and $\mathbf{m}_F$ are the robust and fragile watermarks, respectively, **Cr** is the robust watermarked image, **Crf** is the image that has been watermarked with both, the robust and the fragile watermarks (multi-level watermarked image), **Crf′** is the multi-level watermarked/attacked image, $\mathbf{m}_{RAD}$ is the robust watermark that has been detected from the multi-level watermarked/attacked image, $\mathbf{m}_{FD}$ is the fragile watermark that has been detected from the multi-level watermarked image.

A diversity preserving PSO (Kapp *et al.*, 2011) has been employed in order to optimize the embedding parameters based on the fitness function described above (Vellasques *et al.*, 2011). Therefore, the fitness landscape comprises five dimensions: four in the parameter space (more formally, $\{x_1, x_2, x_3, x_4\} = \{\Delta Q, W, B, S\}$) and one in the fitness space ($f(\boldsymbol{x}),\ \forall\ \{\boldsymbol{x} \in \mathbb{R}^4\}$). Readers are referred to (Vellasques *et al.*, 2011, 2012b) for more information about the optimization problem formulation of digital watermarking.

One of the limitations regarding the use of EC on digital watermarking is that each fitness evaluation requires multiple embedding, detection and attack operations which are very costly. In the given application, a fitness evaluation involves numerous embedding operations, each of which has time complexity $O(|\mathbf{Co}| \cdot log(|\mathbf{Co}|))$ where $|\mathbf{Co}|$ is the number of pixels on cover image **Co** (with a magnitude of 10^6) while performing regression on a GMM has a time complexity of $O(K \times d)$ where K is the number of components in the GMM (usually between 5

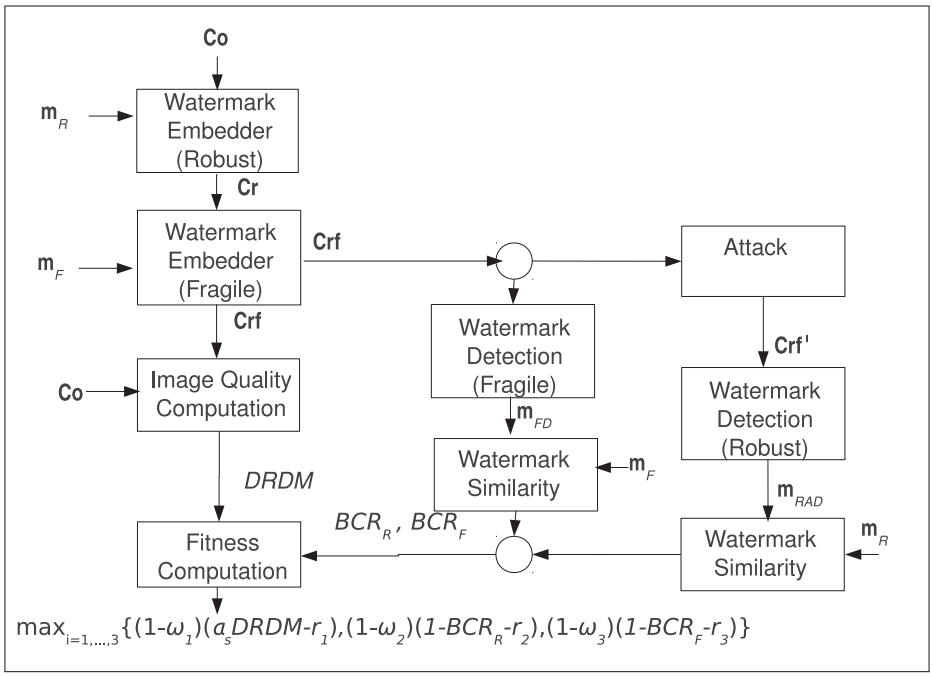


Figure 4.1 Fitness evaluation module.

and 20 for this case) and d is the dimension of the parameter space which is 4. Therefore, the cost of optimizing this bi-tonal watermarking system can be significantly reduced by replacing part of the exact fitness evaluations with regression models.

4.3 Surrogate-based optimization

Surrogate (or model-based optimization) allows tackling the computational burden of fitness evaluation in complex real-world problems. As stated by Queipo *et al.* (Queipo *et al.*, 2005), a surrogate model can be seen as a non-linear inverse problem in which one aims to determine a continuous function $f(\boldsymbol{x}),\ \forall\ \{\boldsymbol{x} \in \mathbb{R}^d\}$ of a set of design variables from a limited amount of available data $\boldsymbol{f} = \{f(\boldsymbol{x}_1), ..., f(\boldsymbol{x}_N)\}$ where $\boldsymbol{x}_i$ is a design variable, d is the dimensionality of the parameter space and N is the number of data points. During optimization, costly calls to $f(\boldsymbol{x})$ are partially replaced by a predicted value $f_P(\boldsymbol{x}, \Theta)$

$$f_P(\boldsymbol{x}, \Theta) = \hat{f}(\boldsymbol{x}, \Theta) - \rho_c \varepsilon(\boldsymbol{x}, \Theta) \tag{4.2}$$

of $f(\boldsymbol{x})$ (Torczon and Trosset, 1998) where $\hat{f}(\boldsymbol{x}, \Theta)$ is an approximation to $f(\boldsymbol{x})$ based on model Θ, ρ_c is a constant that dictates how much emphasis will be put in exploring unknown regions of the model and $\varepsilon(\boldsymbol{x})$ is the prediction error. The basic approach to surrogate modeling assumes that Θ is unknown and then iteratively selects a set of design variables using stratified sampling – also known as design of experiments (DOE) to perform numerical simulations using this set and update the model. Once a given stop criterion has been achieved, the model is validated against $f(\boldsymbol{x})$. Once a good representation of $f(\boldsymbol{x})$ has been obtained, optimization is first performed using $f_P(\boldsymbol{x}, \Theta)$, one or more solutions are re-evaluated on $f(\boldsymbol{x})$ and then, either Θ is refined using new data points or the process is halted in the case that a convergence criterion has been met.

A surrogate model can be either global or local (Dennis and Torczon, 1995). A local model provides a detailed approximation of a specific region of the fitness landscape while a global model provides a general approximation of the whole optimization problem. There are four different strategies to build a surrogate (Praveen and Duvigneau, 2007): (1) data-fitting models, where the approximation is constructed using available data; (2) variable convergence model, where the approximation is based on the numerical solution of a partial differential equation (PDE); (3) variable resolution models where the search space is discretized with the use of a hierarchy of grids; (4) variable fidelity models, where a hierarchy of physical models is employed in order to approximate the fitness function.

Most of the techniques found in the literature rely on data-fitting models. The advantage of such type of approach is that it uses pattern recognition methods such as radial basis functions, clustering, multilayer perceptron, polynomial fitting, Gaussian processes, support vector machines (Shi and Rasheed, 2008) which can be inferred even when domain knowledge is ill-defined (such as IW of stream of document images). Data-fitting approaches can be either off-line or on-line (Praveen and Duvigneau, 2007). An off-line surrogate is first trained with a set of data points that have been evaluated in the exact fitness function, is assumed to be an accurate representation of the exact function and is indicated in situations where computational burden is more important than precision. In contrast, an on-line surrogate is trained

incrementally, closely integrated into the optimization method and is indicated in situations where precision is more important than computational burden.

One of the main issues with surrogate-based optimization is that it is generally difficult to obtain a model with sufficient approximation accuracy due to the lack of data and/or high dimensionality which leads to models with high approximation errors that commonly result in false optima during the optimization phase (Jin *et al.*, 2002). This is an important issue for on-line surrogates since the construction of a good surrogate requires an experimental design which is space-filling in order to capture the essential trends of the fitness landscape. Yet the goal of optimization is to generate points which lead to improvements in the fitness function (El-Beltagy *et al.*, 1999). A surrogate model is therefore subject to a trade-off between decrease in computational burden and model fidelity. This issue can be partially alleviated with the use of evolution control, data selection (Jin *et al.*, 2002), combined local and global models (Zhou *et al.*, 2007), archive of solutions (case-based surrogate) (Fonseca *et al.*, 2009) and incremental stratified sampling (Yan and Minsker, 2011).

Using evolution control, part of the solutions obtained through surrogate optimization are validated against the exact (but costly) fitness function. It provides a mean of avoiding false convergence (Gräning *et al.*, 2005). Since the model provides an approximation of the real problem, it is expected to contain false optima (as observed in (El-Beltagy *et al.*, 1999), in the course of model update, false optima are likely to appear and disappear). Evolution control is subject to a trade-off between false convergence avoidance and computational burden – employing more solutions decreases the risk of false convergence but at a higher computational burden.

In data selection, the samples that will be employed to update the model are selected in order to improve the cost/benefit of model update in situations where such cost is high. Combining global and local models allows using a coarser level of fidelity to tackle exploration and a finer one to tackle exploitation in EC. The use of case-based models in EC can result in poor generalization and imply in high computational burden (in the long term) when compared to density estimates. Incremental stratified sampling allows a better space-filling of the fitness landscape

since it accounts for previously sampled realizations. However, it has been demonstrated in (Vellasques *et al.*, 2012b) that it is possible to obtain a good space-filling of the fitness landscape by combining a diversity preserving mechanism in EC with a model that is incrementally trained over a representative set of optimization problems.

4.4 A dual-surrogate DPSO approach for fast intelligent watermarking

4.4.1 System overview

Figure 4.2 illustrates the DS-DPSO approach. It has four levels with increasing computational cost and fidelity and at each level, an attempt to solve the watermarking problem for current cover image ($\boldsymbol{Co}_i$) is made. If it fails, a next attempt is made, in an upper level with an increased fidelity but at higher computational cost. The first two levels comprise memory recall. The third and fourth are the off-line and on-line optimization levels, respectively. The assumption is that in a situation involving recurrent problems, after a training phase, the memory will be precise enough thus most of the images should result either in a recall to the Short Term Memory (STM) or to the Long Term Memory (LTM). If recall fails, but the memory still provides a good approximation to the given problem, an attempt to optimize the parameters using an off-line surrogate is attempted. If this attempt also fails, then the costlier on-line surrogate is activated.

Figure 4.3 depicts the two recall levels. The STM (represented as $\mathfrak{M}_S$) contains the best solution ($\boldsymbol{p}_{g,S}$) plus a GMM approximation (Θ_S) of the fitness landscape of a single image ($\boldsymbol{Co_S}$). This combination of global best and GMM is called a **probe**. The LTM (represented as $\mathfrak{M}$) contains $|\mathfrak{M}|$ probes. Each probe contains a mixture model (Θ_i) obtained during the optimization of several different images and a global best solution ($\boldsymbol{p}_{g,i}$). LTM probes are sorted in reverse order of their number of successful recalls. It is important to mention that **phenotypic** and **genotypic** data of all solutions found during the optimization of a given image are employed in order to train a GMM.

During STM recall, $\boldsymbol{p}_{g,S}$ plus a set of solutions re-sampled from Θ_S are re-evaluated on image $\boldsymbol{Co}_i$. Then, the similarity between the distribution of the sampled and re-evaluated fitness values is computed with the use of the Kolmogorov-Smirnov (KS) statistical test (NIST/SE-

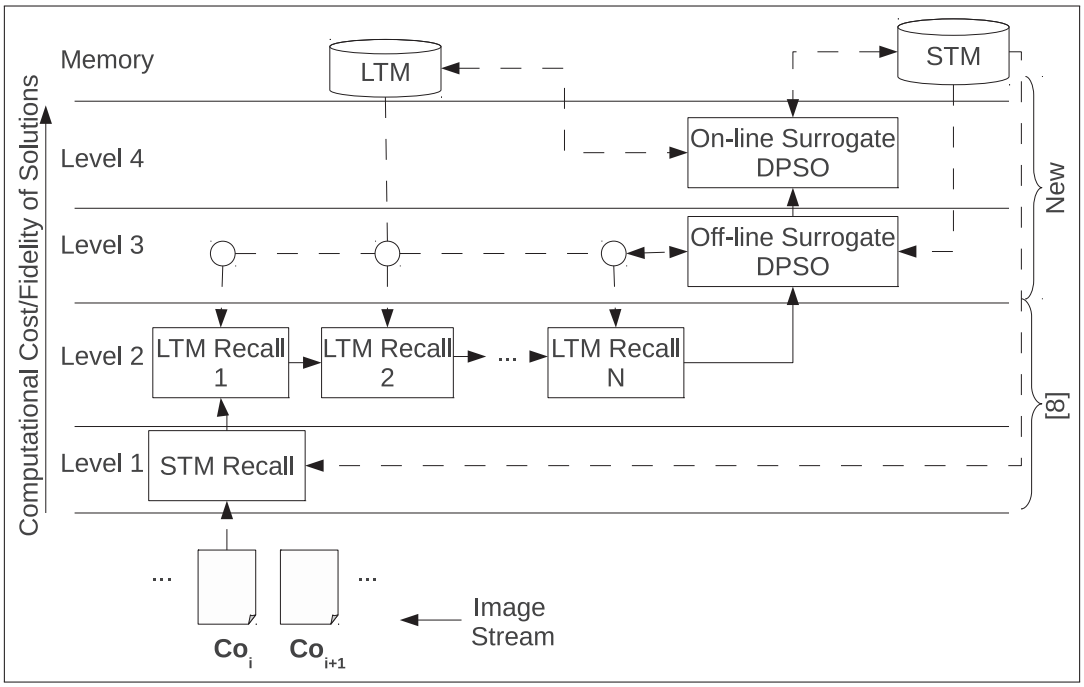


Figure 4.2 Overview of the proposed DS-DPSO technique for intelligent watermarking of document image streams.

MATECH, 2010). If the KS value is below a critical value for a confidence level α_{Crit}, it means that both distributions are similar and the best re-evaluated solutions is employed directly. Otherwise, a change is considered to have occurred in the landscape (compared to that of Θ_S) and level 2 is activated. In level 2, the same process is repeated for each LTM probe until either a case of KS value below the critical value has occurred or all probes have been tested.

Figure 4.4 depicts the first optimization level (off-line surrogate). The underlying principle is that for some failed recall attempts, the best GMM (the one that resulted in the smallest KS value during recall attempts) will already provide a good approximation of the fitness landscape and the re-evaluated fitness values (necessary during the recall) allow improving its fidelity for the new problem (the proposed approach is based on data-fitting model). Therefore, the DPSO technique described in (Kapp *et al.*, 2011) is employed in order to optimize the embedding parameters, but using the best GMM as surrogate most of the time (which has a smaller computational burden than an on-line surrogate approach). The GMM is employed in regression mode, an approach named Gaussian Mixture Regression (GMR) (Sung, 2004).

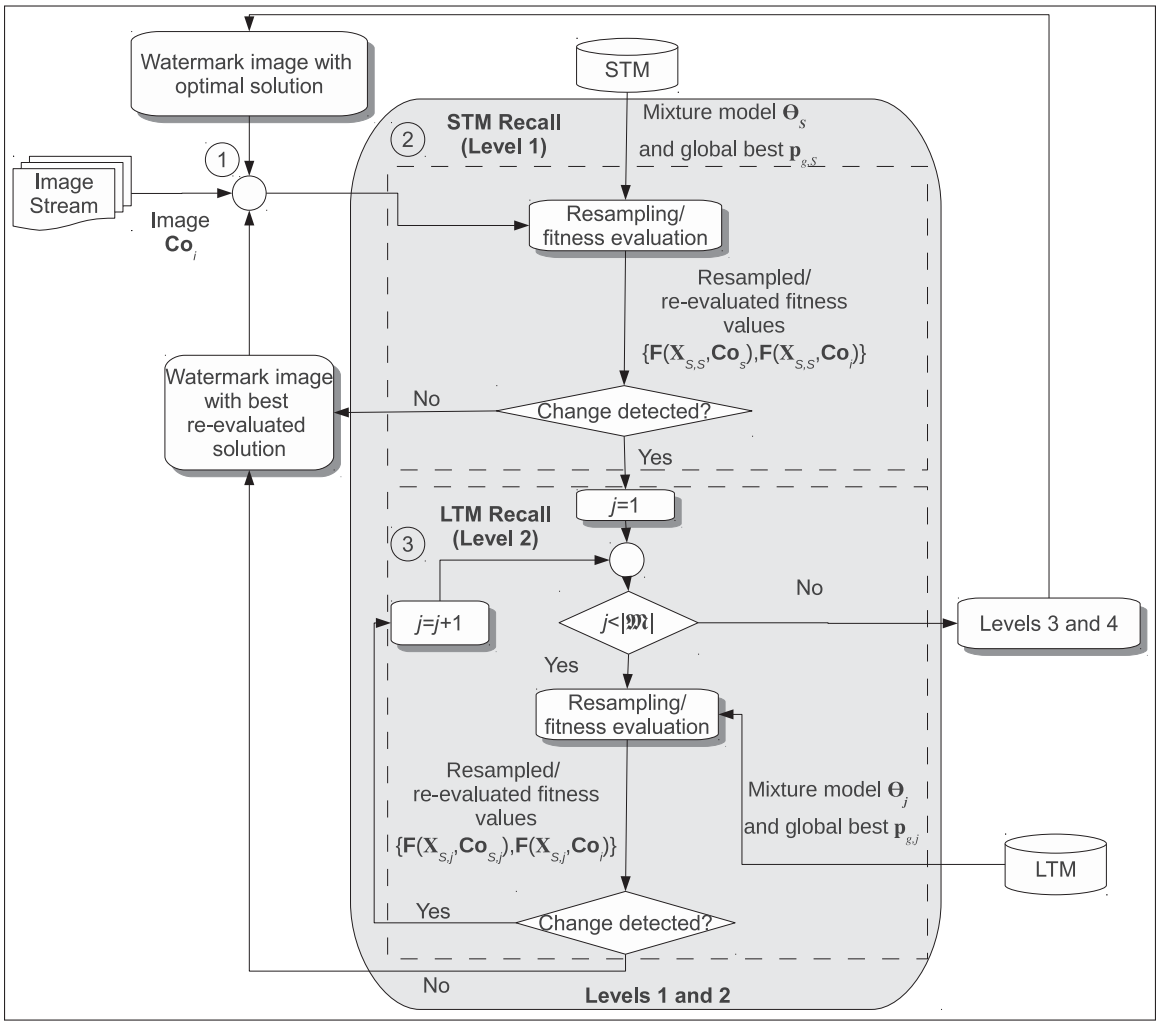


Figure 4.3 Flowchart diagram detailing the recall modules. Anchor points are employed in order to guide the reader. For each image in a stream of document images (step 1), an attempt to recall the STM is performed first (step 2) followed by an attempt to recall LTM, if necessary (step 3).

The surrogate is initialized with the mixture model that resulted in the smallest KS value and updated with re-evaluated solutions obtained during recall. Then, while the stopping criterion has not been reached (for all cases of re-optimization, we propose stopping optimization if global best has not improved for a certain number of generations (Zielinski and Laur, 2007)), an iteration is performed on the surrogate function (swarm $\mathbf{X}_A$), swarm solutions are re-evaluated in the exact function in order to avoid false optima (evolution control) and these solutions are employed in order to update the model. After that, if the surrogate improved the best recalled

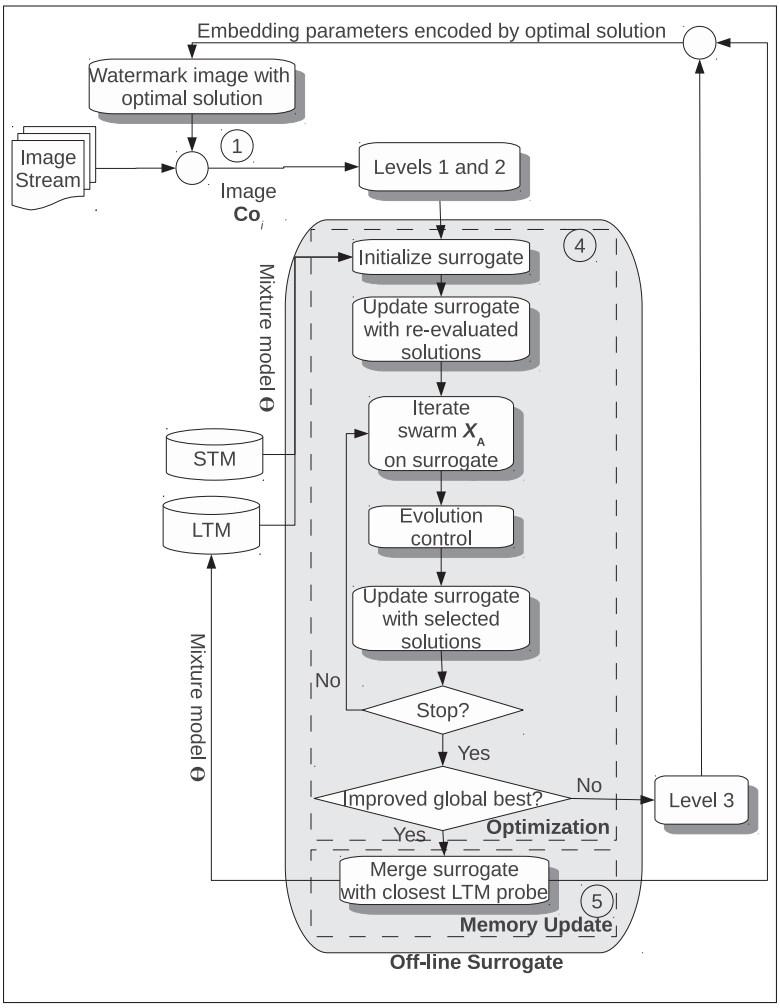


Figure 4.4 Flowchart diagram detailing level 3. Anchor points are employed in order to guide the reader. Whenever leves 1 and 2 fail, optimization is performed primarily on the surrogate (step 4). After that, the LTM is updated with the GMM employed on optimization (step 5).

solution, the most similar LTM probe is updated with the surrogate and the surrogate solution that resulted in the best fitness in the exact function is employed on $\boldsymbol{Co}_i$. Otherwise, level 4 is activated.

