Machine Vision and Applications
DOI 10.1007/s00138-013-0493-1

ORIGINAL PAPER

Rapid blockwise multi-resolution clustering of facial images for intelligent watermarking

Bassem S. Rabil · Robert Sabourin · Eric Granger

Received: 21 August 2012 / Revised: 4 February 2013 / Accepted: 13 February 2013

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

Keywords Multi-objective optimization · Evolutionary computation · Clustering · Population-based incremental learning · Graphics processing units · Intelligent watermarking

B. S. Rabil (✉) · R. Sabourin · E. Granger
Laboratoire d'Imagerie, de Vision, et d'Intelligence Artificielle,
École de Technologie Supérieure, Université du Québec,
1100 Notre-Dame Ouest, Montreal, QC H3C 1K3, Canada
e-mail: bguendy@livia.etsmtl.ca

R. Sabourin
e-mail: Robert.Sabourin@etsmtl.ca

E. Granger
e-mail: Eric.Granger@letsmtl.ca

1 Introduction

Securing grayscale images is critical especially nowadays with the growing number of images transmitted and exchanged via the Internet. Grayscale images are widely used as medical images, biometric templates, or financial documents. These application domains rely on large volumes of grayscale images that need to be secured. Digital watermarking has been used to secure grayscale images by embedding watermarks into these images to ensure their authenticity. Finding optimal embedding parameters is a complex problem with conflicting objectives of image quality and watermark robustness. Image quality is associated with the distortion resulting from watermark embedding while watermark robustness relates to the resistance of the embedded watermark against manipulations on the watermarked image.

In intelligent watermarking (IW), different computational intelligence techniques have been proposed to find optimal embedding parameters. Authors have proposed using evolutionary computation optimization techniques like Genetic Algorithms (GA) [15], Particle Swarm Optimization (PSO) [21], and combinations of GA and PSO [8] to find embedding parameters that maximize the fitness for both quality and robustness [17]. Most of these traditional methods are

Published online: 30 March 2013

based on representing all cover image 8×8 pixels blocks in candidate solutions according to their positional order and iteratively improve the fitness until convergence is reached [15]. These methods use single aggregated objective [15,19]. To date, few authors have proposed mutli-objective formulation [4,12], where the two objectives are optimized simultaneously and multiple non-dominated solutions are located forming a Pareto front. This last approach provides more operational flexibility.

Most of the EC techniques suffer from convergence problems due to the complexity of the search space associated with high-resolution images. This traditional representation [15] assumes at least 1 bit per block and equal number of bits embedded in all image blocks using even embedding scheme. Handling constraints for large-sized candidate solutions is computationally complex. Such constraints avoid using the same frequency coefficient for embedding more than once for the same image block. In this research, grayscale face image templates are considered to be secured. For this type of images, the traditional representation [15] implies that shifting slightly the face pixels inside the image is considered as a new optimization problem, and consequently costly re-optimizations are required. And thus watermarking a stream of high resolution grayscale face images results in a stream of computationally complex optimization problems.

In this paper, a blockwise multi-resolution clustering (BMRC) framework is proposed for rapid intelligent watermarking. The proposed technique is capable of finding optimal embedding blocks and specify their embedding parameters in a computationally efficient manner. It also can handle watermarks of different lengths using proposed robustness score (RS) metric for block clusters. This proposed metric is used also to identify the optimal embedding clusters of blocks. BMRC is based on a multi-objective formulation which satisfies the trade-off between watermark quality and robustness and thus allows adaptability for different application domains, where the objectives priority vary, without the need for costly re-optimizations.

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

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

Proof of concept simulations are performed using the PUT database [7] of high-resolution face images and compared against reference method in terms of complexity and quality of solutions. Simulation results demonstrate that BMRC results in a significant reduction of the computational cost for IW by replacing costly optimization operations with associative memory recalls. The resulting solutions have nearly the same quality and robustness as those obtained with full optimization of each face image. The performance of BMRC is evaluated for different watermark length, and the robustness objective is considered of higher priority to reach up to 99.9 % of bits restored after manipulating the watermarked face image. A sensitivity analysis is performed on BMRC tunable parameters to evaluate the impact of these parameters on both the framework performance and the associative memory size. Parallel implementation using graphics processing units (GPU) is employed for the most complex functionalities of BMRC to evaluate its impact on the overall performance.

This paper is organized as follows: Sect. 2 introduces watermarking concepts, metrics, and terminology needed to understand the paper; then an overview for different IW approaches proposed in literature, and it also introduces population-based incremental learning (PBIL), and texture feature extraction for grayscale images. Section 3 describes the proposed framework for rapid blockwise IW for high-resolution grayscale facial images. The proposed experimental methodology is described in Sect. 4. Results and analysis are presented in Sect. 5.

2 Intelligent watermarking of grayscale images

Digital watermarking is deployed in many domains to assure integrity and authenticity of the original signal via fragile and robust watermarking, respectively [17]. A fragile watermark

is a type of watermark to ensure integrity, but it is broken if the watermarked image is manipulated or altered, while the robust watermark ensures authenticity and can be extracted after manipulating the watermarked image. Semi-fragile watermark considered in this paper is satisfying a trade-off between both the distortion introduced by the watermark and the watermark resistance to manipulations.

Most digital watermarking techniques proposed for grayscale images use different transform domains to embed a watermark that minimizes the visual impact and to deal with the uncorrelated coefficients in the transform domain. The most commonly used transform domains in watermarking literature are discrete cosine transform (DCT) [15] and discrete wavelet transform (DWT) [8]. Using DCT transform inheriting robustness against JPEG compression which is based on DCT transform as well, the host image is divided into small blocks of pixels (8×8 pixels), transformed to frequency domain, and watermark bits are distributed among these blocks by changing frequency bands coefficients of these blocks according to the value of the watermark bit to be embedded. Few authors have considered other transforms based on DFT [9] to improve robustness against geometric attacks since these transforms are more resistant to geometric manipulations.