Figure 4.5 depicts level 4. Here, DPSO will be performed using primarily the exact fitness function, at a higher computational burden than that of the third level, but still with the possibility of a decreased computational burden compared to full optimization (depending on how fast convergence occurs). Solutions are re-sampled from the probe that resulted in the smallest KS

value and injected into the exact function swarm ($\mathbf{X}_B$). Optimization is performed on the surrogate function until a stop criterion has been reached. Then, the best solution is re-evaluated on the exact fitness function and injected into swarm $\mathbf{X}_B$ if it improves a corresponding neighbor. After that, an iteration is performed on the exact fitness function (swarm $\mathbf{X}_B$). When a stop criterion has been reached, a new mixture model is estimated, the best solution and mixture model are added to the STM, replacing the previous STM probe. If the new probe is similar to a given LTM probe, it is merged with that probe. Otherwise it is inserted (the probe with smallest number of successful recalls is deleted if memory limit has been reached).
Such approach allows tackling the optimization of recurrent problems as a machine learning problem. The surrogate might be de-activated in a training environment, where the constraints on computational cost are less severe in order to obtain a high fidelity representation of a given dynamic optimization problem. This should result in a model with good space filling properties, specially because the DPSO technique employed has a diversity preserving capability (Clerc, 2006). Then, the surrogate can be re-activated and the models obtained during training can be employed in order to perform large scale dynamic optimization of recurrent problems in a production (test) environment where computational burden constraints are more severe. This should minimize the issues of high approximation errors and false optima, specially early in optimization, since the on-line surrogate provides a safeguard for the whole system.

The STM/LTM recall (Figure 4.3) and update mechanisms (memory update box in Figures 4.4 and 4.5) are described with details in (Vellasques *et al.*, 2012b). Next, we present in details the key elements of the proposed approach, namely on-line update of GMMs, Gaussian Mixture Regression (GMR), evolution control. Then, we present the off-line and on-line surrogate DPSO modules and how both are integrated.

4.4.2 STM and LTM recall

For every new image $\boldsymbol{Co}_i$ (see 1 in Figure 4.3), an attempt to recall the STM probe is conducted at first. If it fails, the same process is conducted for each LTM probe until either all probes have been tested or a successful recall has occurred. Each recall attempt requires sampling from the respective GMMs. Sampling N_s solutions from a mixture of Gaussians comprises a linear

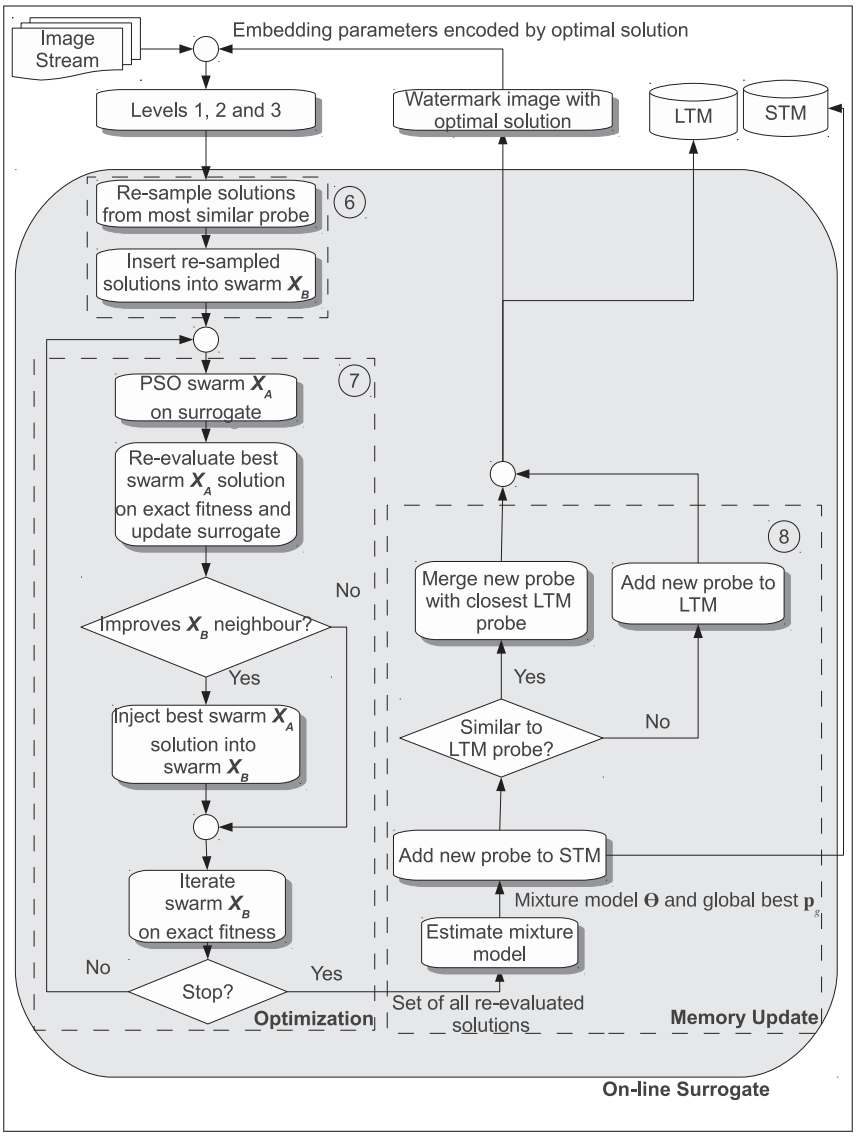


Figure 4.5 Flowchart diagram detailing level 4. Anchor points are employed in order to guide the reader. Whenever level 3 fails, solutions are re-sampled from the most similar probe (step 6) and then, optimization is performed using two different swarms, one for the the exact fitness and another one for the surrogate (step 7). After that, the memory is updated using the optimization history of the swarm employed to optimize the exact fitness (step 8).

combination between a random vector and the eigen-decomposition of the covariance matrix, centered at the mean vector:

$$\boldsymbol{X}_s = \boldsymbol{\mu}_j + \boldsymbol{\Lambda}_j^{\frac{1}{2}} \boldsymbol{U}_j \boldsymbol{R}_s \tag{4.3}$$

where $\boldsymbol{X}_s$ is a sampled solution, s is the index of a solution sampled for the component j in the mixture ($N_s\alpha_j$ solutions are sampled per component, where α_j is the mixing weight of the j^{th} component), $\boldsymbol{\Lambda}_j$ and $\boldsymbol{U}_j$ are the eigen-decomposition of $\boldsymbol{\Sigma}_j$ ($\boldsymbol{\Sigma}_j = \boldsymbol{U}_j\boldsymbol{\Lambda}_j\boldsymbol{U}_j^{-1}$) and $\boldsymbol{R}_s$ is a vector with the same length as $\boldsymbol{\mu}_j$ whose elements are sampled from a normal distribution $N(0, \mathbf{I})$, being $\mathbf{I}$ the identity matrix.

During a STM recall (see 2 in Figure 4.3) solutions are re-sampled from the STM mixture model (Θ_S) and re-evaluated in the new image (along with the global best $\boldsymbol{p}_{g,S}$). The distribution of the sampled set of fitness values $\boldsymbol{F}(\boldsymbol{X}_{S,S}, \boldsymbol{Co}_S)$ is compared against the distribution of re-evaluated fitness values $\boldsymbol{F}(\boldsymbol{X}_{S,S}, \boldsymbol{Co}_i)$ with the use of the Kolmogorov-Smirnov statistical test (KS). If the KS value between both distributions is smaller than a critical value for a confidence level α_{Crit}, the watermarking parameters encoded by the best recalled solution are employed right away for $\boldsymbol{Co}_i$, avoiding a costly optimization operation.

Otherwise (see 3 in Figure 4.3), the same process is repeated for each mixture Θ_j and global best $\boldsymbol{p}_{g,j}$ in the LTM – re-sampling, re-evaluating the re-sampled and global best solutions on the new image and comparing re-sampled fitness values $\boldsymbol{F}(\boldsymbol{X}_{S,j}, \boldsymbol{Co}_S)$ against the re-evaluated values $\boldsymbol{F}(\boldsymbol{X}_{S,j}, \boldsymbol{Co}_S)$ using the KS test. This process is repeated until a case of KS value smaller than the critical value occurs or all probes have been tested. The STM/LTM recall is described more carefully in (Vellasques *et al.*, 2012b).

STM/LTM recall (levels 1 and 2) is expected to be enough in most of the cases (specially for stable problem streams). However, when optimization is triggered too often (in situations involving high variability in the problem stream) the cost of re-optimization becomes a serious issue since a single re-optimization operation is several times more expensive than a recall. Next we propose a strategy to decrease this cost based on knowledge obtained on previous cases of re-optimization.

4.4.3 Off-line/on-line surrogate PSO

Rather than focusing on the level of detail provided by the model (global or local) we will focus in the fidelity/computational burden trade-off. The reason different levels of model fidelity are

employed in the literature is that it is assumed that a model has to be trained from scratch for each new problem. Therefore, the exploration/exploitation has to be addressed at the same time. However, we formulate surrogate-based optimization as a pattern recognition problem: a set of surrogates is built during a training phase and then matched against new problems during a test phase. Since the matching is based on a limited set of sentry points, we propose a multi-level optimization approach where the fidelity is increased as the solution proposed by a preceding level is rejected (at the cost of a higher computational burden).

The underlying assumption behind the dual surrogate mechanism is that whenever a model is a good representation of the new problem but did not result in a successful recall, a near optimal solution can be found through a fine search. Model update requirements in such case is minimal. Otherwise, a full search is required at the cost of a more expensive model update, involving a greater number of exact fitness evaluations.

The recall mechanism will provide the starting surrogate model and a set of fitness values for an initial update. Moreover, the optimal solution ($\boldsymbol{X}_{S,o}$) is initialized with the best recalled solution. The GMM that resulted in the smallest KS value during recall (updated with the re-evaluated solutions) will be chosen as the initial surrogate. We also inject the best recalled solutions of that probe into both, the surrogate and exact fitness swarms (as proposed by Kapp *et al* (Kapp *et al.*, 2011)). Since the GMM has been trained using all solutions found during the optimization of a given image, it should be considerably more precise than a model built using a few sampled points.

Three aspects are crucial in the surrogate optimization levels: updating GMMs with new data in an on-line manner, performing regression on GMMs and validating the evolution of the off-line surrogate against the exact fitness function.

4.4.3.1 On-line update of GMMs

Model update (see "Update surrogate with re-evaluated solutions", "Update surrogate with selected solutions" blocks in Figure 4.4 and "Re-evaluate best swarm $\boldsymbol{X}_A$ solution on exact fitness and update surrogate" block in Figure 4.5) is an essential issue in surrogate-based op-

timization. The baseline intelligent watermarking system already relies on GMM modeling of all solutions found through all generations (optimization history) in order to model a fitness landscape. A GMM is a powerful statistical modeling technique which consists of a linear combination of a finite number of Gaussian models

$$p(\boldsymbol{x}|\Theta) = \sum_{j=1}^{K} \alpha_j \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{4.4}$$

where $p(\boldsymbol{x}|\Theta)$ is the probability density function (pdf) of a continuous random vector $\boldsymbol{x}$ given a mixture model Θ, K is the number of mixtures, α_j is the mixing weights, parameters of the j^{th} model (with $0 < \alpha_j \leq 1$ and $\sum_{j=1}^{K} \alpha_j = 1$) and $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu_j}, \boldsymbol{\Sigma}_j)$ is a multivariate Gaussian probability density function (pdf) with mean vector $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$.

In the baseline approach, a GMM is first estimated in batch mode with optimization history data using Expectation Maximization (EM) (Figueiredo and Jain, 2000). Then, this new GMM is either inserted or employed in order to update an existing GMM in the LTM according to a distance metric (Sfikas *et al.*, 2005). During update, components of the the new and existing GMMs are merged based on their Bhattacharyya distance (Hennig, 2010).

Since the proposed approach relies on GMMs obtained for a training stream of images in order to predict fitness values for a different stream of images, it is crucial to adapt a GMM using new data. An intuitive approach would be to use the same strategy employed in the baseline system (train a new GMM using new data and then merge with the existing GMM). However, the number of data points needed to estimate a covariance matrix is $N_d = d + d(d+1)/2$ which means it grows quadratically with dimension d (Figueiredo and Jain, 2000) making unfeasible the applicability of such approach for a small quantity of data. Engel and Heinen (Engel and Heinen, 2010) tackle this problem by starting with an initial uniform covariance matrix $\boldsymbol{\Sigma}^0 = \sigma_{ini}^2 \boldsymbol{I}$ where σ_{ini} is the width of the initial covariance matrix and $\boldsymbol{I}$ is an identity matrix and incrementally adding new components or updating existing ones based on a novelty criterion. Such approach assumes an untrained GMM and is justified in situations where a new GMM has to be trained from scratch in an on-line fashion. However, there are several practical limitations on training a GMM using a small quantity of data such as initialization of

mixture components, escaping from situations where two or more components share the same data points, defining the appropriate number of components (Figueiredo and Jain, 2000).

Since we rely on an initial GMM trained in batch mode using a technique that can tackle the issues above, we can rely on this initial model and then adjust its components using new data. There are two strategies to do that. The first is to rely on some sort of statistics for each component about the previous update in order to adjust the components using the new datum (Yamanishi *et al.*, 2000; Zhang and Scordilis, 2008). The second is to rely on a learning factor which is gradually decreased (Stauffer and Grimson, 2000). We will employ the second approach (slightly adapted to our specific problem) since the first assumes a fixed number of components and our baseline memory management mechanism employs pruning in order to adjust the number of components according to new data which would result in loss of such statistic. Given a new datum $\boldsymbol{x}_t$ at time t, we first find the index of the component that best fits $\boldsymbol{x}_t$:

$$j^* = argmax_j\{\mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\} \tag{4.5}$$

and then update the mixture weights of the components:

$$\alpha_j^t = \begin{cases} (1-\gamma)\alpha_j^{t-1} + \gamma, & \text{if } j = j^* \\ (1-\gamma)\alpha_j^{t-1}, & \text{otherwise} \end{cases} \tag{4.6}$$

where γ is the learning rate. The mean and covariance matrix of the best fit component are updated in a similar manner:

$$\boldsymbol{\mu}_{j^*}^t = (1-\rho)\boldsymbol{\mu}_{j^*}^{t-1} + \rho\boldsymbol{x}_t \tag{4.7}$$

$$\boldsymbol{\Sigma}_{j^*}^t = (1-\rho)\boldsymbol{\Sigma}_{j^*}^{t-1} + \rho(\boldsymbol{x}_t - \boldsymbol{\mu}_{j^*}^t)^T(\boldsymbol{x}_t - \boldsymbol{\mu}_{j^*}^t) \tag{4.8}$$

where

$$\rho = \alpha_{j*}^{t-1}\mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_{j*}^{t-1}, \boldsymbol{\Sigma}_{j*}^{t-1}) \tag{4.9}$$

4.4.3.2 Gaussian Mixture Regression (GMR)

In the proposed approach, GMR (see "Iterate swarm $\boldsymbol{X}_A$ on surrogate" block in Figure 4.4 and "PSO swarm $\boldsymbol{X}_A$ on surrogate" block in Figure 4.5) allows employing the knowledge of previous cases of optimization to decrease the computational burden of re-optimization. The main motivation for relying on GMMs in order to model the fitness landscape of a stream of optimization problems is that it combines the memorization ability of non-parametric techniques with the compactness of parametric techniques. It has been observed in our previous research, that in this specific application it allows a very precise sampling of the fitness landscape. Sampling solutions from a GMM $\Theta = \{(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), ..., (\boldsymbol{\mu}_K, \boldsymbol{\Sigma}_K)\}$ is straightforward (Equation 4.3).

However, as observed by Sung (Sung, 2004), in order to employ a GMM in regression we must assume a joint density of the form:

$$p(\boldsymbol{a}_1, \boldsymbol{a}_2) = \sum_{j=1}^{K} \alpha_j \mathcal{N}(\boldsymbol{a}_1, \boldsymbol{a}_2; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{4.10}$$

where $\boldsymbol{a}_1 = \boldsymbol{x}$ is the independent (design) variable, $\boldsymbol{a}_2 = \boldsymbol{f}(\boldsymbol{x})$ is the dependent variable and:

$$\boldsymbol{\mu}_j = \begin{bmatrix} \boldsymbol{\mu}_{j,1} \\ \boldsymbol{\mu}_{j,2} \end{bmatrix} \tag{4.11}$$

$$\boldsymbol{\Sigma}_j = \begin{bmatrix} \boldsymbol{\Sigma}_{j,11} & \boldsymbol{\Sigma}_{j,12} \\ \boldsymbol{\Sigma}_{j,21} & \boldsymbol{\Sigma}_{j,22} \end{bmatrix} \tag{4.12}$$

which is not the case in Equation 4.3.

By deriving Equation 4.10, Sung (Sung, 2004) formulated that such partition of a density allows employing a GMM as a regression model:

$$\hat{\boldsymbol{f}}(\boldsymbol{x}, \Theta) = \sum_{j=1}^{K} P_j(\boldsymbol{\theta}_j|\boldsymbol{x}) \boldsymbol{m}_j(\boldsymbol{x}) \tag{4.13}$$

$$\boldsymbol{\varepsilon}^2(\boldsymbol{x}, \Theta) = \sum_{j=1}^{K} P_j(\boldsymbol{\theta}_j|\boldsymbol{x})(\boldsymbol{m}_j(\boldsymbol{x})^2 + \boldsymbol{\sigma}_j^2) - (\sum_{j=1}^{K} P_j(\boldsymbol{\theta}_j|\boldsymbol{x}) \boldsymbol{m}_j(\boldsymbol{x}))^2 \tag{4.14}$$

where:

$$\boldsymbol{m}_j(\boldsymbol{x}) = \boldsymbol{\mu}_{j,1} + \boldsymbol{\Sigma}_{j,21}\boldsymbol{\Sigma}_{j,11}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_{j,1}) \tag{4.15}$$

$$\boldsymbol{\sigma}_j^2 = \boldsymbol{\Sigma}_{j,22} - \boldsymbol{\Sigma}_{j,21}\boldsymbol{\Sigma}_{j,11}^{-1}\boldsymbol{\Sigma}_{j,12} \tag{4.16}$$

$$P_j(\boldsymbol{\theta}_j|\boldsymbol{x}) = \frac{\alpha_j \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{j,11}, \boldsymbol{\Sigma}_{j,11})}{\sum_{j=1}^{K} \alpha_j \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_{j,1}, \boldsymbol{\Sigma}_{j,11})} \tag{4.17}$$

This approach provides a distribution of the predicted value with $\hat{\boldsymbol{f}}(\boldsymbol{x})$ as the mean and $\boldsymbol{\varepsilon}^2(\boldsymbol{x})$ as the covariance matrix. This makes GMR a very interesting approach for situations where a smooth approximation of a function is necessary like robotics (Calinon, 2009). Predicting fitness values using this technique is straightforward, for a given $\boldsymbol{x}$, we compute $f_P(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}) + \varepsilon(\boldsymbol{x})$ using Equations 4.13 and 4.14. It is important noticing that in the given application, the predicted value and error are scalars (mean and variance) rather than a vector and a covariance matrix.

4.4.3.3 Evolution control

Avoiding convergence to false optima is one of the most important issues in harnessing the computational cost savings allowed by surrogate-based optimization. This is specially important for level 3 which relies mostly on surrogate fitness evaluation. For this reason, we propose the use of an evolution control mechanism (see "Evolution control" block in Figure 4.4) for the off-line surrogate in order to mitigate this problem. Because of the space-filling nature of surrogate models, optima will consist many times of an interpolation of many different near optimal points. For this reason, model fidelity tends to be improved as the model is updated. However, this requires re-evaluating more fitness values resulting in an increase in computational burden.

As mentioned before, the model fidelity versus computational burden trade-off varies across different applications and can be adjusted with the use of evolution control. There are two main approaches to evolution control (Jin *et al.*, 2000): (1) controlled individuals; (2) controlled generations. In controlled individuals, the actual and predicted fitness values of part of

the individuals in the population are re-evaluated with the real fitness function. In controlled population, the whole population is re-evaluated at a certain time interval.

We propose using an individual-based approach as it has a smaller computational burden than generation-based approach. In our approach, for each generation, solutions in the surrogate swarm are ranked according to their surrogate fitness value and the N_{s1} best performing solutions are re-evaluated in the exact function $f(\boldsymbol{x})$. If both, the predicted and effective fitness value for the best re-evaluated solution is better than that of the best re-evaluated solution (optimal) found so far, then the optimal solution is replaced by the best re-evaluated solution for that generation. This can be seen as a pre-selection strategy (Gräning *et al.*, 2005) as the parents of the next generation (attractors in PSO terms) are chosen among the best re-evaluated solutions.

4.4.3.4 Off-line surrogate PSO

In the off-line surrogate optimization, the PSO approach described in (Vellasques *et al.*, 2011) will be employed in order to optimize the embedding parameters, but using the surrogate as fitness function (approach described in Section 4.4.3.2). The surrogate is initialized with the best recalled mixture (see 4 in Figure 4.4). The best recalled mixture is the one that resulted in the smallest KS value during STM/LTM recall. After that, the surrogate is updated using all the solutions re-sampled during recall and their re-evaluated fitness solution (based on the approach described in Section 4.4.3.1). At each generation, the velocity and position of surrogate swarm solutions ($\boldsymbol{X}_A$) are updated based on the surrogate fitness and the N_{s1} best solutions are re-evaluated in $\boldsymbol{Co}_i$. The model is updated using these solutions and their re-evaluated fitness. If the best re-evaluated fitness ($f(\boldsymbol{x}_{g,s1})$) improves the candidate optimal solution ($\boldsymbol{X}_{S,o}$) then the surrogate global best ($\boldsymbol{p}_{g^*,s1}$) is replaced with it. This process (optimization, re-evaluation, model update, best solution update) is repeated until no improvement in the best solution occurs for a given number of generations.

It is important to observe that in surrogated-based optimization, predicted improvements in $\boldsymbol{X}_{S,o}$ must correspond to actual improvements. That is, if an improvement has been predicted but not achieved (or the opposite), it means that the surrogate provides little knowledge about

that specific region. Therefore we propose updating $\boldsymbol{X}_{S,o}$ only if an improvement has been predicted and achieved (Dennis and Torczon, 1995) (more specifically, if $\frac{f(\boldsymbol{X}_{S,o})-f(\boldsymbol{x}_{g,s1})}{\boldsymbol{X}_{S,o}-f_P(\boldsymbol{x}_{g,s1},\Theta)} > 0$). After the stop criterion has been reached (no improvement in $\boldsymbol{X}_{S,o}$ for a certain number of generations), if at least one case of improvement has occurred during the whole optimization process, the best re-evaluated solution found will be employed as is and the LTM is updated with the surrogate model. Otherwise, level 4 is activated.

Algorithm 6 summarizes the off-line surrogate level. The optimal solution ($\boldsymbol{x}_{o1}$) is initialized with the best recalled solution (line 1). Then, the surrogate model (Θ_b) is updated with all the re-sampled solutions ($\boldsymbol{X}_S$) and respective fitness values (line 2). After that, the swarm is iterated (velocity and position update) based on the surrogate fitness (line 4). The best N_{s1} solutions are re-evaluated on image $\boldsymbol{Co}$ (line 5). If the best re-evaluated solution improves the optimal solution (line 6), the optimal solution (line 7) and the surrogate swarm global best (line 8) are updated with the best re-evaluated solution. Next, the surrogate model is updated with the best N_{s1} re-evaluated solutions (line 10). Lines 4 to 10 are repeated until a stop criterion has been reached (optimal solution did not improve for a certain number of generations). Next, if at least one improvement occurred in the optimal solution (line 12), the LTM is updated (either merge or insert) with the surrogate (line 13) and the optimal solutions is employed on $\boldsymbol{Co}$ avoiding the costlier level 4.

4.4.3.5 On-line surrogate PSO

The on-line surrogate technique is based on the approach of Parno *et al* (Parno *et al.*, 2011). Two populations ($\boldsymbol{X}_A$ and $\boldsymbol{X}_B$) are employed, one for the surrogate fitness function and another one for the exact fitness. The $\boldsymbol{X}_B$ population is partially initialized with solutions sampled from the same mixture employed in the surrogate initialization (see 6 in Figure 4.5). Optimization is performed first using population $\boldsymbol{X}_A$ on the surrogate fitness function (see 7 in Figure 4.5). The best solution from $\boldsymbol{X}_A$ ($\boldsymbol{p}_{g,s2}$) is re-evaluated in the current image. If it improves the neighborhood best of population $\boldsymbol{X}_B$, that neighborhood best is replaced with $\boldsymbol{p}_{g,s2}$. The surrogate model is updated using the re-evaluated solution. Next, an iteration is performed using population $\boldsymbol{X}_B$ on the exact fitness. This process (optimization on $\boldsymbol{X}_A$, re-evaluation on exact fitness,

Algorithm 6 Off-line surrogate optimization.

Inputs:
$\boldsymbol{Co}$ – cover image.
Θ_b – surrogate model (mixture model which resulted in best KS value during recall).
$\boldsymbol{X}_S$ – set of all solutions sampled during recall.
N_{s1} – number of solutions for evolution control.

Definitions:
$\boldsymbol{X}_{S,o}$ – best recalled solution.
$f_P(\boldsymbol{x}, \Theta)$ – surrogate fitness (Equations 4.2, 4.13 and 4.14).
$\boldsymbol{x}_{g,s1}$ – best re-evaluated solution for current generation.
$\boldsymbol{p}_{g^*,s1}$ – surrogate swarm global best.

Output:
$\boldsymbol{x}_{o1}$ – optimal solution.

$\boldsymbol{x}_{o1} \leftarrow \boldsymbol{X}_{S,o}$
Update Θ_b with $\boldsymbol{X}_S$ (Equations 4.6, 4.7 and 4.8).
repeat
    Iterate swarm (update particles velocity and position) based on $f_P(\boldsymbol{x}, \Theta)$.
    Re-evaluate the best N_{s1} solutions on $\boldsymbol{Co}$.
    if $\frac{f(\boldsymbol{x}_{o1}) - f(\boldsymbol{x}_{g,s1})}{f(\boldsymbol{x}_{o1}) - f_P(\boldsymbol{x}_{g,s1}, \Theta_b)} > 0$ **then**
        $\boldsymbol{x}_{o1} \leftarrow \boldsymbol{x}_{g,s1}$
        $\boldsymbol{p}_{g^*,s1} \leftarrow \boldsymbol{x}_{g,s1}$
    end if
    Update Θ_b with the best N_{s1} solutions and respective re-evaluated fitness values.
until Stop criterion has been reached
if $f(\boldsymbol{x}_{o1}) < f(\boldsymbol{X}_{S,o})$ **then**
    Update LTM with Θ_b.
end if

injecting the re-evaluated solution on $\boldsymbol{X}_B$, iteration on $\boldsymbol{X}_B$) is repeated until a stop criterion has been reached.

This approach allows avoiding the extra cost of stratified sampling since (1) the initial model is expected to provide some knowledge about the new problem; (2) surrogate in level 4 is more like an insurance policy for the previous levels (in the worst case, the surrogate will provide no improvement and the performance will be equivalent to that of completely reseting the swarm for each new image). However, as observed in (Parno *et al.*, 2011), such approach generally results in a speed up in convergence time compared to full optimization. The reason is that

it relies primarily on exact fitness evaluations, which should compensate any false optimum found in the surrogate fitness. Thus, evolution control is not an issue in level 4.

After optimization if finished, the best solution is employed for the given image and all solutions found during the course of the optimization of the exact function are employed in order to train a GMM (see 8 in Figure 4.5). The resulting GMM and best solution will form a probe that will replace the current STM probe. The LTM update works as follows: the GMM of the new probe is either merged with the GMM of the most similar probe in the LTM or inserted based on a C2 distance (Sfikas *et al.*, 2005) threshold (computed over the last T cases of re-optimization). The mean value of the smallest C2 distance for each update operation (μ_δ^t) is computed for the T last cases of re-optimization. An insert occurs if $C2 - \mu_\delta^t$ is greater than the standard deviation (σ_δ^t) for the same time-frame. Otherwise a merge operation is performed. The LTM update procedure is described more carefully in (Vellasques *et al.*, 2012b).

Algorithm 7 summarizes level 4. Initially, N_i solutions are re-sampled from the surrogate model (Θ_b) and injected into the exact fitness swarm ($\boldsymbol{X}_B$, line 1). The optimal solution ($\boldsymbol{x}_{o2}$) is initialized with the best recalled solution (line 2). Then, the solutions in the surrogate swarm ($\boldsymbol{X}_A$) are initialized randomly (line 4) and $\boldsymbol{X}_A$ is optimized based on the surrogate function (line 5) until a stop criterion has been reached (global best did not improve for a certain number of iterations). Next, the surrogate global best ($\boldsymbol{p}_{g^*,s2}$) is re-evaluated on $\boldsymbol{Co}$ (line 6) and the surrogate model is updated with the re-evaluated $\boldsymbol{p}_{g^*,s2}$ (line 7). After that, the corresponding best neighbor in $\boldsymbol{X}_B$ is updated with $\boldsymbol{p}_{g^*,s2}$ accordingly (lines 8 to 11). Next, $\boldsymbol{X}_B$ is iterated based on the exact fitness function (line 12) and the optimal solution is updated with the best of generation ($\boldsymbol{x}_{B,g}$) accordingly (lines 13 to 15). The procedure between lines 4 and 15 is repeated until the stop criterion has been reached ($\boldsymbol{x}_{B,g}$ did not improve $\boldsymbol{x}_{o2}$ for a certain number of generations). Finally, a new GMM is created using genotypic and phenotypic data from all re-evaluated solutions (including recall) and the STM/LTM memory is updated (line 17).

Algorithm 7 On-line surrogate optimization.

Inputs:
$\boldsymbol{Co}$ – cover image.
Θ_b – mixture model which resulted in best KS value during recall.
N_i – amount of injected solutions.

Definitions:
$\boldsymbol{p}_g$ – exact fitness swarm neighborhood best.
$\boldsymbol{X}_A$ – surrogate population.
$\boldsymbol{X}_B$ – exact function population.
$\boldsymbol{x}_{B,g}$ – best of generation ($\boldsymbol{X}_B$).
$\boldsymbol{p}_{g^*,s2}$ – surrogate swarm global best.
$\boldsymbol{X}_{\boldsymbol{x},k}$ – k nearest neighbors of $\boldsymbol{x}$ in $\boldsymbol{X}_B$.

Output:
$\boldsymbol{x}_{o2}$ – optimal solution from $\boldsymbol{X}_B$.

Re-sample N_i solutions from Θ_b and inject into $\boldsymbol{X}_B$.
$\boldsymbol{x}_{o2} \leftarrow \boldsymbol{X}_{S,o}$
repeat
  Re-randomize $\boldsymbol{X}_A$.
  Optimize $\boldsymbol{X}_A$ based on Θ_b.
  Re-evaluate $\boldsymbol{p}_{g^*,s2}$ on $\boldsymbol{Co}$.
  Update Θ_b with $\boldsymbol{p}_{g^*,s2}$.
  $\boldsymbol{p}_g \leftarrow min_{f(x)}\{\boldsymbol{X}_{\boldsymbol{p}_{g^*,s2},k}\}$
  if $f(\boldsymbol{p}_{g^*,s2}) < \boldsymbol{p}_g$ **then**
    $\boldsymbol{p}_g \leftarrow \boldsymbol{p}_{g^*,s2}$
  end if
  Iterate $\boldsymbol{X}_B$ (update particles velocity and position) based on $\boldsymbol{Co}$.
  if $f(\boldsymbol{x}_{B,g}) < f(\boldsymbol{x}_{o2})$ **then**
    $\boldsymbol{x}_{o2} \leftarrow \boldsymbol{x}_{B,g}$
  end if
until Stopping criterion (on $\boldsymbol{X}_B$) has been reached
Generate new GMM using phenotypic and genotypic data from all re-evaluated solutions from all levels (including optimization history of level 4) and update STM and LTM with new GMM and $\boldsymbol{p}_{g^*,s2}$.

4.5 Experimental methodology

For proof-of-concept simulations, two watermarks are employed in all experiments for all databases as in (Vellasques *et al.*, 2011, 2012b): the 26×36 resolution BancTec logo (Fig-

ure 4.6a) as robust watermark and the 36×26 resolution Université du Québec logo (Figure 4.6b) as fragile watermark.

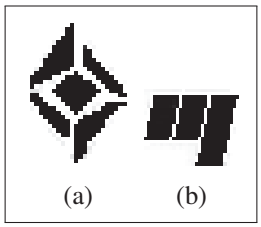


Figure 4.6 Bi-tonal logos used as watermarks: (a) BancTec, and (b) Université du Québec.

The experiments were conducted using the University of Oulu's MediaTeam (Sauvola and Kauniskangas, 1999) (OULU-1999) document image database, which is considerably heterogeneous. The same protocol was followed as in (Vellasques *et al.*, 2012b): the images were binarized and 15 of the 512 images were discarded because they have less than 1872 flippable pixels (Muharemagic, 2004; Wu and Liu, 2004) which is the minimum required to embed the watermarks presented above. Then, the 497 images were randomly split into a training set containing 100 images (OULU-1999-TRAIN), and test set, containing 397 images (OULU-1999-TEST). Figure 4.7 shows some examples from the OULU-1999-TRAIN database. Two more homogeneous databases: TITI-61 and CVIU-113-3-4 containing respectively 61 and 342 binarized pages from issues 113(3) and 113(4) of the Computer Vision and Image Understanding journal as described in (Vellasques *et al.*, 2011) were employed. A database – named SHUFFLE – comprising images from both Oulu and CVIU databases, but with their positions shuffled was also employed.

The proposed approach was evaluated for optimization of embedding parameters for a bi-tonal watermarking system by considering four main situations: (1) no attack; (2) cropping 1%; (3) cropping 2%; (4) salt and pepper with intensity 0.02.

The technique described in (Vellasques *et al.*, 2012b) was applied to OULU-1999-TRAIN, TITI-61 data and a combination of both. Simulations were conducted based on the four situations described above in order to create the memories for the DS-DPSO simulations. These

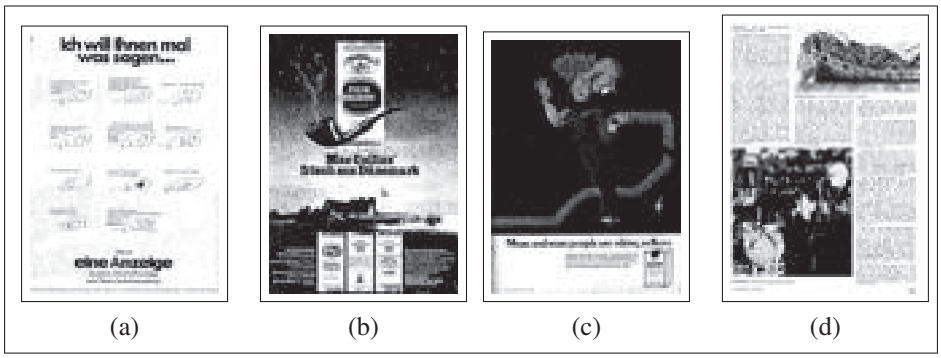


Figure 4.7 Examples of document images from OULU-1999-TRAIN: (a) image 1, (b) image 2, (c) image 5, and (d) image 6.

were conducted on the OULU-1999-TEST, CVIU-113-3-4 and SHUFFLE streams in order to validate the following cases.

Case I – adaptation performance

Tackling adaptation in scenarios involving significant variations in the stream of optimization problems is the motivation behind the proposed approach. In order to validate adaptability, the memory of OULU-1999-TRAIN is employed with no attack as a starting point for OULU-1999-TRAIN with cropping of 2%. Next, the resulting memory for OULU-1999-TRAIN is employed with salt and pepper 0.02. Finally, the resulting memory is employed as starting point in four separate scenarios for OULU-1999-TEST: (I) no attack, (II) cropping of 2%, (III) salt and pepper with intensity of 0.02, (IV) randomly chosen attacks (no attack, cropping 2%, salt and pepper 0.02) and (IVa) same as IV but without the use of a memory of previous cases of optimization (to validate the impact of previous knowledge in such challenging scenario).

Case II – comparison to previous DPSO approach (Vellasques *et al.*, 2012b)

In order to evaluate the performance of the proposed approach in a more stable scenario, simulations are performed using no attack and cropping of 1% on all streams. Simulations with and without the use of a previous memory are performed for the test streams in order to assess the impact of a previous memory in the performance of the proposed approach.

Case III – memorization capacity

In the memorization experiment, the memory management mechanism is de-activated first (all re-optimizations result in a LTM insert operation) in order to avoid any possible bias caused by the merge operators (memory management is resumed in the test phase). Then, a memory is created by applying the proposed technique with both, level 3 and surrogate of level 4 de-activated to OULU-1999-TRAIN with cropping of 1%. A more restrictive confidence level (α_{Crit}) of 0.8 during training is proposed for this particular case in order to obtain high fidelity probes (we propose a less restrictive confidence level of 0.95 for all the other simulations). Then, a probe from OULU-1999-TRAIN is chosen and has its performance evaluated for the off-line and on-line surrogate mechanisms (on OULU-1999-TRAIN as well) in two situations: (1) for cases where the selected probe resulted in a successful recall; (2) for cases where re-optimization was triggered. The motivation for this experiment is to understand the impact of previous knowledge in the computational cost of a given optimization task and also to understand at what point previous knowledge can be helpful when the new problem is knowingly different from any previous problem.

Case IV – management of different attacks

To validate how well the proposed approach can tackle other attacks, we created two other memories using OULU-1999-TRAIN: one using cropping of 2% and another one using salt and pepper with 0.02 intensity. Then, these memories are employed in OULU-1999-TEST for the same two attacks. We also evaluated the performance of the proposed approach without the use of a previous memory on both, the OULU-1999-TRAIN and OULU-1999-TEST streams.

Parameters values

In the first two levels, 19 solutions are re-sampled and are re-evaluated along with the global best for change detection. DPSO parameters for levels 3 and 4 are set as in (Vellasques *et al.*,

2011). Constants c_1 and c_2 are set to 2.05 while χ is set to 0.7298. Population size is set to 20 particles and optimization halts if the global best has not improved for 20 iterations. The neighborhood size of the L-Best topology is set to 3.

The number of solutions employed in the evolution control for level 3 (N_{s1}) was set to 6 which corresponds to 30% of the population. The constant ρ_c defines the trade-off between exploitation and exploration for the surrogate and was set to 1. The LTM size was limited to 20 probes. In all cases, the DPSO stops optimization if the global best has not improved for 20 generations. The number of previous cases of re-optimizations employed in order to compute the insert/update threshold (T) was set to 10. In level 4, surrogate-based DPSO is performed for each generation of exact fitness DPSO. The neighborhood size for the DPSO approach (and for the comparison in the on-line surrogate update) was set to 3. The learning rate of the GMM update technique (γ) was set to 0.02 at the beginning of each re-optimization and decreased for each sample ($\gamma^t = d_\gamma \gamma^{t-1}$) where $d_\gamma = 0.99$ is the learning rate decay.