2.1 Watermarking metrics

Digital watermarking system can be characterized using three main aspects: watermark quality, watermark robustness, and watermark capacity. Watermark quality measures the distortion resulting from watermark embedding; there are limits defined in literature [20] where the human vision cannot recognize the distortion resulting from the embedding. Watermark robustness measures the resistance to different manipulations and processing on the watermarked image; this is measured by the correlation between the extracted watermark after the manipulations and the original watermark. Watermark capacity measures the number of embedded bits per block given thresholds for watermark quality and/or watermark robustness.

Watermark quality and robustness are commonly measured using weighted peak signal-to-noise ratio (wPSNR) and normalized correlation (NC), respectively. Peak signal-to-noise ratio (PSNR) is calculated between original image $X_c(w,h)$ and watermarked image $X_{cw}(w,h)$ of resolution $M_c \times N_c$ using the mean squared error (MSE), where w, and h represents the index of pixels for width and height, respectively:

$$\begin{aligned} \mathrm{MSE_c} &= \frac{1}{M_c.N_c}\sum_{w=1}^{M_c}\sum_{h=1}^{N_c}\left(X_c\left(w,h\right)-X_{cw}\left(w,h\right)\right)^2 \\ \mathrm{PSNR_c} &= 10\log_{10}\cdot\left(\frac{255^2}{\mathrm{MSE_c}}\right)[\mathrm{dB}] \end{aligned} \tag{1}$$

Weighted PSNR uses an additional parameter called noise visibility function (NVF) which is a texture masking function defined by Voloshynovskiy et al. [20]. NVF arbitrarily uses a Gaussian model to estimate how much texture exists in any area of an image. For flat and smooth areas, NVF is equal to 1, and thus wPSNR has the same value of PSNR. For any other textured areas, wPSNR is slightly higher than PSNR to reflect the fact that human eye will have less sensitivity to modifications in textured areas than smooth areas. Weighted PSNR shown in Eq. 2 is proposed in the latest benchmarking for watermarking systems introduced by Pereira et al. [11].

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

The normalized correlation (NC) is calculated between embedded watermark $W(w,h)$ of resolution $M_W \times N_W$ where w and h represent the index of pixels for width and height, respectively, and the extracted watermark from the attacked image $W'(w,h)$ using Eq. 3.

$$\mathrm{NC} = \frac{\sum_{w=1}^{M_W}\sum_{h=1}^{N_W}\left[W\left(w,h\right)\ W'\left(w,h\right)\right]}{\sum_{w=1}^{M_W}\sum_{h=1}^{N_W}\left[W\left(w,h\right)\right]^2} \tag{3}$$

2.2 Watermark embedding and extraction

The watermark embedding/extracting algorithm considered in this paper is an algorithm proposed by Shieh et al. [15], where the original cover image is not required during extraction of the watermark; this reduces the required space needed to store the original cover images. Using this algorithm, the cover image X_c to be watermarked of size $M_c \times N_c$ is divided into 8×8 blocks and transformed into DCT domain where the resultant matrix $Y_{(m_c,n_c)}(a)$ for each image block at row m_c and column n_c of cover image blocks has the upper left corner as DC co-efficient and the rest of matrix are the AC coefficients, where the DCT coefficients index a ranging from 0 to 63 for 8×8 blocks are placed in zigzag order. The DCT-transformed image $Y_{(m_c,n_c)}(a)$ is then used to get the ratio between DC and AC coefficients $R(a)$ for all AC coefficients a using

$$R\left(a\right) = \sum_{m_c=1}^{M_c/8}\sum_{n_c=1}^{N_c/8}\left(\frac{Y_{m_c,n_c}\left(0\right)}{Y_{m_c,n_c}\left(a\right)}\right),\quad a\in[1,2,\ldots,63] \tag{4}$$

Then polarities P are calculated using the Eq. 5.

$$P_{(m_c,n_c)}(a) = \begin{cases} 1\ \text{if}(Y_{(m_c,n_c)}(a).R(a)) \geq Y_{(m_c,n_c)}(0) \\ \quad a\in\{\mathrm{eb}_i\},\ i = m_c\times(M_c/8)+n_c \\ 0\ \text{otherwise;} \end{cases} \tag{5}$$

Next, the watermarked DCT coefficient Y' is obtained using the Eq. 6. The index of DCT coefficients modified

belonging to $\{\text{eb}_i\}$ referred to as embedding bands for block b_i with i equal to $m_\text{c} \times (M_\text{c}/8) + n_\text{c}$. The embedding capacity for block b_i is defined as C_i in bits per block, and the watermark bits allocated for block at m_c row and n_c column $W_{(m_\text{c},n_\text{c})}(e)$, where e represents the index of set of embedding bands and finally the watermarked image X_cw is obtained using the inverse DCT for Y'.

$$Y'_{(m_\text{c},n_\text{c})}(a) = \begin{cases} Y_{(m_\text{c},n_\text{c})}(a) & \text{if } P_{(m_\text{c},n_\text{c})}(a) = W_{(m_\text{c},n_\text{c})}(e) \\ & a \in \{\text{eb}_i\},\ i = m_\text{c} \times (M_\text{c}/8) + n_\text{c} \\ (Y_{(m_\text{c},n_\text{c})}(0)/R(a)) + 1 & \text{if } P_{(m_\text{c},n_\text{c})}(a) = 0 \\ & W_{(m_\text{c},n_\text{c})}(e) = 1 \\ & a \in \{\text{eb}_i\},\ i = m_\text{c} \times (M_\text{c}/8) + n_\text{c} \\ (Y_{(m_\text{c},n_\text{c})}(0)/R(a)) - 1 & \text{otherwise} \end{cases} \tag{6}$$

2.3 Intelligent watermarking

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

EC methods like GA and PSO have attracted authors' attention due to simplicity of these techniques and the ease in adapting them to many different types of watermarking systems. Moreover, EC does not assume a distribution of the parameters space represented by selected frequency bands for embedding [15].