Table 4.1 Parameters employed in most of the simulations.

Parameter	**Description**	**Value**
α_{Crit}	Confidence level	0.95
γ^0	Initial learning rate	0.02
d_γ	Learning rate decay	0.99
$\|X\|$	Population size	20
N_{s1}	Evolution control population size	6
ρ_c	Surrogate exploration/exploitation trade-off	1
c_1	Acceleration constant 1	2.05
c_2	Acceleration constant 2	2.05
T	Number of previous re-optimizations to compute the insert/update threshold	10
χ	Constriction factor	0.7298

4.6 Simulation results

4.6.1 Case I – adaptation performance

The simulations involving adaptation over heterogeneous streams (Tables 4.2 and 4.3) show the main advantage of the proposed DS-DPSO approach. Since adaptation involves a more restrictive confidence level (0.8) which leads to more re-optimizations, the surrogate optimizers become more dominant than for homogeneous streams.

In the first transition (OULU-1999-TRAIN with no attack to OULU-1999-TRAIN with cropping 2%), the proposed approach allowed substantial decrease in computational burden. In the 8 times re-optimization was triggered, the off-line surrogate allowed an improvement in 3 cases, avoiding costly on-line optimization. For this reason, the total number of fitness values suffered decreased of 26.6% compared to the GMM-based approach (from 13500 to 9903).

The same improvement in computational performance was noticed for the second transition (to OULU-1999-TRAIN with salt and pepper 0.02). This time, off-line surrogate optimization was enough for 5 of the 14 cases of re-optimization. This led to a decrease of 12.9% in the number of fitness evaluations compared to the GMM-based approach (from 18360 to 15990). It is worth noticing that such decrease was made possible despite a higher number of re-optimizations for the DS-DPSO approach (14 versus 12). The same phenomenon was repeated for the OULU-1999-TEST with cropping of 2% (a decrease of 24.6%), salt and pepper 0.02 (a decrease of 36%).

In all cases, DS-DPSO had a smaller computational burden when compared to the previous approach while the watermarking performance was practically the same.

4.6.2 Case II – comparison to previous DPSO approach (Vellasques *et al.*, 2012b)

In terms of computational burden, the DS-DPSO approach resulted in improvement for most cases (Table 4.4). All this, with a comparable precision (Table 4.5).

Table 4.2 Average computational cost performance. $AFPI$ is the average number of fitness evaluations per image. Values in parentheses are standard deviation. F_{Evals} is the cumulative number of fitness evaluations required to optimize the whole stream and DFE is the decrease in the number of fitness evaluations compared to full optimization.

Attack	Database	Full PSO		GMM-based			DS-DPSO		
		$AFPI$	F_{Evals}	$AFPI$	F_{Evals}	DFE	$AFPI$	F_{Evals}	DFE
Cropping 2%	OULU-1999-TRAIN	860 (335)	86040	135 (297)	13500	84.3%	99 (298)	9903	88.5%
S&P 0.02		893 (354)	89280	184 (417)	18360	79.4%	160 (309)	15990	82.1%
No attack (I)	OULU-1999-TEST	1007 (341)	399840	112 (278)	44600	88.9%	85 (171)	33649	91.6%
Cropping 2% (II)		828 (309)	328900	72 (187)	28400	91.4%	59 (166)	23283	92.9%
S&P 0.02 (III)		978 (379)	388220	123 (300)	49020	87.4%	79 (218)	31366	91.9%
Random (IV)		951 (344)	377640	138 (307)	54880	85.5%	123 (244)	48709	87.1%
Random (IVa)		951 (344)	377640	221 (410)	87720	76.3%	149 (287)	58979	84.1%

Table 4.3 Average watermarking performance, where the mean μ and standard deviation σ of each metric are presented as $\mu(\sigma)$.

Attack	Database	Full PSO			GMM-based			DS-DPSO		
		DRDM	BCR robust	BCR fragile	DRDM	BCR robust	BCR fragile	DRDM	BCR robust	BCR fragile
Cropping 2%	OULU-1999-TRAIN	0.04 (0.05)	98.2 (2.7)	99.9 (0.4)	0.04 (0.05)	97.0 (3.6)	99.7 (1.0)	0.05 (0.05)	96.5 (4.9)	99.9 (0.5)
S&P 0.02		0.03 (0.03)	97.9 (2.6)	99.7 (0.5)	0.03 (0.04)	97.3 (3.6)	99.5 (1.2)	0.03 (0.03)	96.9 (4.6)	99.5 (1.0)
No attack (I)	OULU-1999-TEST	0.00 (0.00)	100 (0.0)	100 (0.0)	0.01 (0.02)	99.9 (0.1)	99.9 (0.1)	0.00 (0.02)	100 (0.0)	100 (0.0)
Cropping 2% (II)		0.04 (0.04)	98.0 (3.0)	99.8 (0.7)	0.04 (0.05)	93.3 (6.0)	99.1 (2.0)	0.05 (0.06)	95.8 (6.1)	99.7 (1.3)
S&P 0.02 (III)		0.03 (0.04)	98.0 (2.4)	99.6 (0.6)	0.04 (0.04)	97.1 (3.7)	99.3 (1.1)	0.03 (0.04)	97.4 (3.4)	99.4 (1.0)
Random (IV)		0.02 (0.03)	98.7 (2.3)	99.8 (0.6)	0.03 (0.04)	97.3 (4.3)	99.4 (1.4)	0.03 (0.05)	97.8 (4.1)	99.7 (0.6)
Random (IVa)		0.02 (0.03)	98.8 (2.1)	99.8 (0.6)	0.03 (0.04)	97.6 (3.7)	99.6 (1.0)	0.02 (0.03)	98.0 (3.5)	99.7 (0.8)

Table 4.4 Average cost performance. $AFPI$ is the average number of fitness evaluations per image. Values in parentheses are standard deviation. F_{Evals} is the cumulative number of fitness evaluations required to optimize the whole stream and DFE is the decrease in the number of fitness evaluations compared to full optimization. An asterisk (*) indicates results extracted from (Vellasques *et al.*, 2011).

Attack	**Database**	**Learning**	**Full PSO**		**GMM-based**			**DS-DPSO**		
			$AFPI$	F_{Evals}	$AFPI$	F_{Evals}	DFE	$AFPI$	F_{Evals}	DFE
No attack	OULU-1999-TRAIN	No	925 (286)	92520	66 (194)	6580	92.9%	98 (243)	9809	89.4%
	OULU-1999-TEST	No	1007 (341)	399840	59 (188)	23280	94.2%	56 (177)	22153	94.5%
	OULU-1999-TEST	Yes	1007 (341)	399840	42 (133)	16700	95.8%	62 (204)	24489	93.9%
	TITI-61	No	844 (226)*	51460*	84 (224)	5140	92.6%	130 (336)	7910	88.7%
	CVIU-113-3-4	No	882 (251)*	301580*	76 (233)	26000	91.4%	75 (183)	22066	93.9%
	CVIU-113-3-4	Yes	882 (251)*	301580*	49 (157)	16600	95.4%	41 (121)	14090	96.1%
	SHUFFLE	No	1026 (345)	758500	66 (189)	48840	93.6%	51 (137)	37986	95%
	SHUFFLE	Yes	1026 (345)	758500	54 (179)	40220	94.7%	56 (130)	41129	94.6%
Cropping 1%	OULU-1999-TRAIN	No	887 (340)	88740	179 (363)	17860	79.9%	71 (205)	7139	92%
	OULU-1999-TEST	No	860 (310)	341520	83 (212)	32920	90.4%	66 (194)	26104	92.4%
	OULU-1999-TEST	Yes	860 (310)	341520	67 (205)	26760	92.2%	48 (87)	18890	94.5%
	TITI-61	No	911 (237)*	55580*	52 (178)	3200	94.8%	58 (189)	3553	94.2%
	CVIU-113-3-4	No	872 (251)*	298100*	50 (166)	16980	94.5%	58 (182)	19708	93.6%
	CVIU-113-3-4	Yes	897 (310)	306860	21 (4)	7120	97.7%	21 (4)	7080	97.7%
	SHUFFLE	No	887 (320)	798100	67 (194)	49780	93.8%	52 (138)	38155	95.2%
	SHUFFLE	Yes	887 (320)	798100	49 (136)	36300	95.5%	32 (52)	23690	97%

Table 4.5 Average watermarking performance, where the mean μ and standard deviation σ of each metric are presented as $\mu(\sigma)$. An asterisk (*) indicates results extracted from (Vellasques *et al.*, 2011).

Attack	Database	Learning	Full PSO			GMM-based			DS-DPSO		
			DRDM	BCR robust	BCR fragile	DRDM	BCR robust	BCR fragile	DRDM	BCR robust	BCR fragile
No attack	OULU-1999-TRAIN	No	0.00 (0.00)	100 (0.0)	100 (0.0)	0.00 (0.00)	100 (0.0)	100 (0.0)	0.00 (0.00)	100 (0.0)	100 (0.0)
	OULU-1999-TEST	No	0.00 (0.00)	100 (0.0)	100 (0.0)	0.00 (0.00)	100 (0.0)	100 (0.0)	0.00 (0.00)	100 (0.0)	100 (0.0)
	OULU-1999-TEST	Yes	0.00 (0.00)	100 (0.00)	100 (0.0)	0.00 (0.00)	99.9 (0.4)	99.9 (0.7)	0.00 (0.00)	100 (0.0)	100 (0.0)
	TITI-61	No	0.00 (0.00)*	99.9 (0.5)*	99.7 (0.6)*	0.00 (0.00)	100 (0.0)	100 (0.0)	0.00 (0.00)	100 (0.0)	100 (0.0)
	CVIU-113-3-4	No	0.00 (0.00)*	99.5 (3.6)*	99.3 (3.0)*	0.00 (0.00)	100 (0.0)	100 (0.0)	0.00 (0.00)	100 (0.0)	100 (0.0)
	CVIU-113-3-4	Yes	0.00 (0.00)	100 (0.0)	100 (0.0)	0.00 (0.00)	100 (0.0)	100 (0.0)	0.00 (0.00)	100 (0.0)	100 (0.0)
	SHUFFLE	No	0.00 (0.00)	100 (0.0)	100 (0.0)	0.00 (0.00)	100 (0.0)	100 (0.0)	0.00 (0.00)	100 (0.0)	100 (0.0)
	SHUFFLE	Yes	0.00 (0.00)	100 (0.0)	100 (0.0)	0.00 (0.00)	100 (0.0)	100 (0.0)	0.00 (0.00)	100 (0.0)	100 (0.0)
Cropping 1%	OULU-1999-TRAIN	No	0.03 (0.03)	98.4 (2.1)	99.7 (0.6)	0.03 (0.03)	97.1 (3.8)	99.4 (1.0)	0.03 (0.04)	96.6 (4.6)	99.3 (1.2)
	OULU-1999-TEST	No	0.03 (0.04)	98.4 (2.2)	99.6 (0.6)	0.03 (0.03)	96.7 (4.0)	99.1 (1.5)	0.03 (0.05)	96.1 (4.9)	98.9 (1.7)
	OULU-1999-TEST	Yes	0.03 (0.03)	98.4 (2.2)	99.6 (0.6)	0.03 (0.04)	97.5 (3.3)	99.4 (1.1)	0.03 (0.04)	96.9 (4.0)	99.3 (1.0)
	TITI-61	No	0.00 (0.00)*	92.0 (6.5)*	94.0 (4.0)*	0.03 (0.03)	99.0 (1.8)	99.7 (0.4)	0.02 (0.03)	99.1 (1.6)	99.7 (0.4)
	CVIU-113-3-4	No	0.00 (0.00)*	89.6 (7.1)*	92.5 (5.3)*	0.04 (0.05)	98.3 (3.0)	99.5 (0.8)	0.03 (0.06)	98.3 (3.6)	99.4 (0.9)
	CVIU-113-3-4	Yes	0.02 (0.04)	98.8 (2.3)	99.6 (0.4)	0.04 (0.06)	98.1 (3.0)	99.4 (1.0)	0.04 (0.04)	98.0 (3.7)	99.4 (1.0)
	SHUFFLE	No	0.03 (0.04)	98.6 (2.2)	99.6 (0.5)	0.03 (0.04)	97.1 (4.4)	98.9 (1.8)	0.03 (0.05)	96.9 (4.8)	99.0 (1.5)
	SHUFFLE	Yes	0.03 (0.04)	98.6 (2.2)	99.6 (0.5)	0.03 (0.04)	97.1 (4.3)	99.1 (1.4)	0.03 (0.04)	97.6 (3.1)	99.3 (0.8)

Figures 4.8a to 4.8h show the computational cost for the recall, off-line and on-line levels (no attack) compared to full optimization while Figures 4.9a to 4.9h shows the same but for the cropping 1% simulations.

4.6.2.1 Heterogeneous streams

For the OULU-1999-TEST stream with no attack with training, re-optimization was triggered 14 times. The on-line surrogate was triggered in 12 of these cases which is twice the number of re-optimizations for the GMM-based approach. For this reason, there was an increase of 46.6% in the computational burden compared to the GMM-based approach. Yet, it is important to notice however that the levels 3 and 4 have a smaller computational burden than completely reseting the swarm (Figure 4.8c).

For the SHUFFLE stream with no attack with training, re-optimization was triggered 18 times (versus 16 for the GMM-based approach) and the off-line surrogate replaced the more expensive on-line surrogate for 3 of these cases. The proposed approach was 2.3% costlier than the GMM-based approach. But again, it is worth noticing that in average, levels 3 and 4 are still less expensive than completely resetting the swarm (see Figure 4.8h).

It is possible to observe that for the "no attack" case, the use of a training sequence was not helpful since, for both, OULU-1999-TEST and SHUFFLE streams, there was even a slight increase in the number of fitness evaluations when a training sequence was employed. It is also worth noticing that for the OULU-1999-TRAIN stream, the performance of the proposed approach was even worse than that of the GMM-based approach.

The OULU-1999-TEST with cropping 1% resulted in 8 re-optimizations (versus 16 for the GMM-based approach). The off-line surrogate was enough in 3 of these cases. Combined, the smaller number of re-optimizations and use of surrogates allowed a decrease of 29.4% in the number of fitness evaluations (from 26760 to 18890) compared to the GMM-based approach.

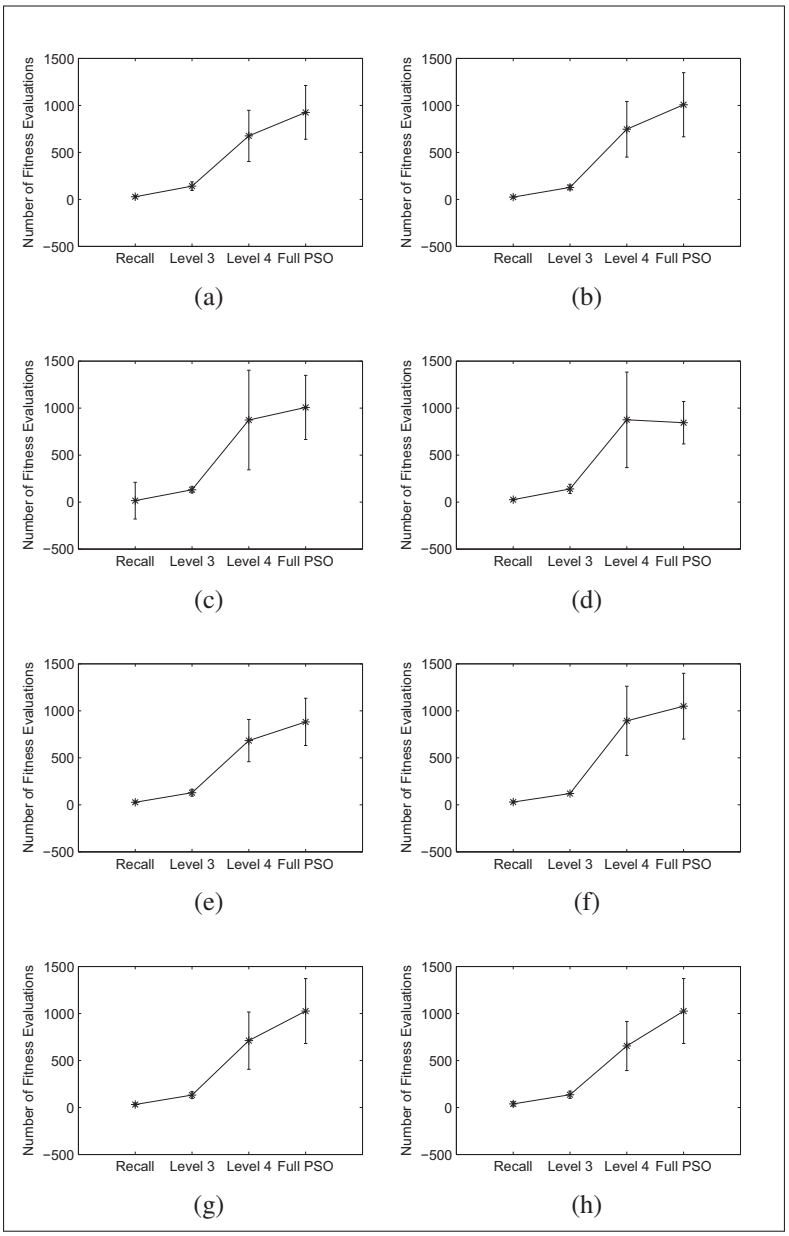


Figure 4.8 Breakdown of computational cost for the "no attack" simulations (compared to full optimization). (a) OULU-1999-TRAIN, no training. (b) OULU-1999-TEST, no training. (c) OULU-1999-TEST, training. (d) TITI-61, no training. (e) CVIU-113-3-4, no training. (f) CVIU-113-3-4, training. (g) SHUFFLE, no training. (h) SHUFFLE, training.

The SHUFFLE stream with cropping 1% resulted in a single re-optimization (versus 17 for the GMM-based approach). This led to a decrease of 34.7% in the number of fitness evaluations between both techniques in the given scenario.

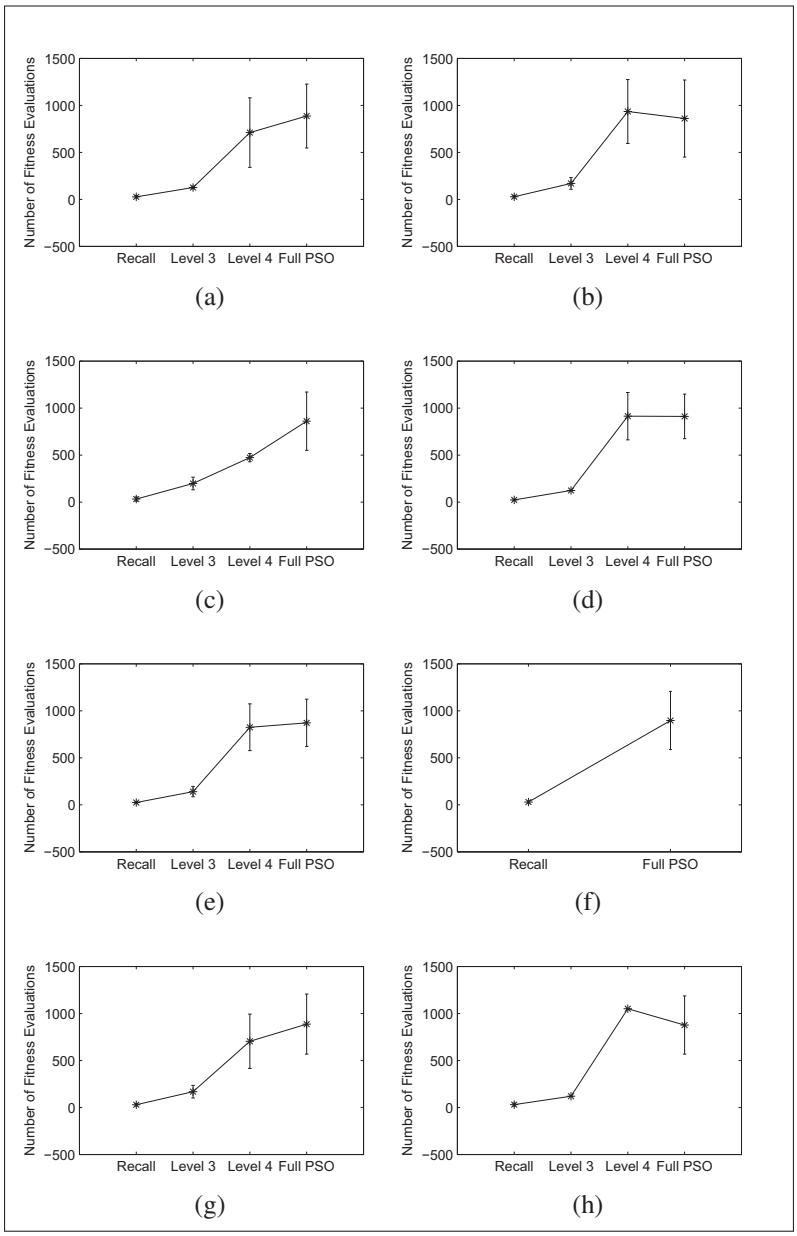


Figure 4.9 Breakdown of computational cost for the cropping 1% simulations (compared to full optimization). (a) OULU-1999-TRAIN, no training. (b) OULU-1999-TEST, no training. (c) OULU-1999-TEST, training. (d) TITI-61, no training. (e) CVIU-113-3-4, no training. (f) CVIU-113-3-4, training. (g) SHUFFLE, no training. (h) SHUFFLE, training.