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

In these traditional methods, all cover image blocks are represented in optimization candidate solutions, and the selected embedding bands are altered along optimization iteratively to maximize both the watermark quality fitness (QF) and robustness fitness (RF) simultaneously. All cover image blocks have to be represented in the optimization candidate solutions as shown in Fig. 1 to allow distribution of watermark bits among cover image blocks. The optimization problem can be formalized as

$$\begin{aligned} \max_{\text{EB}_{X_\text{c}}} \quad & \{\text{QF}(\text{EB}_{X_\text{c}}), \text{RF}(\text{EB}_{X_\text{c}})\} \\ \text{EB}_{X_\text{c}} &= \{\text{eb}_1, \text{eb}_2, \ldots, \text{eb}_i, \ldots, \text{eb}_\text{NB}\}, \text{where NB} \\ &= (M_\text{c}/8) \times (N_\text{c}/8) \\ \text{eb}_i &= \{a_1, a_2, \ldots, a_e, \ldots, a_{C_i}\}, \text{where } a_e \text{ is 6-bit} \\ & \text{binary representation for embedding bands} \\ & \text{index for block } b_i \text{ with } a_e \in [0, 1, \ldots, 63] \\ \text{s.t. } a_e &\neq 0, \text{where } 1 < e < C_i, \quad \text{and} \quad 1 < i < \text{NB} \\ a_{e1} &\neq a_{e2}, \text{where } 1 < e1, e2 < C_i \end{aligned} \tag{7}$$

where b_i represents the 8×8 block in cover image of resolution $M_\text{c} \times N_\text{c}$, the total number of blocks equal to NB, a_e represents the eth embedding band for block b_i , and the embedding capacity for block b_i is C_i. The first constraint considered ensures avoiding DC coefficient a_e for embedding, and the second constraint considered ensures avoiding using the same embedding bands for the same image block.

The traditional optimization formulation [15] for IW problem implies that embedding capacity are equal for all blocks of the cover image and at least 1 bit per block is embedded. In watermarking literature, this is referred to as even embedding scheme where the embedding capacities are equal for all cover image blocks. From watermarking perspective [22], uneven embedding scheme is more suitable for better watermarking fitness where higher textured blocks are utilized for more bits to embed and smooth textured blocks are avoided for embedding.

Many authors have proposed aggregating both quality and robustness fitness into one objective for simplicity utilizing different aggregation weights for the objectives to resolve the issue of different scaling of these different types of objectives and to favor one objective over the others using these weights. Shieh et al. [15] have used Genetic Algorithm for optimizing the aggregated fitness for both quality and robustness, while Wang et al. [21] have used Particle Swarm Optimization for optimization. Other authors [8] have proposed combining both GA and PSO for optimizing the aggregated fitness for quality and robustness.

Different formulations for watermark embedding optimization have been evaluated and compared in literature [12]. Multi-objective formulation corresponds to the trade-off among different quality and robustness objectives. It provides multiple optimal non-dominated solutions (Pareto front) which gives a system operator the ability to choose among multiple solutions to tune the watermarking system [12] resolving the challenge of operating flexibility pointed out in [5].

The data flow of the multi-objective formulation is shown in Fig. 2, where this formulation deals efficiently with conflicting objectives like watermark quality and robustness against different attacks. In this formulation quality fitness $\text{QF}(\text{EB}_{X_\text{c}})$ and robustness fitness $\text{RF}(\text{EB}_{X_\text{c}})$ are optimized

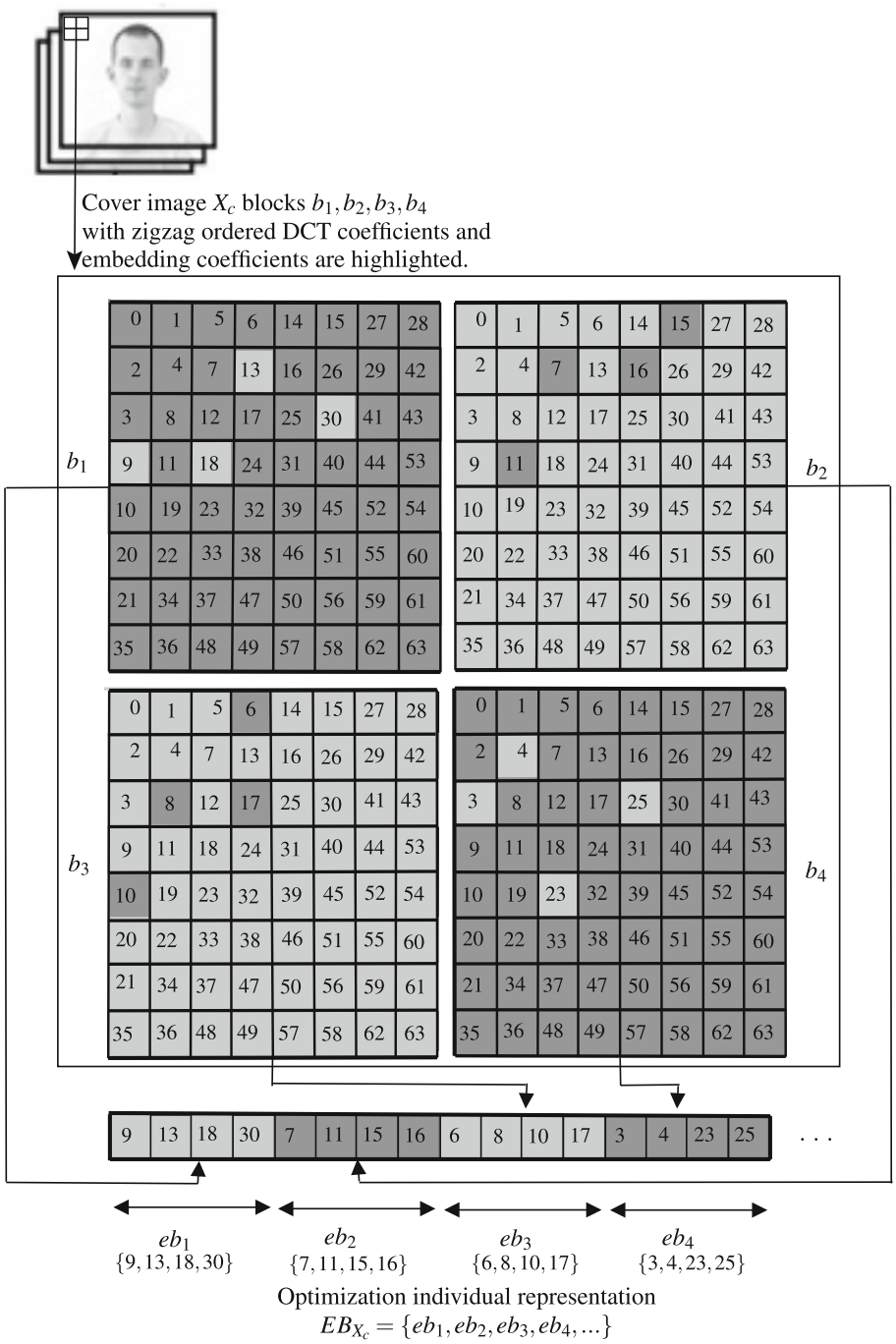

Fig. 1 Optimization candidate solutions' representation in traditional methods [15] with embedding capacity equal to four bits per block

simultaneously without favoring any objective over the other using aggregation weights. The fitness for both objectives are improved iteratively till stop criterion is reached, and optimal embedding bands eb_i for all blocks b_i are concluded from a single non-dominated solution belonging to the resulting Pareto front. For single objective formulation, the fitness of both quality $\text{QF}(\text{EB}_{X_c})$ and robustness $\text{RF}(\text{EB}_{X_c})$ are aggregated into one objective to be optimized. The objectives can be aggregated using weighted sum [15,18] or Chebyshev aggregation [19].

2.4 Population-based incremental learning

Population-based incremental learning (PBIL) method was proposed by Baluja in 1994 [1]. Its salient feature is the introduction of a real-valued probability vector. The value of each element of the vector is the probability of having a 1 in that particular bit position of the encoded chromosome. PBIL has proved efficiency with IW problem where utilizing the previous experience in subsequent generations ensures better convergence properties [12] compared with GA and PSO.

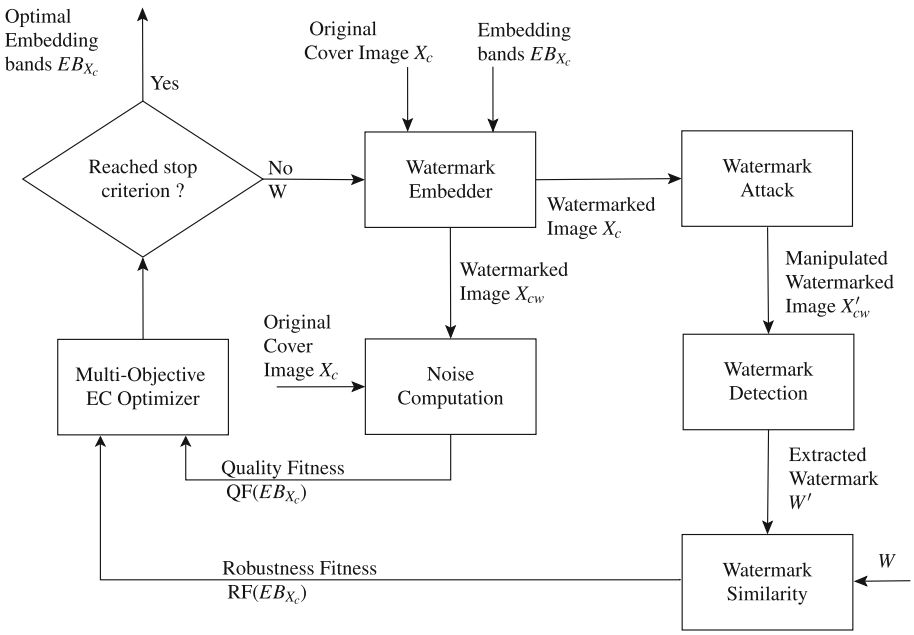

Fig. 2 Data flow diagram depicting the multi-objective formulation for watermark embedding optimization to find optimal embedding bands EB_{X_c} for cover image X_c

Also the probability vector is considered a good representation for optimization landscape that can be recalled to reproduce the landscape without the need to go through complex iterations. Bureerat and Sriworamas [2] proposed changes to PBIL algorithm to handle multi-objective optimization problems. In this algorithm the probability vector is replaced with probability matrix, where each row in this matrix represents the probability vector to create sub-population individuals.

2.5 Texture features of grayscale images

Texture features are extracted from the grayscale images using 8×8 pixels blocks granularity. The most commonly used texture features can be classified into spatial features like gray-level covariance matrix (GLCM), and other domains features like discrete cosine transform (DCT), and gabor wavelet transform. Taking the computational complexity into consideration, using spatial features would have lower complexity compared with other domains' features like DCT domain. However, the watermark embedding and extraction methods based on spatial domain would have lower robustness against different image alterations and lower watermark embedding capacity. In literature, many authors have considered DCT for extracting texture features of grayscale images. Yu et al. [24] have proposed zoning method using the most significant 39 coefficients. Sorwar and Abraham [16] have proposed selecting two different texture features based on lower coefficients, and directional edges coefficients, where the coefficients of the most upper and left regions represent vertical and horizontal edge information, respectively.

Therefore, the traditional methods described in this section lack the support for uneven embedding to utilize more textured areas for more bits to be embedded. Also smaller watermarks need padding to satisfy the constraint of having at least 1 bit per block capacity for traditional formulations [15]. Moreover, the computational complexity is not affordable using modest computational resources, and thus a novel formulation is essential for IW a stream of high-resolution images. Most of authors in intelligent watermarking were focusing on single image watermarking and did not pay enough attention to high volume of grayscale images watermarking. Only for high-volume bi-tonal images, Vellasques et al. [18] proposed a high-throughout watermarking by considering optimization of a stream of images as single dynamic optimization problem. This approach is not efficient with grayscale face images [14] due to positional representation of blocks.