In the cropping 1% case, the use of a memory of previous solutions affected the computational cost positively. For the OULU-1999-TEST stream, the use of a training sequence led to a decrease of 27.6% in the number of fitness evaluations (from 26104 to 18890) while for the

SHUFFLE stream the use of a training sequence led to a decrease of 37.9% (from 38155 to 23690). This time, the computational burden of the proposed approach for the OULU-1999-TRAIN stream was smaller than that of the previous approach.

4.6.2.2 Homogeneous streams

As observed in Tables 4.4 and 4.5, the proposed technique performance for the CVIU-113-3-4 stream with no attack resulted in a decrease of 15% in the number of fitness values (14090 versus 16600) compared to the GMM-based approach at an equivalent watermarking performance. Re-optimization was triggered 4 times (versus 7 for the GMM-based approach) and in all cases led to level 4 of the approach. For the cropping 1% case, optimization was not triggered at all (as for the GMM-based approach) therefore the computational burden performance of both approaches was nearly identical in this case.

For the no attack case, the use of a training sequence led to a decrease in the number of fitness evaluations for the proposed approach. As for the heterogeneous streams, the proposed approach performed worse than the previous approach for shorter streams.

4.6.3 Case III – memorization capacity

Re-optimization was triggered 21 times in training mode (OULU-1999-TRAIN). A probe was picked and tested against a set of 23 positive images (which resulted in successful recall for that probe) and a set of negative images (which resulted in re-optimization).

Table 4.6 shows the computational cost performance (exact fitness evaluations) for surrogate-based optimization versus no surrogate full optimization in both cases (positive and negative). It is possible to observe that the off-line surrogate resulted in a considerable decrease in the number of fitness evaluations. It is also worth noticing that for the on-line surrogate, although the fitness evaluations are performed primarily on the images, there was still a considerable decrease in the number of exact fitness evaluations which shows that the surrogate increases the convergence speed of the main population.

Figure 4.10 shows the difference between the fitness values (ΔFitness) of full optimization (no surrogate) and surrogate-based optimization (both, off-line and on-line), for each of the positive images. The off-line surrogate (Figure 4.10a) resulted in a slight fitness degradation for a few images, but for most of them, the fitness values were quite similar to those obtained in full optimization. For the on-line surrogate instead (Figure 4.10b), it was possible to observe even a slight improvement in the fitness values for some images.

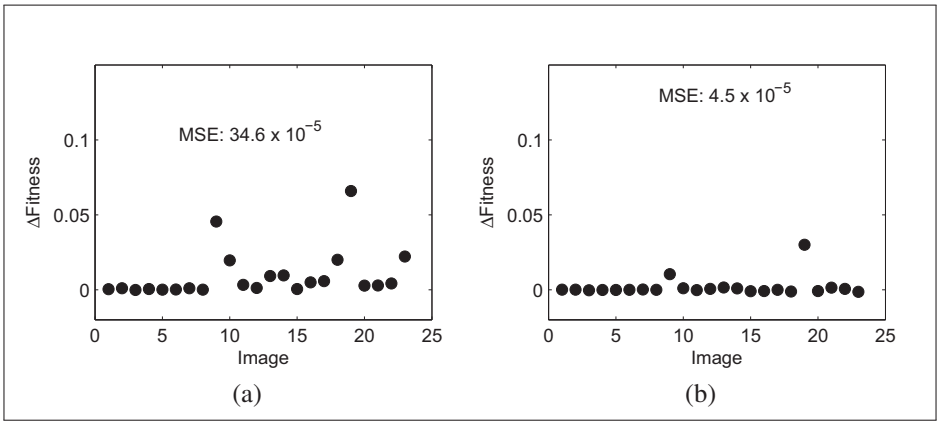


Figure 4.10 Surrogate optimization performance for positive images. (a) Off-line surrogate. (b) On-line surrogate.

Figure 4.11 shows ΔFitness for the negative images. Here it is possible to observe a greater instability for the off-line surrogate (Figure 4.11a) while the on-line surrogate (Figure 4.11b) resulted in a similar performance to that observed for the positive images (as before, there was even an improvement in performance for some images). This demonstrates that as expected, the on-line surrogate is more sensitive to prior knowledge than the off-line surrogate. But it is also worth noticing that the fitness performance was quite good for some images of the off-line surrogate (despite being negative images). This justifies the dual surrogate approach.

It is important to observe that the Mean Squared Error (MSE) between the fitness values obtained in full optimization and in the proposed approach are negligible. However, for both subsets, the MSE obtained for the off-line surrogate is greater than that obtained for the on-line surrogate. It is also worth noticing a considerable deterioration in MSE between the positive and negative images. This justifies the use of an on-line surrogate as a safeguard.

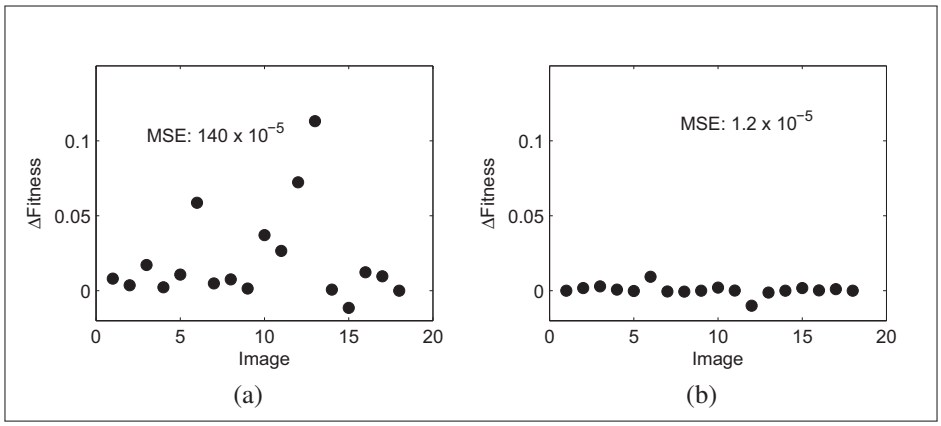


Figure 4.11 Surrogate optimization performance for negative images. (a) Off-line surrogate. (b) On-line surrogate.

4.6.4 Case IV – management of different attacks

The performance for cropping 2% and salt and pepper 0.02 (Tables 4.7 and 4.8) was compatible with that of cropping 1%.

In the case of cropping 2% (OULU-1999-TEST with learning), re-optimization was triggered twice. Only one of these cases required the on-line surrogate (level 4). The number of fitness evaluations suffered a decrease of 7.4% (from 19800 to 18339) when compared to the GMM-based approach.

For the salt and pepper 0.02 (OULU-1999-TEST with learning), re-optimization was triggered once. However, this single case was costlier than full reset (1365 versus 1000 fitness evaluations) as the off-line surrogate did not result in an improvement.

4.6.5 Discussion

Overall, the simulation results demonstrated that the off-line surrogate allows a considerable decrease in the number of fitness evaluations for the cases where re-optimization is triggered. The on-line surrogate operates as a safeguard for the whole system. Since the objective of the on-line surrogate is to improve convergence speed of the population of the exact fitness function, it can be said that its efficiency is tied to how inefficient is the main population. For this

reason, in some cases, when the fourth level was required, it implied in a larger computational burden than full reset. But it is important to remark that this cost also involves the cost of the previous three levels. Therefore, it can be said as a safeguard, the use of a surrogate is preferred to simply using full reset.

The adaptation on more heterogeneous streams simulations demonstrated the main advantage of the proposed approach. In situations involving substantial variability in the stream of optimization problems, the number of re-optimizations is expected to increase considerably. In such case, replacing costly full reset by a lighter surrogate-based optimization becomes crucial. The off-line surrogate was enough in numerous cases in such scenario, allowing even a more substantial decrease in computational burden compared to the more stable scenarios. This advantage can be seen more clearly in Figure 4.12.

For cases of re-optimization (heterogeneous streams), the off-line surrogate successfully replaced the heavier on-line surrogate in numerous situations. Moreover, in many of the cases where it failed, the on-line surrogate gave a boost to the convergence speed of the main swarm, resulting in a further decrease (despite the last resort nature of the fourth level). It was observed that for the no attack case, the use of a memory of previous solutions is irrelevant in what regards decreasing the computational burden. The reason is that the memory in such case is a tool to aid adaptation to novel cases of optimization. Thus, it becomes less important in situations involving little adaptation as the no attack case.

In the memorization simulations, it was possible to observe in general that previous knowledge plays an important role in the performance of the off-line surrogate. It was also possible to observe that for the negative examples (which represent the exact situation in which the surrogate based-optimization is expected to work) the off-line surrogate still allows good watermarking performance for a small fraction of the computational burden of full optimization with no surrogate. And yet, the on-line surrogate works as a safety net for the whole system but also with a smaller computational burden than full optimization with no surrogate.
Finally, the simulations involving homogeneous streams showed one limitation of the proposed approach. The gain obtained by the surrogate-based approach is limited by the number

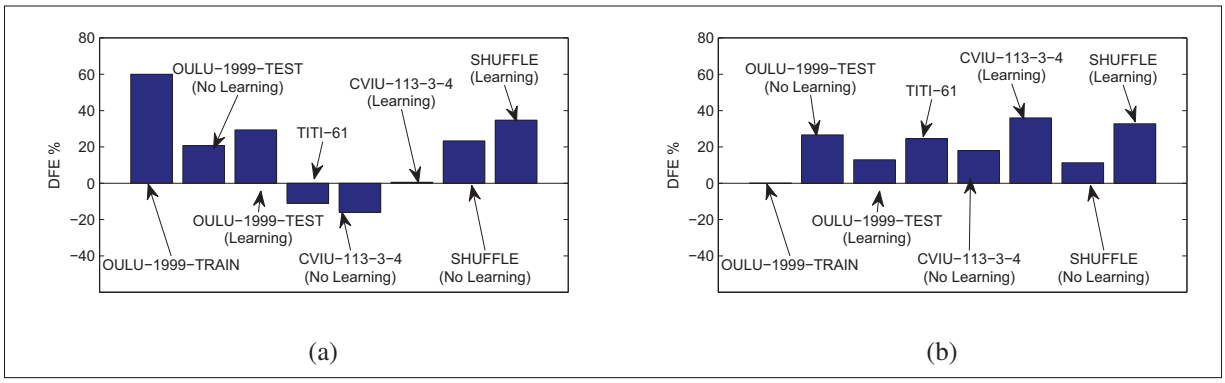


Figure 4.12 Decrease in fitness evaluations (DFE) of DS-DPSO versus GMM-based approach. (a) Cropping 1%, no adaptation. (b) Adaptation simulations.

of re-optimizations. And the number of re-optimizations depends on probe precision. Since solutions obtained during the course of optimization are employed in order to create a probe, probe precision varies depending on the amount of novelty brought by these solutions. Because of their smaller computational burden, memory recall operations are preferred over re-optimization. However, in a case of re-optimization, the use of a surrogate allows a considerable decrease in computational burden compared to full reset.

These experimental results support our strategy of employing two surrogates with different trade-offs between fidelity and computational burden rather than focusing on the detail level (global or local) of each surrogate. It also shows that the use of a memory of surrogates, trained with a separate set of images, contributes even further to the performance of the dual surrogate. Since it was observed that the use of a memory stream is irrelevant for small and stable streams, employing the previous approach is recommended in order to create a memory (using a training stream) and then, employing the proposed approach for larger streams, in situations requiring adaptability.

4.7 Conclusion

In this chapter, a multi-level intelligent watermarking system was proposed. This system is based in four levels. Each level increases the precision of the preceding level at the cost of higher computational burden. The first two levels, as defined in a previous research, comprise

memory recall. These two levels allow matching new optimization problems to previously seen problems stored in a memory of GMMs and recalling ready-to-use solutions for similar problems. The other two levels (3 and 4) are optimization levels and are only activated when the recall modules fail (if embedding parameters require a significant adaptation). During optimization, the most adequate GMM is employed as a surrogate, which is initially updated with the fitness values obtained during recall. The third level performs exploitation and aims at optimizing problems where the optimum is near the surrogate optimum, but could not be found during recall. The fourth level works as a safety net for the whole system, but relies on a surrogate in order to boost convergence.

This approach of using a memory of previously learned surrogates, matched to the new problem using sampling and statistical test is novel and is one of the main contributions of our research. Moreover, this idea of focusing on the trade-off between cost and fidelity of the surrogate rather than on the detail level is also novel and is a secondary contribution of our research.

Experimental results demonstrate that when previous knowledge is available, the off-line surrogate is expected to result in a fitness performance comparable to that of full optimization (with no surrogate) but at a fraction of its cost. It was also demonstrated that in a real situation where the recall failed, it will allow avoiding a more costly on-line surrogate optimization. The on-line surrogate by its way, resulted in a fitness performance that is nearly identical to that of no surrogate (even for cases where recall failed) but with a computational burden that is usually cheaper than that of no surrogate. For this reason, the proposed approach allowed computational savings of up to 93% compared to full optimization in scenarios involving heterogeneous image streams with changing attacks.

These results validate our research hypothesis that whenever re-optimization is triggered, the best fit model should provide a good starting point for building a surrogate for the new problem and even if it cannot, it could still decrease the computational burden of optimization by speeding up convergence. It also demonstrates that knowledge of a new problem can be incorporated into knowledge of previous problems in order to make the model better fit to the new problem. Finally, the results demonstrate the main advantage of the proposed approach which

is tackling intelligent watermarking in scenarios involving substantial variability in the stream of optimization problems.

Table 4.6 Average computational cost performance (surrogate optimization). $AFPI$ is the average number of fitness evaluations per image. Values in parentheses are standard deviation. F_{Evals} is the cumulative number of fitness evaluations required to optimize the whole stream and DFE is the decrease in the number of fitness evaluations compared to full optimization.

Attack	**Database**	**Type**	**No surrogate**		**Off-line**			**On-line**		
		$\boldsymbol{AFPI}$	$\boldsymbol{F_{Evals}}$	$\boldsymbol{AFPI}$	$\boldsymbol{F_{Evals}}$	$\boldsymbol{DFE}$	$\boldsymbol{AFPI}$	$\boldsymbol{F_{Evals}}$	$\boldsymbol{DFE}$	$\boldsymbol{AFPI}$
Cropping 1%	OULU-1999-TRAIN	Positives	1059 (362)	24360	212 (33)	4878	80%	955 (349)	17191	29.4%
		Negatives	941 (336)	16940	212 (45)	3810	77.5%	874 (376)	20110	-18.7%

Table 4.7 Average computational cost performance. $AFPI$ is the average number of fitness evaluations per image. Values in parentheses are standard deviation. F_{Evals} is the cumulative number of fitness evaluations required to optimize the whole stream and DFE is the decrease in the number of fitness evaluations compared to full optimization.

Attack	Database	Learning	Full PSO		GMM-based			DS-DPSO		
			$AFPI$	F_{Evals}	$AFPI$	F_{Evals}	DFE	$AFPI$	F_{Evals}	DFE
Cropping 2%	OULU-1999-TRAIN	No	860 (335)	86040	72 (187)	7240	91.6%	77 (204)	7684	91.1%
	OULU-1999-TEST	No	828 (309)	328900	64 (179)	25560	92.2%	77 (199)	30747	90.7%
	OULU-1999-TEST	Yes	828 (309)	328900	50 (150)	19800	94%	46 (84)	18339	94.4%
S&P 0.02	OULU-1999-TRAIN	No	893 (354)	89280	163 (360)	16320	81.7%	121 (308)	12055	86.5%
	OULU-1999-TEST	No	978 (379)	388220	92 (281)	36360	90.6%	59 (161)	23327	94%
	OULU-1999-TEST	Yes	978 (379)	388220	42 (133)	16560	95.7%	33 (54)	13183	96.6%

Table 4.8 Average watermarking performance, where the mean μ and standard deviation σ of each metric are presented as $\mu(\sigma)$.

Attack	Database	Learning	Full PSO			GMM-based			DS-DPSO		
			DRDM	BCR robust	BCR fragile	DRDM	BCR robust	BCR fragile	DRDM	BCR robust	BCR fragile
Cropping 2%	OULU-1999-TRAIN	No	0.04 (0.05)	98.2 (2.7)	99.9 (0.4)	0.04 (0.06)	97.1 (3.8)	99.8 (0.6)	0.04 (0.05)	96.5 (4.6)	99.7 (0.9)
	OULU-1999-TEST	No	0.04 (0.04)	98.0 (3.0)	99.8 (0.7)	0.04 (0.04)	95.4 (5.7)	99.3 (2.0)	0.04 (0.05)	94.7 (7.4)	99.1 (2.0)
	OULU-1999-TEST	Yes	0.04 (0.04)	98.0 (3.0)	99.8 (0.7)	0.04 (0.05)	94.7 (6.4)	99.1 (1.9)	0.04 (0.06)	96.0 (5.2)	99.8 (0.8)
S&P 0.02	OULU-1999-TRAIN	No	0.03 (0.03)	97.9 (2.6)	99.7 (0.5)	0.03 (0.03)	97.1 (4.3)	99.3 (1.3)	0.03 (0.04)	97.2 (3.4)	99.5 (1.0)
	OULU-1999-TEST	No	0.03 (0.04)	98.0 (2.4)	99.6 (0.6)	0.03 (0.04)	97.2 (3.6)	99.4 (1.0)	0.03 (0.04)	96.7 (4.5)	99.3 (1.1)
	OULU-1999-TEST	Yes	0.03 (0.04)	98.0 (2.4)	99.6 (0.6)	0.03 (0.04)	97.1 (4.0)	99.2 (1.2)	0.03 (0.04)	96.3 (4.3)	99.1 (1.3)

GENERAL CONCLUSION

In this thesis intelligent watermarking in scenarios involving long streams of document images was investigated. The main objective of the research conducted was to find means of decreasing the computational burden of intelligent watermarking in such scenario. This was achieved by a sequence of investigations on some of the essential aspects in tackling intelligent watermarking of long streams of document images.

The first contribution (Chapter I) comprised a literature review on intelligent watermarking. That study allowed identifying some of the main issues in the area. One of these issues was that most intelligent watermarking techniques rely on the use of evolutionary computing to optimize embedding parameters for every image which is very costly for real world applications.

This led to the second contribution (Chapter II) where intelligent watermarking was first formulated as a dynamic optimization problem and a technique that allows replacing costly re-optimization operations with recalls to a memory of static solutions was proposed. That technique was tailored to scenarios involving homogeneous streams of document images. A change detection mechanism, allowed precisely measuring the similarity between new and previous cases of optimizations. With this, ready-to-use solutions stored in the memory could be employed directly in situations where a new problem was similar to a previously seen problem. The benefit of replacing re-optimization with memory recall in such case was demonstrated empirically.

In the third contribution (Chapter III) an adaptive memory scheme was devised, based on the use of Gaussian Mixture Models (GMMs). The use of GMM resulted in memory elements that are less biased to the problems that generated them. Moreover, the proposed memory scheme allowed learning the stream of optimization problems in an incremental manner. This concept of storing density estimates of fitness and parameter information of all solutions found during optimization is to the best of our knowledge, novel. The result was a memory that can adapt to variations in the streams of optimization problems like in scenarios involving heterogeneous image streams. Experimental results demonstrate that such type of memory has

a better learning capability compared to a memory of static solutions for heterogeneous image streams.

Finally, in the fourth contribution (Chapter IV), a strategy that allows employing the memory of GMMs in order to further decrease the cost of intelligent watermarking by replacing fitness evaluations with Gaussian Mixture Regression (GMR) during re-optimization was proposed. The four-level optimization scheme is a consolidation of the research conducted on the previous two approaches. At each level, an attempt to solve the specific problem instance is made. If it fails, another attempt is made, but in a higher level with increasing precision but at a higher computational burden. The last level works as a safeguard to the whole system and at that point, attempts to decrease computational burden are performed in a best case basis. This allowed a machine learning formulation of optimization – approximations of the fitness landscape are first built in a controlled environment and can then be deployed to a test environment where the computational burden constraints are more severe. Experimental results demonstrate that such approach decreases significantly the cost of re-optimization compared to the alternative of completely resetting the population.

Future work

Three main directions to future investigations can be considered:

- *Evaluating the proposed DPSO technique in other recurrent problems.* Numerous real world applications involve optimization of recurrent streams of optimization problems, specially in scenarios involving streamed data like video, audio and images. One of those is tracking moving objects in video sequences. In machine learning, applications requiring optimization of heuristic parameters of classifiers in scenarios involving dynamic data streams is very common.
- *Evaluating other types of watermarking systems.* The motivation for relying on bi-tonal watermarking is that most document analysis applications rely on bi-tonal images. However, the proposed technique sees a watermarking system as a black-box. Therefore, it

would be very interesting to evaluate how does the proposed DPSO technique behaves in grey-scale and/or color watermarking.

- *Synthetic benchmark functions.* One of the main difficulties of real-world problems is that it is hard to fully know their properties. Thus, another possible line of investigation would be to employ a set of recurrent benchmark functions in order to better understand among other things how well can the proposed approach learn a stream of optimization problems, what are its limitations in terms of adaptability.
- *Module improvements/validation.* The final system is considerably modular. Therefore, an interesting research direction would be to evaluate alternatives for some of its modules, including one or more PSO variants. The EC technique employed in the optimization module must be capable of preserving population diversity. Therefore, a study of alternative approaches which can improve the diversity preserving performance of the approach employed on this research would be valuable. A study of alternatives to current GMM and change detection modules would also be valuable.

APPENDIX I

BASELINE BI-TONAL WATERMARKING SYSTEM

1 Overview

The bi-tonal watermarking of Wu and Liu (Wu and Liu, 2004) was chosen as a test case due to its modularity and flexibility. This solution has two main components, embedder and detector. This system can be viewed as a blind communication system, facing an Additive White Gaussian Noise (AWGN) attack (Figure AI.1), where x is the cover signal, m is the message to be encoded. w is the encoded message, to be embedded as a watermark, s is the watermarked signal, v is the Additive White Gaussian Noise, r is the watermarked signal after being attacked and $\hat{m}$ is the detected message.

The two main components of this system are the watermark embedder and the watermark detector. During embedding, the cover image is partitioned into blocks of equal size and each bit of the message (watermark) is embedded on each of these blocks through manipulation of the quantity of black pixels. Detection is the reverse process: the watermarked image is partitioned on blocks of the same size employed on embedding and each bit is decoded from each block by computing the number of black pixels for that block. The main advantage of a blind watermarking system is that the original (cover) image is not required during detection. In Wu and Liu's system, this is attained by setting the number of black pixels per block to be either an even number (to embed a '0') or and odd number (to embed a '1'). This is known as odd/even embedding and its main advantage is that only a few embedding parameters (such as the block size) need to be known by the detector which makes it very good choice for distributed applications. However, the main limitation of odd/even embedding is that for any given block, the

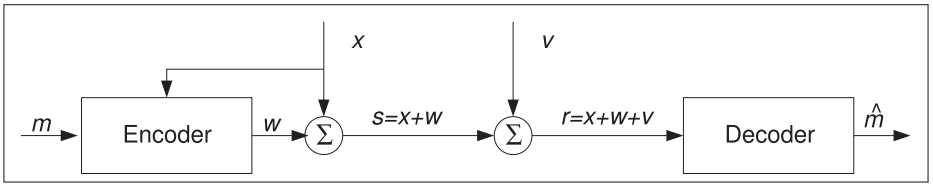


Figure AI.1 Blind watermarking system viewed as a communication problem.

Figure AI.2 Illustration of multi-level watermarking. (a) Two watermarks are embedded into a bank cheque. (b) The bank cheque is tampered but the robust watermark can be recovered while the fragile is destroyed.

value of a embedded bit can be modified by merely flipping one black pixel in that block. To cope with this, Wu and Liu proposed quantizing the number of black pixels based on a quantization step size. Such strategy allows embedding multiple watermarks with different levels of robustness to cope with different aspects of image security. Figure AI.2 illustrates a typical application involving multi-level watermarking. In Figure AI.2a, two watermarks (a robust and a fragile) are embedded into a bank cheque. Then in Figure AI.2b the numerical amount of the watermarked cheque is modified (from $40.00 to $990.00, a clear case of fraud). The fragile watermark is destroyed, which allows detecting that tampering has occurred while the robust watermark resists the attack allowing for example to identify which person or institution was the legal owner of that document image.

The watermarking process is subject to a trade-off between watermark robustness and image quality, which can be seen as an optimization problem. In this thesis, different techniques are proposed to allow the optimization of embedding parameters for long streams of document images. One of they key concepts regards the use of Particle Swarm Optimization (PSO) to tune the watermarking parameters for a given image and pair of watermarks. This concept is depicted in Figure AI.3.

Below, the key elements of the approach employed in this thesis are described in details.

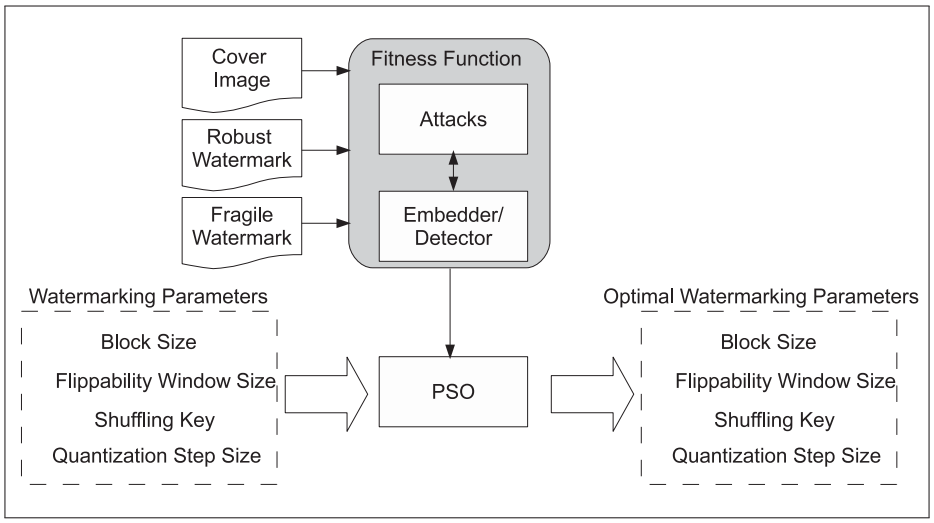


Figure AI.3 Watermarking as an optimization problem.

1.1 Watermark embedder

1.1.1 Identification of flippable pixels

Since randomly flipping black pixels can lead to visual artifacts, numerous bi-tonal watermarking systems employ some sort of flippability analysis technique which provides a ranking of the pixels, based on how perceptible will be flipping them from black to white or vice-versa. Therefore, this is one of the first steps of bi-tonal watermarking and is a process that only needs to be performed on the embedder (as detection does not require flipping pixel values). Wu and Liu (Wu and Liu, 2004) employ a flippability analysis technique based on the use of look-up tables. However, such approach lacks flexibility in a scenario involving the optimization of embedding parameters with the use of evolutionary computing. Muharemagic (Muharemagic, 2004) proposes a more flexible flippability metric named Structural Neighborhood Distortion Measure (SNDM). This method uses a reciprocal distance matrix D_m in order to compute the flippability of a pixel, based on its $m \times m$ neighbourhood.

The SNDM of a candidate pixel (cp) is computed as follows:

$$SNDM_{cp} = \frac{(cp \ \oplus \ N_m) \bullet D_m}{\mid D_m \mid} \tag{18}$$

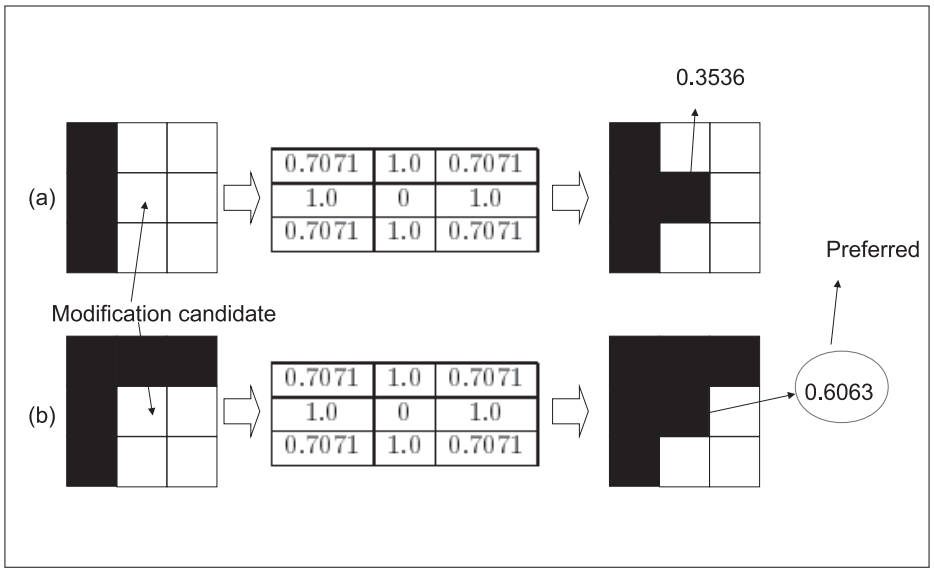


Figure AI.4 Illustration of SNDM flippability analysis (Muharemagic, 2004).

where N_m represents the $m \times m$ neighborhood of cp.

Figure AI.4 from (Muharemagic, 2004) illustrates the flippability analysis process for two different image blocks of size 3×3. In both cases, the pixel being analyzed is located at the center of the 3×3 window. It is clear that the flipping the value of the central pixel in Figure AI.4a will be much more perceptible than flipping the value of the central pixel in Figure AI.4b. Consequently, the SNDM score for the block in Figure AI.4b is greater than that of Figure AI.4a.

1.1.2 Shuffling of image pixels

Since one of the main uses of binary images is in the processing and storage of document images, these images contain vast amount of white spaces, with little or no embedding pixels. As will be shown later, the embedding is done per image block and if the pixel distribution is uneven, some blocks will contain no embedding pixels, which will reduce the embedding capacity for some blocks. Wu and Liu (Wu and Liu, 2004) demonstrated that shuffling allows distributing the flippable pixels equally across the embedding blocks, leading to an optimal embedding capacity. Shuffling consists of randomly shifting pixel positions across the image. Figure AI.5 from (Wu and Liu, 2004) illustrates the effect of shuffling in the distribution of

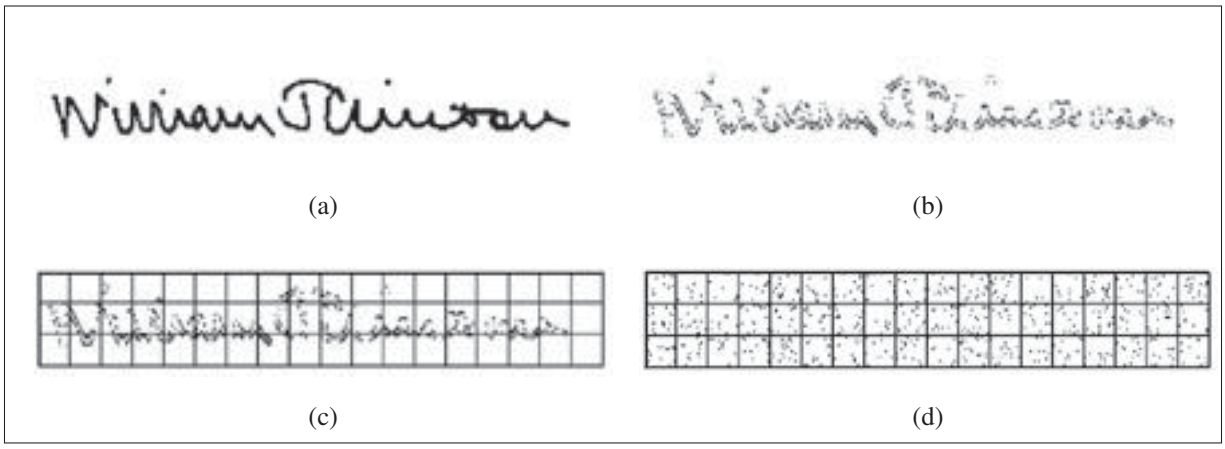


Figure AI.5 Effect of shuffling on distribution of flippable pixels (Wu and Liu, 2004). (a) Sample image (President Bill Clinton's signature). (b) Flippable pixels. (c) Partition block and flippable pixels before shuffling. (d) Partition block and flippable pixels after shuffling.

flippable pixels. It is possible to observe that before shuffling, some blocks contain no flippable pixel at all (Figure AI.5c). After shuffling, all blocks contain approximately the same amount of flippable pixels (Figure AI.5d).

Muharemagic (Muharemagic, 2004) proposed a method that performs the random shifting with a $O(N)$ complexity. The method works as follows (image I of width w and height h is represented in a single dimension of length $N = w \times h$):

a. A shuffling key S is created. This key will contain the mapping of the original pixel co-ordinates to the shuffled co-ordinates. This array can be initialized with its index. Starting with the last element, a random index (with value smaller than current index) is chosen. The value of the the current position is flipped with the value pointed by the random index. Figure AI.6 depicts this process. In this example, a 5-elements array is initialized with index values. Then a random number r is chosen. The current array value is flipped with the value indexed by r. The process is repeated for every element (towards the first).

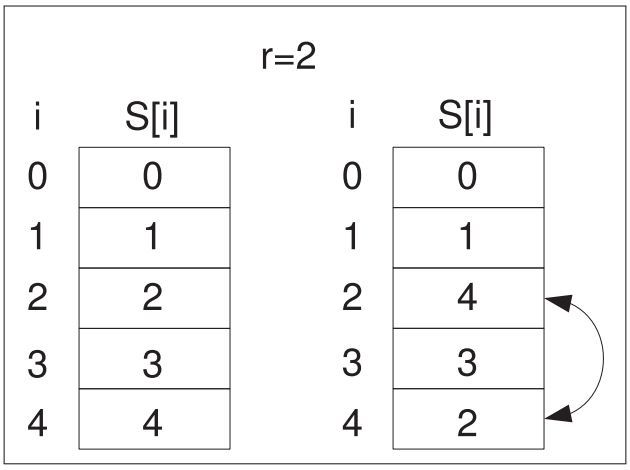


Figure AI.6 Shuffling key generation.

b. Mapping is applied to image I. Swap pixel using the shuffling key:

$$I[i] \leftrightarrow I[S[i]] \; \forall \; I[i]. \tag{19}$$

Since a pseudo-random number generator is used in order to create the key, the seed is enough for re-generating the key on the detector side.

1.1.3 Partitioning of the image into blocks

The image is divided into blocks of equal size. This process makes possible embedding a multi-bit message into a given cover image as each bit is embedded into each block of the cover image. Is important to notice that the same block size must be employed on embedding and detection.

1.1.4 Conversion of watermark to a bit stream

Here the message to be embedded (which can be a logo, an integer number or a text string) must be converted to a bit stream. Knowledge of message length (as long as about what type of message is embedded) must be available at the detector. For example, throughout this thesis, two watermarks (a robust and a fragile) of 936 bits are embedded and their dimensions are known at both, the embedder and at the detector.

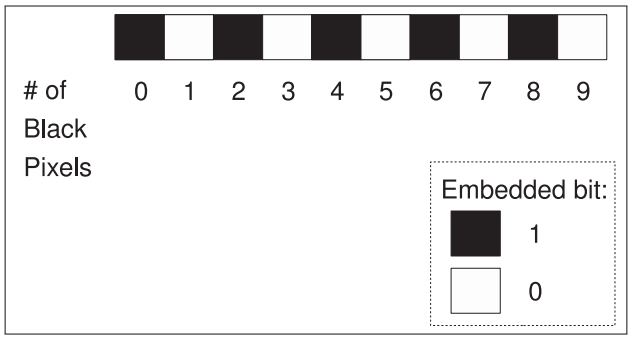


Figure AI.7 Simple odd-even embedding.

1.1.5 Embedding of the bit stream into cover image with the use of Uniform Quantization (UQ) (Chen and Wornell, 2001) (Eggers *et al.*, 2003)

The embedding of the bit stream into the cover image is performed in a one bit per block basis. The feature used to embed a bit is the quantity of black pixels in the block. The *naïve* approach is to force the quantity of black pixels to be even in order to embed a given value (e.g. '1') or odd to embed another (e.g. '0'). A pixel with a high flippability score can be flipped in order to force this property. However this approach has a drawback: once a single pixel is changed, the embedded bit will also change. This approach is not practical since it does not allow robust embedding. Figure AI.7 depicts this embedding scheme. It can be seen that, any change on the quantity of black pixels will change the embedded value.

An alternative is to quantize the number of black pixels, using a given quantization step (Q). In this case, the quantity of black pixels must be $2kQ$ to embed a '1' or $(2k+1)Q$ to embed a '0'. This has the effect of creating an "embedding bin", that is the quantity of black pixels can float in the $\pm Q/2$ range without affecting the embedded value. This approach is illustrated on Figure AI.8.

Eggers *et al* (Eggers *et al.*, 2003) Scalar Costa Scheme (SCS) is a generalization of Chen and Wornell (Chen and Wornell, 2001) and for that reason, it was the approach employed in this thesis. In this method, in order to embed a bit d_n into a given element of the cover signal x_n (in this case, quantity of black pixels of block n), a quantization of the cover signal must be done

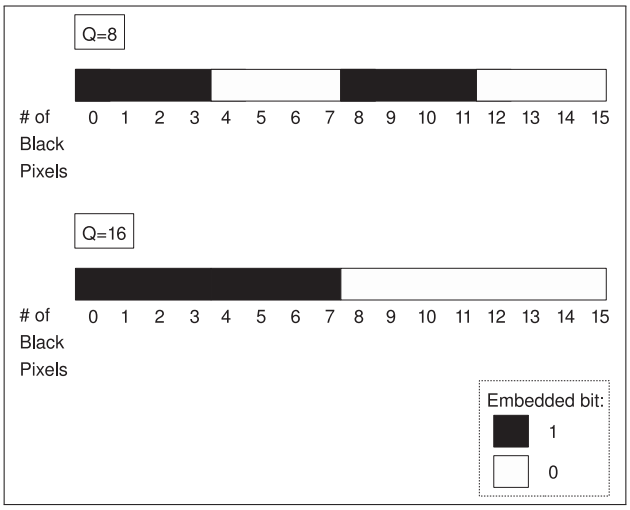


Figure AI.8 Uniform quantization.

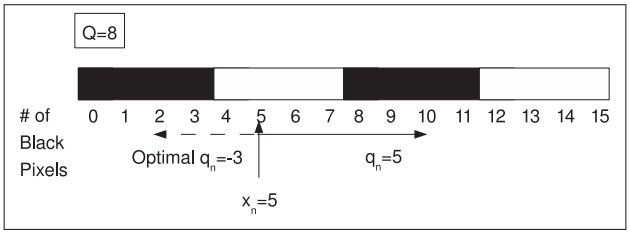


Figure AI.9 Optimization on SCS scheme.

first

$$q_n = Q_\Delta\{x_n - \Delta(\frac{d_n}{D} + k_n)\} - (x_n - \Delta(\frac{d_n}{D} + k_n)) \tag{20}$$

where $Q_\Delta\{\}$ is the scalar uniform quantization operation, Δ is the quantization step size (Q), D is the alphabet size (2 for binary encoding), and k_n is a pseudo-random number in the $[0, 1)$ range, used for security purpose.