3 Rapid blockwise multi-resolution clustering

BMRC shown in Fig. 3 finds optimal embedding bands in textured blocks for a stream of high-resolution face images using modest computational complexity. This is accomplished by replacing computationally expensive full optimization with memory recalls from an associative memory representing prior optimization knowledge. Face images blocks are clustered according to their texture, and then optimal embedding bands for all of blocks of same texture are selected together using prior knowledge stored in associative block cluster memory (BCM).

Solutions recalled from BCM, representing different number of clusters, are proposed to be ranked using watermark fitness to find number of blocks clusters for each face image. The number of clusters is referred to as clustering resolution

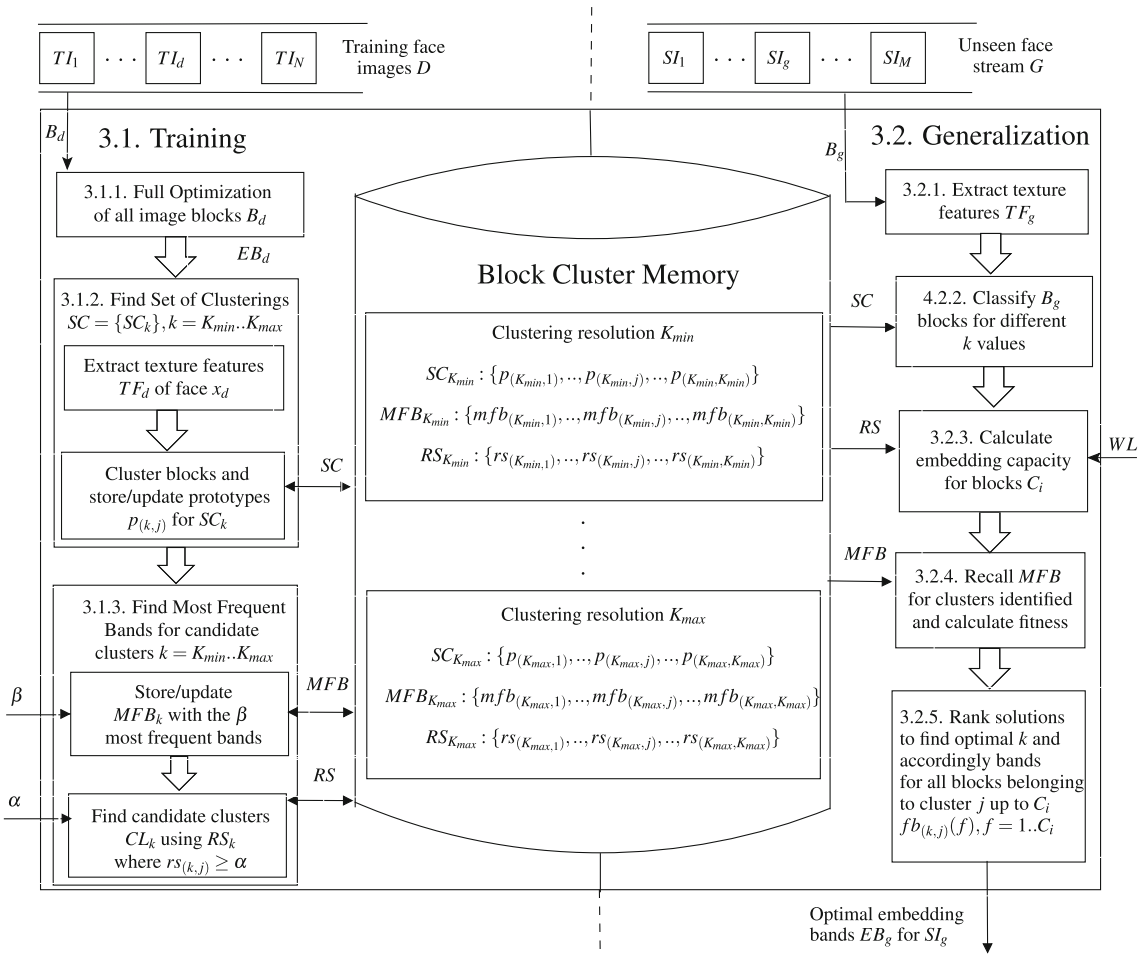

Fig. 3 General overview of BMRC architecture and processing steps for training and generalization phases. BCM is organized based on clustering resolution k ranging from $K_{\min}$ to $K_{\max}$

in this paper. This implements multi-hypothesis approach where all alternative solutions are stored and the hard decision to choose among these solutions is postponed.

The training set D consists of training face images defined as $D = \{\mathrm{TI}_1, \mathrm{TI}_2, \ldots, \mathrm{TI}_d, \ldots, \mathrm{TI}_N\}$, where TI_d represents face image of index d from the training set D of resolution $M_\mathrm{c} \times N_\mathrm{c}$. The number of face images in the training set is equal to N. For each training face image TI_d, the image is divided into 8×8 pixels blocks $B_d = \{b_i\}$, where $i = m_\mathrm{c} \times (M_\mathrm{c}/8) + n_\mathrm{c}$ with m_c, and n_c defines the row and column index of 8×8 blocks, respectively. The total number of blocks $\mathrm{NB} = (M_\mathrm{c}/8) \times (N_\mathrm{c}/8)$, and thus $i = [1, 2, \ldots, \mathrm{NB}]$.

Training face image blocks B_d are transformed into DCT domain $\mathrm{DCT}_d = \{\mathrm{dct}_i\}$, where $\mathrm{dct}_i = \{\mathrm{ac}_0, \mathrm{ac}_1, \ldots, \mathrm{ac}_a, \ldots, \mathrm{ac}_{63}\}$ with ac_0 defines the DC coefficient of the 8×8 block b_i, and ac_a defines the $\underline{a}$th DCT coefficient of the same block b_i. The texture features TF_d are extracted from DCT_d, where TF_d defines the most significant DCT coefficients from DCT_d for training face image TI_d. The texture feature vectors are defined as $\mathrm{TF}_d = \{\mathrm{tf}_i\}$, where tf_i defines the texture feature vector of block b_i. This feature vector is defined as $\mathrm{tf}_i = \{\mathrm{ac}_a\}$, where $a \in [0, 1, \ldots, 63]$, and $a = \{a_1, a_2, \ldots, a_t, \ldots, a_T\}$. The number of coefficients used to extract features is equal to T, and t is the index of texture feature in the feature vector tf_i.