It is possible to do a small optimization on this algorithm. If modulo of q_n is greater than $\Delta/2$, it means that it is possible to embed the given bit by "targeting" the opposite neighbor bin. That can be done by subtracting Δ from the modulo of q_n. Figure AI.9 shows an hypothetical situation, where $|q_n| > \Delta/2$ (in this case, $q_n = 5$). Subtracting Δ from q_n allows embedding the given bit, but by changing only three pixels rather than five.

The transmitted watermark sequence is obtained by multiplying q_n by the embedding strength α according to Eggers *et al* (Eggers *et al.*, 2003), Chen and Wornell (Chen and Wornell, 2001) can be seen a special case of SCS where $\alpha = 1.0$

$$w = \alpha q \tag{21}$$

Finally, the watermark w is added to cover signal x

$$s = x + w \tag{22}$$

In this given application, s_n will represent the number of black pixels in block n that is necessary in order to embed bit d_n. That is, the flippable pixels on block n must be changed accordingly (if $s_n > x_n$, it is necessary to flip $s_n - x_n$ white pixels, if $s_n < x_n$, $x_n - s_n$ black pixels to white, otherwise no pixel has to be flipped). The pixels are first sorted according to their flippability score, and those with higher score are flipped first, until the condition $s_n = x_n$ is obtained. If such condition is not obtained, embedding of bit d_n on block n with quantization step Q does not occurr.

Since this method handles real values (k_n), it is necessary to round the value of s_n. The rounding error can be easily recovered on the detector side. The flippable pixels of block n must be changed accordingly in order to force the number of black pixels to be equal to s_n.

The parameter α is related with the choice of Δ and the given watermark power σ_w^2

$$\alpha = \frac{\sigma_w \sqrt{12}}{\Delta} \tag{23}$$

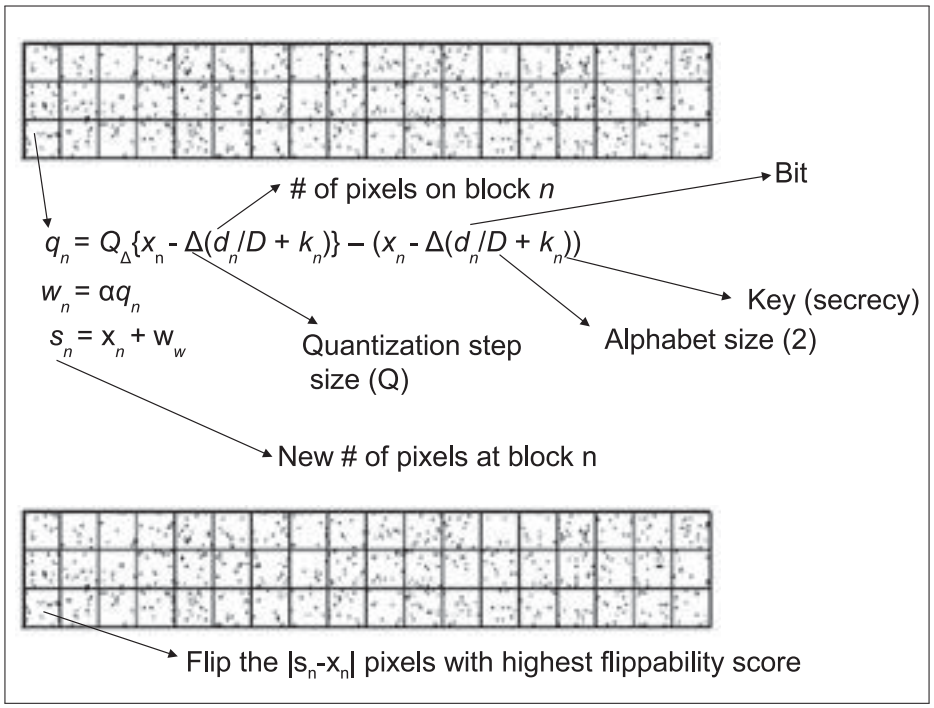


Figure AI.10 Illustration of Uniform Quantization (UQ) embedding.

The authors also present a manner of calculating approximated optimal values for α and Δ for given watermark power (σ_w^2) and noise power (σ_v^2)

$$\alpha_{SCS,approx} = \sqrt{\frac{\sigma_w^2}{\sigma_w^2 + 2.71\sigma_v^2}} \tag{24}$$

$$\Delta_{SCS,approx} = \sqrt{12(\sigma_w^2 + 2.71\sigma_v^2)} \tag{25}$$

The UQ embedding process is illustrated in Figure AI.10.

1.1.6 De-shuffling of watermarked image

The image is de-shuffled by applying the reverse operation of Equation 19.

1.2 Watermark detector

1.2.1 Shuffling

The same process used during encoding is applied. The main issue here is shuffling key distribution, since the key must be exactly the same as used on encoding. As mentioned before, a reasonable alternative is to distribute only the seed (since the key generation relies on a pseudo-random number sequence).

1.2.2 Partitioning

The same partitioning process applied on encoding is used on decoding.

1.2.3 Detection of the bit stream on cover image with the use of Uniform Quantization (UQ) (Chen and Wornell, 2001) (Eggers *et al.*, 2003)

Here the reverse process is applied to the received signal r (which contains the watermark w and the noise signal v), that is $r = x + w + v$. Firstly, y_n, is obtained with the use of the uniform quantizer

$$y_n = Q_\Delta\{r_n - k_n\Delta\} - (r_n - k_n\Delta) \tag{26}$$

After that, a linear decision is applied in order to extract the bit value from y_n. If the value of y_n is close to either Q or 0, it means the corresponding bit is $d_n = 0$. If the instead, the value of y_n is close to $Q/2$, it means the corresponding bit is $d_n = 1$. Figure AI.11 depicts the decision function.

The UQ detection process is illustrated in Figure AI.12.

1.2.4 Reconstruction of watermark with the use of detected bit stream

In the given application, the dimensions of the embedded logo image must be known on the detector. With this, it is possible to reconstruct the logo using the content of the bit stream $\hat{m}$.

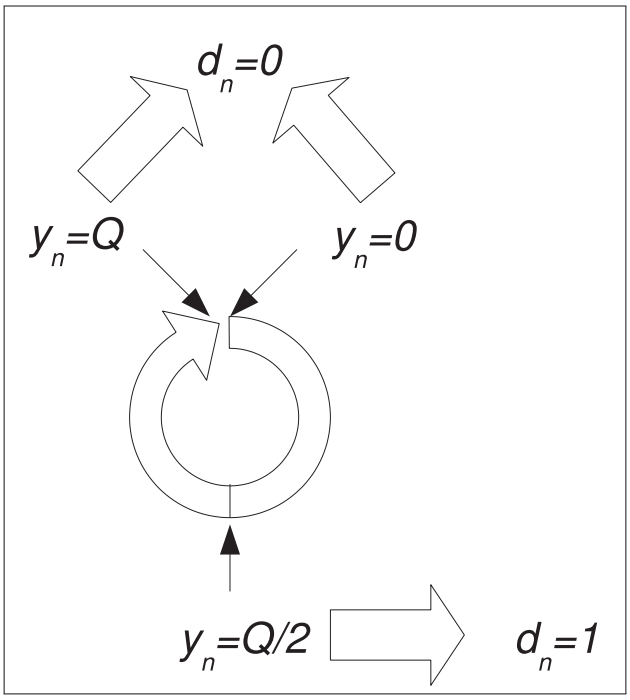


Figure AI.11 Detection decision.

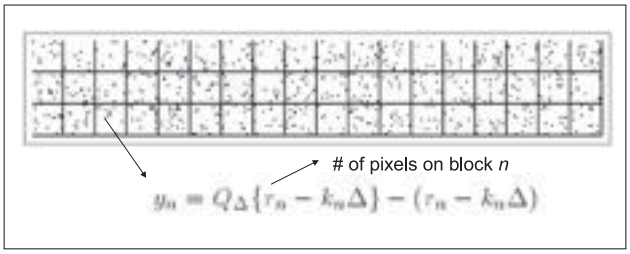


Figure AI.12 Illustration of Uniform Quantization (UQ) detection.

2 Evaluating watermarking performance

In the communication model of digital watermarking, embedding of a given message into an image is constrained by the robustness of the embedded watermark against image processing operations and the impact of the watermarking process on image quality. Such trade-off can be adjusted through manipulation of embedding parameters but is constrained by the embedding capacity of each image (that is the main reason for the widespread use of evolutionary computing to find such trade-off for each image).

2.1 Visual impact

Three main metrics can be employed to assess the visual impact of such bi-tonal watermarking system (Muharemagic, 2004), namely Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR) and Distance Reciprocal Distortion Measure (DRDM).

2.1.1 MSE

In order to compute the MSE for a binary image, it is necessary to first convert the grey-level encoding, where white pixels value is equal to 255 and black pixel value is equal to 0 to a binary encoding (all white pixels are set to 0 and all black pixels are set to 1). After that, for each pixel of the two images (original or cover I_O and watermarked I_W), the square of their difference is computed and summed and the result is divided by the number of pixels.

$$MSE(I_W, I_O) = \frac{1}{n}\sum_{N}(i_W[i] - i_O[i])^2 \tag{27}$$

2.1.2 PSNR

PSNR is used to evaluate the relation between the maximum signal (in the binary image case 1) and the noise caused by the embedding process. It is expressed using the logarithmic decibel scale.

$$PSNR(I_W, I_O) = 10log_{10}(\frac{1}{\sum_N(i_W[i] - i_O[i])^2}) \tag{28}$$

2.1.3 DRDM

The Distance-Reciprocal Distortion Measure (DRDM) (Lu *et al.*, 2004) is a distortion metric specifically proposed to evaluate the distortion between two binary images. Changes in a binary image may affect the structure of elements within that image and this type of change affects drastically the quality of the image. For this reason, care must be taken in order to avoid such changes. The DRDM is based in the assumption that changes in pixels close to viewer's focus are more noticeable. Also, due to particularities of HVS, changes in diagonal neighbors of

a pixel are less noticeable than changes on its immediate vertical and horizontal neighbors (4-neighborhood).

A normalized weight matrix W_m, with size $m \times m$ is used to compute the distortion between two binary images. Each element of this matrix represents the reciprocal distance, relative to the center pixel. The distortion between two binary images is calculated as:

$$d = \frac{\sum d_k}{K} \tag{29}$$

where K is the number of non-uniform (not all black or all white pixels) blocks and d_k is the distortion calculated for a given pixel with the use of a $m \times m$ window

$$d_k = \sum_{m \times m} [|a_m - b_m| \times W_m] \tag{30}$$

2.2 Capacity

Capacity is usually computed in comparison with another metric. One of the tools based on this concept is the capacity versus Watermark-to-Noise Ratio curve (Figure AI.13).

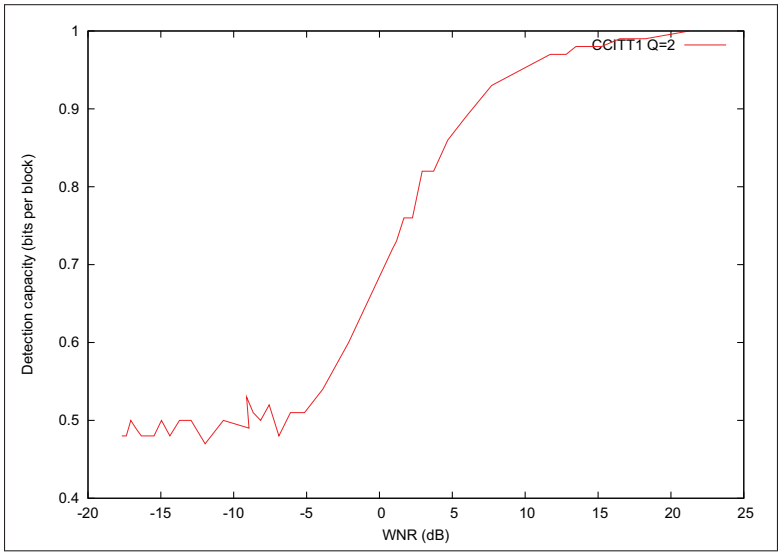


Figure AI.13 Example of capacity versus watermark-to-noise ratio curve ($Q = 2$, CCITT-1 image).

APPENDIX II

EMPIRICAL RUNTIME PERFORMANCE

Although the standard approach in the evolutionary computing literature is to report computational cost performance in terms of number of fitness evaluations, when it comes to practical applications it is important know how this fitness evaluation performance translates in terms of CPU time. For this purpose, the CPU time performance of Full PSO, case-based and GMM-based approaches are provided in this appendix. For the Full PSO and case-based approaches the main aspect defining the the total CPU time to optimize an image stream is the number of fitness evaluations and their respective CPU time. For the GMM-based and DS-DPSO approaches, there is the additional cost of training GMMs. Put differently, the processing time for Full PSO and case-based approaches is just a factor of the number of fitness evaluations while for the GMM-based and DS-DPSO it is a factor of the number of fitness evaluations and the number of re-optimizations (since every re-optimization involves training a new GMM).

It is important to observe that the all experiments in this thesis were performed in a computer cluster containing 17 nodes. Each of these nodes contain an Intel ® Core 2 ™ Quad Q6600 CPU with four cores of 2.4GHz and 7 GB of memory. All prototypes were coded in C++ and rely on the Message Passing Interface (MPI) for parallelization. The architecture chosen was the master-slave where a master node sends work for slave nodes. This means that all PSO and memory management operations are performed by the master in a sequential manner while the slaves perform fitness evaluations (watermark embedding and detection, attacks, BCR and DRDM computation) in a parallel manner. Considering that the population size employed in all simulations is 20 particles, this means that in this scenario, a total of 21 nodes are required (one master and 20 slaves).

Full PSO CPU time performance is summarized in Table AII.1. For each case, the total CPU time, the average CPU time per fitness evaluation plus the average CPU time per image are reported. It is important to remark that these results were not scaled by the number of nodes (they reflect effectively the time it took to run the simulations). It is possible to observe that in

Table AII.1 Details of computational performance (CPU time) of Full PSO. CPU time per fitness and per image are presented in the following form: mean (standard deviation).

Variant	Dataset	Attack	CPU time (seconds)		
			Total	Per fitness	Per image
Chapter 2	TITI-61	No Attack	11400	0.22 (0.13)	187 (51)
Chapter 2	TITI-61	Cropping 1%	16380	0.29 (0.16)	269 (72)
Chapter 2	CVIU-113-3-4	No Attack	55380	0.18 (0.11)	162 (43)
Chapter 2	CVIU-113-3-4	Cropping 1%	70620	0.24 (0.14)	206 (55)
Chapter 3	OULU-1999-TRAIN	No Attack	26820	0.29 (0.42)	268 (180)
Chapter 3	OULU-1999-TRAIN	Cropping 1%	26220	0.30 (0.42)	262 (181)
Chapter 3	OULU-1999-TEST	No Attack	108180	0.27 (0.37)	272 (181)
Chapter 3	OULU-1999-TEST	Cropping 1%	94680	0.28 (0.40)	238 (166)

general, it takes between 3 and 4.5 minutes to optimize each image. In this parallel configuration, the average time per fitness evaluation is around 0.3 per second (the throughput is close to 3 fitness evaluations per second). The total time to optimize a whole image stream varies from 3 to 30 hours.

The CPU time performance obtained for Full PSO translate directly for the case-based approach since the only relevant additional cost here is the cost of fitness evaluations for recall (the cost of memory update is negligible). Table AII.2 summarizes the CPU time performance for such scenario. Here it is possible to observe that the CPU time per image varies from 6 to 10 seconds for homogeneous streams while it varies from 42 to 164 seconds for heterogeneous streams. Here the total time to optimize an image stream varies from 10 minutes (homogeneous streams) to 15 hours (heterogeneous streams).

CPU time performance for the GMM-based approach is presented in Table AII.3 Here it is possible to observe that in the current configuration, GMM training adds a considerable processing time to the system since it has not been parallelized. This cost itself would be negligible if this process had been also parallelized, but in the given architecture, GMM training is performed in a sequential manner. Still, when it comes to heterogeneous image streams, this additional time is worth when compared to the case-based approach. The total CPU time to optimize streams of heterogeneous images varied from 0.7 to 3.4 hours for the GMM-based approach versus 3 to 15 hours for the case-based approach.

Table AII.2 Details of computational performance (CPU time) of case-based approach.

Variant	Dataset	Attack	Learning	CPU time (seconds)	
				Total	Per image
Chapter 2	TITI-61	No Attack	No	607	10
Chapter 2	TITI-61	Cropping 1%	No	1148	19
Chapter 2	CVIU-113-3-4	No Attack	No	1930	6
Chapter 2	CVIU-113-3-4	No Attack	Yes	1901	6
Chapter 2	CVIU-113-3-4	Cropping 1%	No	2098	6
Chapter 2	CVIU-113-3-4	Cropping 1%	Yes	2035	6
Chapter 3	OULU-1999-TRAIN	No Attack	No	16350	164
Chapter 3	OULU-1999-TRAIN	Cropping 1%	No	10530	105
Chapter 3	OULU-1999-TEST	No Attack	No	28960	73
Chapter 3	OULU-1999-TEST	No Attack	Yes	49729	125
Chapter 3	OULU-1999-TEST	Cropping 1%	No	19684	50
Chapter 3	OULU-1999-TEST	Cropping 1%	Yes	16503	42

Table AII.3 Details of computational performance (CPU time) of GMM-based approach.

Dataset	Attack	Learning	Total CPU time (seconds)		
			Optimization/recall	GMM training	Combined
OULU-1999-TRAIN	No Attack	No	1908	533	2439
OULU-1999-TRAIN	Cropping 1%	No	5358	2481	7839
OULU-1999-TEST	No Attack	No	6286	3715	10001
OULU-1999-TEST	No Attack	Yes	4509	1168	5677
OULU-1999-TEST	Cropping 1%	No	9218	3194	12412
OULU-1999-TEST	Cropping 1%	Yes	7493	2509	10002

BIBLIOGRAPHY

Areef, T. E., H. S. Heniedy, and O. M. O. Mansour. 2005. "Optimal transform domain watermark embedding via genetic algorithms". *ITI 3rd International Conference on Information and Communications Technology (ICICT).*, p. 607–617.

Arsalan, M., S. A. Malik, and A. Khan. 2010. "Intelligent threshold selection for reversible watermarking of medical images". In *GECCO '10: Proceedings of the Genetic and Evolutionary Computation Conference*. p. 1909–1914. ACM.

Arsalan, M., S. A. Malik, and A. Khan. 2012. "Intelligent reversible watermarking in integer wavelet domain for medical images". *Journal of Systems and Software*, vol. 85, n. 4, p. 883 – 894.

Awan, I., S. A. M. Gilani, and S. A. Shah. 2006. "Utilization of Maximum Data Hiding Capacity in Object-Based Text Document Authentication". In *International Conference on Intelligent Information Hiding and Multimedia (IIH-MSP)*. (Washington, DC, USA 2006), p. 597–600.

Banks, A., J. Vincent, and C. Anyakoha. 2008. "A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications". *Natural Computing*, vol. 7, n. 1, p. 109–124.

Barni, M. and F. Bartolini, 2004. *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*. CRC Press.

Bishop, C. M., 1996. *Neural Networks for Pattern Recognition*. Oxford University Press.

Blackwell, M. 2005. "Particle swarms and population diversity". *Soft Comput.*, vol. 9, n. 11, p. 793–802.

Blackwell, T., 2007. *Evolutionary Computation in Dynamic Environments*, chapter Particle swarm optimization in dynamic environments. Springer.

Blackwell, T. M. and P. J. Bentley. 2002. "Dynamic Search With Charged Swarms". In *GECCO '02: Proceedings of the Genetic and Evolutionary Computation Conference*. (San Francisco, CA, USA 2002), p. 19–26. Morgan Kaufmann Publishers Inc.

Blekas, K. and I. E. Lagaris. 2007. "Split-Merge Incremental Learning (SMILE) of Mixture Models.". In *ICANN*. p. 291–300.

Branke, J. 1999. "Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems". In *Congress on Evolutionary Computation (CEC)*. p. 1875–1882. IEEE.

Calinon, S., 2009. *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press.

Carlisle, A. and G. Dozier. 2002. "Tracking changing extrema with adaptive particle swarm optimizer". *Proceedings of the 5th Biannual World Automation Congress, 2002.*, vol. 13, p. 265-270.

Chang, C. and I. Lin, 2004a. *Intelligent Watermarking Techniques*, chapter Robust Image Watermarking Systems Using Neural Networks. World Scientific Co.