After the full optimization process for face image TI_d, the optimal embedding bands EB_d are concluded for face image TI_d, where $EB_d = \{\mathrm{eb}_i\}$ with eb_i representing the optimal embedding bands for block b_i. The embedding bands define the index of DCT coefficients which are modified during embedding the watermark. It can be defined as $\mathrm{eb}_i = \{a_1, a_2, \ldots, a_e, \ldots, a_{C_i}\}$, where e is the index of the embedding bands in eb_i, and C_i is the number of embedding bands for block b_i representing the embedding capacity for b_i.

The generalization set of unseen face image G is defined as $G = \{\mathrm{SI}_1, \mathrm{SI}_2, \ldots, \mathrm{SI}_g, \ldots, \mathrm{SI}_M\}$, where the size of generalization set equals M. The subscript g is used instead of d for the data structures used in generalization phase. Thus

B_g, DCT_g, and TF_g defines the 8×8 blocks, the DCT transformed blocks, and the texture features of the face image SI_g, respectively.

Algorithm 1 describes the main steps of the proposed framework, where the training phase (lines 10–18) is performed on training set face image TI_d, and the generalization phase (lines 8–14) is triggered using generalization face image SI_g. BCM associative memory is populated (lines 1–7) with prior knowledge during the two-step training phase, where the first step finds the multi-resolution clusterings for face image blocks (line 5), and the second step calculates the most frequent embedding bands associated with each and every cluster prototype defined in the first step (line 6). Prior knowledge is recalled for different resolutions k (lines 10–13) and fitness is calculated for these resolutions (line 12), and finally solutions are ranked (line 14) to take the hard decision at the end of the process.

Algorithm 1 Main steps of the BMRC framework.

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

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

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

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

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

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

3.1 Training

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

This knowledge is stored for first training face image and gets updated with following training face images.

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

Algorithm 2 Proposed BMRC training phase.

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

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

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

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

3.1.1 Full optimization

Each face image TI_d from the training set D is fed into multi-objective optimizer [2]. Multi-objective optimization is performed to find optimal embedding bands EB_d for all individual face image blocks satisfying the trade-off between the conflicting objectives of watermark quality and robustness. All individual face image blocks b_i are encoded positionally in the optimization candidate solutions such that all blocks of face image TI_d have one bit per block to be embedded. This full optimization process is computationally complex because of large dimension of the optimization search space due to representing all face image blocks in the optimization problem.

Multi-objective optimization results in multiple optimal solutions called non-dominated solutions, where improving one objective fitness results in suffering for other objective considered. Choosing the optimal solution among these solutions is based on the priority of objectives in the application domain. This feature ensures the adaptability of the proposed system in different application domains without computationally expensive re-optimizations.

For example, quality is the most important issue with medical imaging applications where the watermarked image still goes through feature extraction process; on the other hand, robustness is the most important issue with biometrics application where the embedded watermark represents biometric traits which are used for recognition after watermark bits extraction. In this research, we employ a fixed trade-off between robustness and quality by fixing quality requirements for optimal solution weighted PSNR at 42 dB [20] which is considered acceptable from a human vision system (HVS) standpoint as shown in Fig. 4 for training set face images TI_1, TI_2, and TI_3.

3.1.2 Find set of clusterings SC

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

Extract texture features: After the full optimization process is over, optimization results obtained for all blocks with similar texture properties will provide the prior knowledge to decrease computational burden. Texture features can be extracted from spatial domain or transform domain for

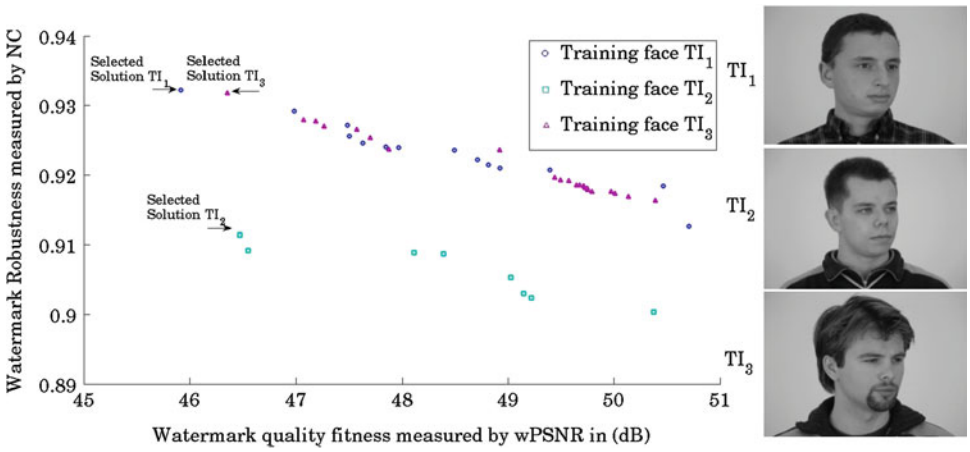

Fig. 4 Selecting solution among Pareto front for training set face images TI_d from PUT database [7] based on application domain objectives' priorities, where embedding capacity is eight bits per block

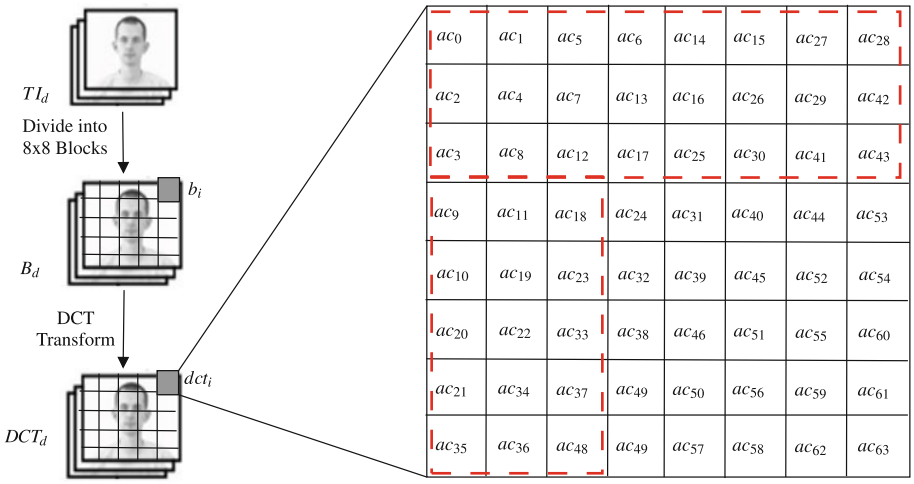

Fig. 5 Zoning method to select the most significant DCT coefficients to extract texture features tf_i for block b_i [24], $\mathrm{tf}_i = \{\mathrm{ac}_a\}$, where $a = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 33, 34, 35, 36, 37, 41, 42, 43, 48\}$

grayscale images. Although extracting texture features from spatial domain is less complex, in the proposed approach the transform domain texture features are already available for embedding and extracting the watermark. Texture feature vectors are extracted from the DCT transformed 8×8 blocks DCT_d of training face TI_d, where the most significant DCT coefficients are employed as texture feature vectors tf_i for each block b_i. Using zoning approach proposed by Yu et al. [24], texture feature vector for block b_i is defined as $\mathrm{tf}_i = \{\mathrm{ac}_a\}$ where $a = \{a_1, a_2, \ldots, a_t, \ldots, a_T\}$ with $T = 39$ as shown in Fig. 5.

Store/update multi-resolution clustering SC_k: The blocks b_i of the training face image TI_d are clustered based on their texture feature vectors tf_i. The clustering is performed using multi-resolution for different values of k. In this approach k different partitions of the texture space, ranging from $K_{\min}$ to $K_{\max}$, are computed. One of the least complexity category of clustering algorithms is k-means, which attracted lots of attention for authors in the large data clustering literature where it has time complexity of $O(nkm)$, and space complexity of $O(k)$ [6], where k is the number of clusters, n is the number of patterns to be clustered, and m is the number of iterations. K-means is widely used in content-based image retrieval (CBIR) [23] with large data. K-means results in the prototypes of clusters represented by centroids in texture feature space.

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

3.1.3 Find most frequent bands for candidate clusters

The second step of training phase involves collecting embedding band statistics for the set of clusterings SC using the full optimization results EB_d for all training face TI_d blocks. The most frequent bands $\mathrm{mfb}_{(k,j)}$ for each cluster j of blocks represented by the prototype $p_{(k,j)}$ using clustering resolution k are stored in BCM. Then robustness scores $\mathrm{rs}_{(k,j)}$ for these clusters are calculated to identify order of embedding, and candidate embedding clusters based on robustness threshold α. The optimal embedding clusters of blocks are those clusters whose $\mathrm{rs}_{(k,j)}$ are higher than the threshold α; however, some of these clusters can be ignored during embedding for watermarks of small lengths WL, and thus these are referred to as candidate embedding clusters. The order of embedding is utilizing robustness scores $\mathrm{rs}_{(k,j)}$, where the clusters of higher $\mathrm{rs}_{(k,j)}$ are used first for embedding until all watermark bits are embedded.

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

Find candidate embedding clusters: After storing/updating multi-resolution clusterings SC, robustness scores $\mathrm{rs}_{(k,j)}$ are evaluated using watermark robustness fitness of JPEG compression attack of intensity 80 % when only

embedding 1 bit per block for all blocks belonging this cluster j. Robustness scores are calculated for all clustering separately such that $\text{rs}_{(k,j)}$ is calculated for jth cluster using clustering resolution k at a time using a random watermark W_r. The length WL of this random watermark equals the number of blocks $\text{nb}_{(k,j)}$ belonging to the cluster represented by the prototype $p_{(k,j)}$ in the training image TI_d.

Algorithm 3 Identifying candidate embedding clusters.

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

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

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

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

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

Figure 6 shows an example using resolution $k = 4$, where 4 random watermarks W_r are generated of lengths $\text{WL} = \{\text{nb}_{(4,1)}, \text{nb}_{(4,2)}, \text{nb}_{(4,3)}, \text{nb}_{(4,4)}\}$ and embedded one at a time in blocks belonging to cluster j using recalled most frequent band $\text{fb}_{(4,j)}(1)$ from $\text{mfb}_{(4,j)}$. Watermark robustness against JPEG compression is measured using NC for all clusters j to calculate $\text{rs}_{(4,j)}$. The scores calculated show that the highest scores are found for edges textured blocks 0.99 and the lowest 0.65 for background smooth textured blocks. Using the robustness score threshold α ensures embedding in textured blocks and avoiding smooth background blocks. Within the textured blocks the clusters of blocks are ranked based on $\text{rs}_{(k,j)}$ to use the highest rs first for embedding and descendingly use the clusters of lower rs. Thus CL_4 would contain indices of clusters 1, 2, 4 ordered for embedding.

Figure 7 shows the order of embedding for different message length for resolutions $k = 11$ and 13, where the white pixels represent the embedding blocks belonging to candidate embedding clusters CL_k for watermarks of length WL equal 1.6, 5, 7, and 10 k. The embedding blocks are chosen based on the $\text{rs}_{(k,j)}$ of the cluster to which the block belongs to. For shorter WL only edges clusters of blocks are used for embedding; meanwhile, less textured clusters are used for longer WL until all bits are allocated.