Chang, C. and I. Lin, 2004b. *Intelligent Watermarking Techniques*, chapter A Perceptually Tuned Watermarking Scheme for Digital Images Using Support Vector Machines. World Scientific Co.

Chen, B. and G.W. Wornell. 2001. "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding". *IEEE Transactions on Information Theory*, vol. 47, n. 4, p. 1423-1443.

Chen, C. and C. Lin. 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, M., E. K. Wong, N. Memon, and S. Adam. 2001. "Recent developments in document image watermarking and data hiding". *Proc. SPIE*, vol. 4518, p. 166-176.

Clerc, M., 2006. *Particle Swarm Optimization*. London : ISTE Publishing Company.

Coello, C., G.T. Pulido, and M.S. Lechuga. June 2004. "Handling multiple objectives with particle swarm optimization". *Evolutionary Computation, IEEE Transactions on*, vol. 8, n. 3, p. 256-279.

Collette, Y. and P. Siarry. 2008. "On the sensitivity of aggregative multiobjective optimization methods". *CIT*, vol. 16, n. 1, p. 1–13.

Corriveau, G., R. Guibault, A. Tahan, and R. Sabourin. 2012. "Review and study of genotypical diversity measures for real-coded representations.". *IEEE Transactions on Evolutionary Computation*, vol. 16, n. 5, p. 695-710.

Corriveau, G., R. Guibault, A. Tahan, and R. Sabourin. 2013. "Review of Phenotypic Diversity Formulations for the Development of an Optimizer Diagnostic Tool.". *Applied Soft Computing*, vol. 13, n. 1, p. 9-26.

Costa, M. May 1983. "Writing on dirty paper (Corresp.)". *Information Theory, IEEE Transactions on*, vol. 29, n. 3, p. 439-441.

Cox, I. J., J. Kilian, T. Leighton, and T. Shamoon. 1996. "A Secure, Robust Watermark for Multimedia". In *Workshop on Information Hiding*. p. 1–16.

Cox, I., M.L. Miller, and J.A. Bloom, 2002. *Digital Watermarking*. Morgan Kaufmann Publishers.

Davis, K. J. and K. Najarian. 2001. "Maximizing strength of digital watermarks using neural networks". In *Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on*. p. 2893-2898.

Deb, K., 2001. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. "A fast and elitist multiobjective genetic algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation*, vol. 6, p. 182-197.

Dennis, J. E. and V. Torczon. 1995. "Managing Approximation Models in Optimization". In *Multidisciplinary Design Optimization: State-of-the-Art*. p. 330–347.

Díaz, D. S. and M. G. Romay. 2005. "Introducing a watermarking with a multi-objective genetic algorithm". In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*. (New York, NY, USA 2005), p. 2219-2220. ACM.

Duda, R. O., P. E. Hart, and D. G. Stork, 2000. *Pattern Classification*. Wiley-Interscience Publication.

Eggers, J., J.K. Su, and B. B. Girod. 2003. "Scalar Costa Scheme for Information Hiding". *IEEE Transactions on Signal Processing*, vol. 51, n. 4.

El-Beltagy, M., P. B. Nair, and A. J. Keane. 1999. "Metamodeling Techniques For Evolutionary Optimization of Computationally Expensive Problems: Promises and Limitations". In *GECCO '99: Proceedings of the 8th annual conference on Genetic and evolutionary computation*. p. 196–203.

Engel, P. M. and M. R. Heinen. 2010. "Incremental Learning of Multivariate Gaussian Mixture Models". In *SBIA*. p. 82–91.

Farina, M., K. Deb, and P. Amato. 2004. "Dynamic multiobjective optimization problems: test cases, approximations, and applications". *IEEE Transactions on Evolutionary Computation*, vol. 8, n. 5, p. 425–442.

Figueiredo, M. A. T. and A. K. Jain. 2000. "Unsupervised Learning of Finite Mixture Models". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, p. 381–396.

Fonseca, L., H. Barbosa, and A. Lemonge. 2009. "A similarity-based surrogate model for enhanced performance in genetic algorithms". *OPSEARCH*, vol. 46, p. 89–107.

Gennari, J. H., P. Langley, and D. Fisher. 1989. "Models of incremental concept formation". *Journal of Artificial Intelligence*, vol. 40, p. 11–61.

Gräning, L., Yaochu Jin, and Bernhard Sendhoff. 2005. "Efficient evolutionary optimization using individual-based evolution control and neural networks: A comparative study". In *ESANN*. p. 273–278.

Hennig, C. 2010. "Methods for merging Gaussian mixture components". *Advanced Data Analysis and Classification*, vol. 4, p. 3–34.

Ho, A. T. S., N. B. Puhan, P. Marziliano, A. Makur, and Y. L. Guan. 2004a. "Perception Based Binary Image Watermarking". In *2004 IEEE International Symposium on Circuits and Systems (ISCAS)*. p. 23-26.

Ho, A., N.B. Puhan, P. Marziliano, A. Makur, and Y.L. Guan. May 2004b. "Perception based binary image watermarking". In *International Symposium on Circuits and Systems (ICAS)*. p. 37–40.

Holland, J. H., 1992. *Adaptation in natural and artificial systems*. Cambridge, MA, USA : MIT Press.

Jain, A. K., M. N. Murty, and P. J. Flynn. 1999. "Data clustering: a review". *ACM Computing Surveys*, vol. 31, n. 3, p. 264–323.

Jain, S., S. Lange, and S. Zilles. 2006. "Towards a better understanding of incremental learning". *Algorithmic Learning Theory*, vol. 4264, n. 10, p. 169–183.

Japkowicz, N., C. Myers, and M. Gluck. 1995. "A Novelty Detection Approach to Classification". In *Proceedings of the Fourteenth Joint Conference on Artificial Intelligence*. p. 518–523.

Ji, R., H. Yao, S. Liu, and L. Wang. 2006. "Genetic Algorithm Based Optimal Block Mapping Method for LSB Substitution". In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*. p. 215–218.

Jin, Y., M. Olhofer, and B. Sendhoff. 2000. "On evolutionary optimization with approximate fitness functions". In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO*. p. 786–793. Morgan Kaufmann.

Jin, Y., M. Olhofer, and B. Sendhoff. 2002. "A framework for evolutionary optimization with approximate fitness functions". *IEEE Transactions on Evolutionary Computation*, vol. 6, n. 5, p. 481–494.

Kapp, M. N., R. Sabourin, and P. Maupin. 2009. "A PSO-based framework for dynamic SVM model selection". In *GECCO 2009*. (Montreal, Québec, Canada 2009), p. 1227–1234. ACM.

Kapp, M. N., R. Sabourin, and P. Maupin. 2011. "A Dynamic Model Selection Strategy for Support Vector Machine Classifiers". *Applied Soft Computing (accepted for publication)*.

Kennedy, J. 2007. "Some issues and practices for particle swarm". In *Proc. IEEE Swarm Intelligence Symposium*.

Kennedy, J. and R. Eberhart. November/December 1995. "Particle Swarm Optimization". In *IEEE International Conference on Neural Networks*. (Perth, Australia 1995).

Kennedy, J. and R.C. Eberhart. Oct 1997. "A discrete binary version of the particle swarm algorithm". *Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation'., 1997 IEEE International Conference on*, vol. 5, p. 4104-4108.

Khan, A. and A. M. Mirza. 2007. "Genetic perceptual shaping: Utilizing cover image and conceivable attack information during watermark embedding". *Information Fusion*, vol. 8, n. 4, p. 354–365.

Khan, A., S. F. Tahir, A. Majid, and T. Choi. 2008. "Machine learning based adaptive watermark decoding in view of anticipated attack". *Pattern Recognition*, vol. 41, n. 8, p. 2594 – 2610.

Kumsawat, P., K. Attakitmongcol, and A. Srikaew. 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, Z., S. Lin, S. Su, and C. Lin. 2007. "A hybrid watermarking technique applied to digital images". *Applied Soft Computing*, vol. 8, n. 1, p. 798-808.

Li, X. and J. Wang. 2007. "A steganographic method based upon JPEG and particle swarm optimization algorithm". *Information Sciences*, vol. 177, n. 15, p. 3099-3109.

Lu, H., X. Shi, Y. Q. Shi, A. C. Kot, and L. Chen. 2002. "Watermark embedding in DC components of DCT for binary images". In *IEEE Workshop on Multimedia Signal Processing*. p. 300–303.

Lu, H., A. C. Kot, and Y. Q. Shi. February 2004. "Distance-reciprocal distortion measure for binary document images". *IEEE Signal Processing Letters*, vol. 11, n. 2, p. 228–231.

Ma, J. and S. Perkins. 2003. "Online novelty detection on temporal sequences". In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining*. (New York, NY, USA 2003), p. 613–618. ACM.

Marchand-Maillet, S. and Y. M. Sharaiha, 2000. *Binary digital image processing - a discrete approach.* Academic Press.

Markou, M. and S. Singh. 2003a. "Novelty detection: a review–part 1: statistical approaches". *Signal Processing*, vol. 83, n. 12, p. 2481 – 2497.

Markou, M. and S. Singh. 2003b. "Novelty detection: a review–part 2: neural network based approaches". *Signal Processing*, vol. 83, n. 12, p. 2499 – 2521.

Mei, Q., E. K. Wong, and N. Memon. Jan. 2001. "Data hiding in binary text documents". In *SPIE Proc. Security and Watermarking of Multimedia Contents III*. (San Jose, USA Jan. 2001).

Muharemagic, E. December 2004. "Adaptive Two-Level Watermarking for Binary Document Images". PhD thesis, Florida Atlantic University.

Nickabadi, A., M. M. Ebadzadeh, and R. Safabakhsh. June 2008. "DNPSO: A Dynamic Niching Particle Swarm Optimizer for multi-modal optimization". In *IEEE World Congress on Computational Intelligence*. p. 26-32.

NIST/SEMATECH. March 2010. "NIST/SEMATECH e-Handbook of Statistical Methods". http://www.itl.nist.gov/div898/handbook/.

ÓRuanaidh, J. J. K. and T. Pun. 1998. "Rotation, scale and translation invariant spread spectrum digital image watermarking". *Signal Process.*, vol. 66, n. 3, p. 303-317.

Pan, H., Y. Chen, , and Y. Tseng. 2000. "A secure data hiding scheme for two-color images". In *IEEE Symposium on Computers and Communication*. p. 750-755.

Pan, J., H. Huang, and L.C. Jain, 2004. *Intelligent Watermarking Techniques*, chapter Genetic Watermarking on Spatial Domain. World Scientific Co.

Parno, M. D., T. Hemker, and K. R. Fowler. 2011. "Applicability of surrogates to improve efficiency of particle swarm optimization for simulation-based problems". *Engineering Optimization (in print)*.

Parsopoulos, K. E. and M. N. Vrahatis. 2002. "Recent approaches to global optimization problems through Particle Swarm Optimization". *Natural Computing: an international journal*, vol. 1, n. 2-3, p. 235–306.

Pelikan, M., D. E. Goldberg, and F. G. Lobo. 2002. "A Survey of Optimization by Building and Using Probabilistic Models". *Computational Optimization and Applications*, vol. 21, n. 1, p. 5–20.

Pérez-Cruz, F. 2008. "Kullback-Leibler Divergence Estimation of Continuous Distributions". In *IEEE International Symposium on Information Theory (ISIT)*. (Toronto, Canada 2008).

Petitcolas, F., R.J. Anderson, and M.G. Kuhn. 1998. "Attacks on Copyright Marking Systems". In *Proceedings of the Second International Workshop on Information Hiding*. (London, UK 1998), p. 218–238. Springer-Verlag.

Petitcolas, F., R. J. Anderson, and M. G. Kuhn. 1999. "Information hiding – a survey". *Proceedings of the IEEE*, vol. 87, n. 7, p. 1062–1078.

Poli, R., J. Kennedy, and T. Blackwell. June 2007. "Particle swarm optimisation: an overview". *Swarm Intelligence Journal*, vol. 1, n. 1, p. 33-57.

Praveen, C. and R. Duvigneau. December 2007. *Metamodel-assisted particle swarm optimization and application to aerodynamic shape optimization*. Technical Report 6397. INRIA.

Puhan, N. B. and A. T. S. Ho. 2005. "Binary Document Image Watermarking for Secure Authentication Using Perceptual Modeling". In *2005 IEEE International Symposium on Signal Processing and Information Technology*. p. 393-398.

Queipo, N. V., R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker. 2005. "Surrogate-based analysis and optimization". *Progress in Aerospace Sciences*, vol. 41, n. 1, p. 1 – 28.

Rezazadeh, S. and M. Yazdi. 16-20 2006. "An Adaptive Watermarking Scheme Based on Morphological Binary Wavelet Decomposition". In *Signal Processing, The 8th International Conference on*.

Sal, D., M. Grana, and A. d'Anjou. July 31, August 04 2006. "A MOGA to place the Watermark in an Hyperespectral Image". In *2006 IEEE International Geoscience and Remote Sensing Symposium*. (Denver, USA 2006).

Sauvola, J. and H. Kauniskangas. 1999. "MediaTeam Document Database II, a CD-ROM collection of document images, University of Oulu, Finland".

Sfikas, G., C. Constantinopoulos, A. Likas, and N. P. Galatsanos. 2005. "An analytic distance metric for Gaussian mixture models with application in image retrieval". In *Proceedings of the 15th international conference on Artificial neural networks: formal models and their applications - Volume Part II*. (Berlin, Heidelberg 2005), p. 835–840. Springer-Verlag.

Shi, L. and K. Rasheed. 2008. "ASAGA: an adaptive surrogate-assisted genetic algorithm". In *GECCO '08: Proceedings of the 2008 conference on Genetic and evolutionary computation*. p. 1049–1056.

Shieh, C., H. Huang, F. Wang, and J. Pan. 2004. "Genetic watermarking based on transform-domain techniques". *Pattern Recognition*, vol. 37, n. 3, p. 555-565.

Shih, F. Y. and Y. Wu. 2004. "Enhancement of image watermark retrieval based on genetic algorithms". *Journal of Visual Communication and Image Representation*, vol. 16, n. 2, p. 115-133.

Stauffer, C. and W. E. L. Grimson. 2000. "Learning Patterns of Activity Using Real-Time Tracking". *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, n. 8, p. 747–757.

Sung, H. G. 2004. "Gaussian Mixture Regression and Classification". PhD thesis, Rice University.

Tahir, S. F., A. Khan, A. Majid, and A. M. Mirza. 2005. "Support Vector Machine based Intelligent Watermark Decoding for Anticipated Attack". *International Journal of applied Mathematics and Computer Sciences*, vol. 1, n. 1, p. 7-12.

Tax, D. M. J. and R. P. W. Duin. 2004. "Support Vector Data Description". *Machine Learning*, vol. 54, n. 1, p. 45–66.

Torczon, V. and M. W. Trosset. 1998. "Using Approximations To Accelerate Engineering Design Optimization". In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis and Optimization Symposium*. (Saint Louis, USA 1998).

Tseng, Y. and H. Pan. 2001. "Secure and invisible data hiding in 2-color images". *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 2, p. 887-896.

Ueda, N., R. Nakano, Z. Ghahramani, and G. E. Hinton. 2000. "SMEM Algorithm for Mixture Models". *Neural Computation*, vol. 12, n. 9, p. 2109–2128.

Usman, I. and A. Khan. 2010. "BCH coding and intelligent watermark embedding: Employing both frequency and strength selection". *Applied Soft Computing*, vol. 10, n. 1, p. 332 – 343.

Vapnik, V., 1995. *The Nature of Statistical Learning Theory*. Springer.

Vellasques, E., E. Granger, and R. Sabourin, 2010a. *Handbook of Pattern Recognition and Computer Vision,* 4^{th} *ed.*, chapter Intelligent Watermarking Systems: A Survey., p. 687 – 724. World Scientific Review. ISBN 9814273384.

Vellasques, E., R. Sabourin, and E. Granger. 2010b. "Intelligent watermarking of document images as a dynamic optimization problem.". In *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*.

Vellasques, E., R. Sabourin, and E. 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, E., R. Sabourin, and E. Granger. 2012a. "Fast intelligent watermarking of heterogeneous image streams through mixture modeling of PSO populations". *Applied Soft Computing (In Press, doi : 10.1016/j.asoc.2012.08.040)*.

Vellasques, E., R. Sabourin, and E. Granger. 2012b. "Gaussian mixture modeling for dynamic particle swarm optimization of recurrent problems". In *GECCO '12: Proceedings of the Genetic and Evolutionary Computation Conference*. p. 73–80.

Vellasques, E., R. Sabourin, and E. Granger. July 2012c. "DS-DPSO: a dual surrogate approach for the intelligent watermarking of bi-tonal document image streams". *Applied Soft Computing (submitted)*.

Voloshynovskiy, S., S. Pereira, T. Pun, J.J. Eggers, and J.K. Su. 2001. "Attacks on digital watermarks: classification, estimation based attacks, and benchmarks". *IEEE Communications Magazine*, vol. 39, n. 8, p. 118–126.

Wang, H., D. Wang, and S. Yang. 2007. "Triggered Memory-Based Swarm Optimization in Dynamic Environments". In *EvoWorkshops*. p. 637-646.

Wang, J. 2007. Genetic particle swarm optimization based on estimation of distribution. p. 287–296.

Wei, Z., H. Li, J. Dai, and S. Wang. 2006. "Image Watermarking based on Genetic Algorithm". In *IEEE International Conference on Multimedia and Expo (ICME)*. p. 1117–1120.

Wessel, P. "Critical values for the two-sample Kolmogorov-Smirnov test (2-sided)". http://www.soest.hawaii.edu/wessel/courses/gg313/Critical_KS.pdf.

Wu, J., X. S. Hua, and B. Zhang. July 2005. "Tracking concept drifting with Gaussian mixture model". In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. p. 1562–1570.

Wu, M. 2001. "Multimedia Data Hiding". PhD thesis, Princeton University.

Wu, M. and B. Liu. 2003. "Data Hiding in Image and Video: Part I - Fundamental Issues and Solutions". *IEEE Transactions on Image Processing*, vol. 12, n. 6, p. 685-695.

Wu, M. and B. Liu. 2004. "Data Hiding in Binary Image for Authentication and Annotation". *IEEE Transactions on Multimedia*, vol. 6, n. 4, p. 528-538.

Wu, M., H. Yu, and B. Liu. 2003. "Data Hiding in Image and Video: Part II - Fundamental Issues and Solutions". *IEEE Transactions on Image Processing*, vol. 12, n. 6, p. 696-705.

Wu, Y. and F. Y. Shih. 2006. "Genetic algorithm based methodology for breaking the steganalytic systems". *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 36, n. 1, p. 24–31.

Yamanishi, K., J. Takeuchi, G. Williams, and P. Milne. 2000. "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms". In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. p. 320–324. ACM.

Yan, S. and B. Minsker. June 2011. "Applying dynamic surrogate models in noisy genetic algorithms to optimize groundwater remediation designs". *Journal of Water Resources Planning and Management*, vol. 137, n. 3.

Yang, H. and A. C. Kot. Dec. 2006. "Binary Image Authentication With Tampering Localization by Embedding Cryptographic Signature and Block Identifier". *Signal Processing Letters, IEEE*, vol. 13, n. 12, p. 741-744.

Yang, S. 2005. "Population-based incremental learning with memory scheme for changing environments". In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*. (New York, NY, USA 2005), p. 711–718. ACM.

Yang, S. and X. Yao. Oct. 2008. "Population-Based Incremental Learning With Associative Memory for Dynamic Environments". *IEEE Transactions on Evolutionary Computation*, vol. 12, n. 5, p. 542–561.

Zhang, C. and Z. Qiu. August 2005. "Fragile Watermarking with quality control for binary images". In *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*. p. 4952-4956.

Zhang, Y. and M. S. Scordilis. 2008. "Effective online unsupervised adaptation of Gaussian mixture models and its application to speech classification". *Pattern Recognition Letters*, vol. 29, n. 6, p. 735–744.

Zhao, J. and E. Koch. 21-25 1995. "Embedding Robust Labels into Images for Copyright Protection". In *International Congress on Intellectual Property Rights for Specialised Information, Knowledge and New Technologies*. (Vienna, Austria 1995).

Zhou, Z., Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum. January 2007. "Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization". *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, n. 1, p. 66 –76.

Zielinski, K. and R. Laur. 2007. "Stopping Criteria for a Constrained Single-Objective Particle Swarm Optimization Algorithm". *Informatica*, vol. 31, n. 1, p. 51-59.