By the end of the training phase, the associative memory BCM is populated with prior optimization knowledge. The prior knowledge should cover all clustering resolutions k, because it is infeasible to predict the number of clusters discovered in face images in the stream with different light conditions and different textured clothes, nor the number of blocks belonging to each texture cluster. The proposed associative memory holds the knowledge for different number of clusters k, and the decision of the optimal number of clusters k is postponed to generalization phase. The associative memory is organized based on value of k, where for each value of k the relevant cluster prototypes $p_{(k,j)}$ are stored along with most frequent embedding bands $\text{mfb}_{(k,j)}$ for the candidate embedding clusters whose indices are included CL_k.

3.2 Generalization

The generalization-phase processing steps are shown in Algorithm 4. Unseen set of face images $\mathbf{G} = \{\text{SI}_1, \text{SI}_2, \ldots, \text{SI}_g, \ldots, \text{SI}_M\}$ utilize the prior knowledge stored in BCM to find optimal embedding blocks and bands. Blocks b_i of SI_g image are classified based on texture feature vectors tf_i for different resolutions k using recalled clustering SC_k from BCM (line 2–4). The blocks b_i belonging to candidate embedding clusters CL_k are identified and empirical capacity C_i is calculated for these blocks using recalled $\text{rs}_{(k,j)}$ and CL_k from BCM (line 6–7); these steps (lines 6–7) are not required in even embedding scheme with equal embedding capacity in all cover image blocks.

Based on the capacity, the most frequent embedding bands $\text{mfb}_{(k,j)}$ are recalled from BCM and then used for embedding

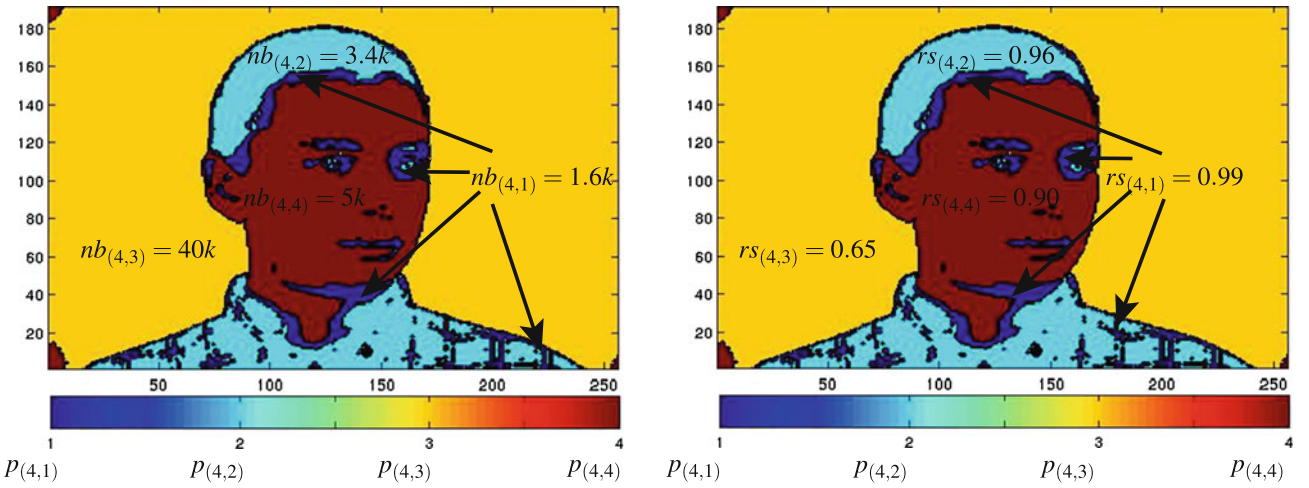

Fig. 6 Robustness scores $\mathrm{rs}_{(k,j)}$ calculation example for $k = 4$, where $\mathrm{SC}_4 = \{p_{(4,1)}, p_{(4,2)}, p_{(4,3)}, p_{(4,1)}\}$, $\mathrm{NB}_4 = \{1.6k, 3.4k, 40k, 5k\}$, $\mathrm{RS}_4 = \{0.99, 0.96, 0.65, 0.90\}$, and $\mathrm{CL}_4 = \{1, 2, 4\}$

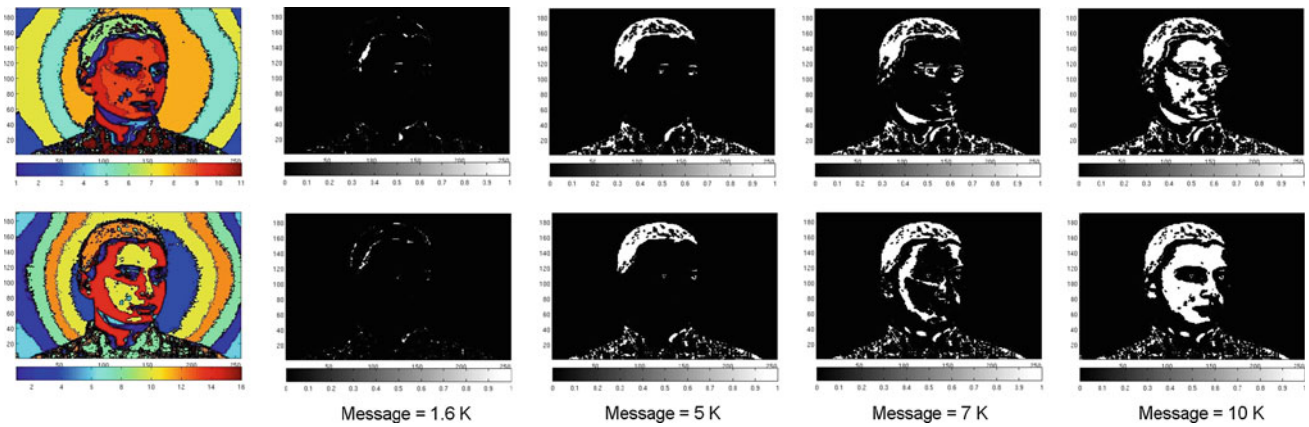

Fig. 7 Embedding blocks belonging to CL_k of clustering resolutions $k = 11, 16$ for different WL on face image from PUT database [7]

to calculate watermarking fitness for quality and robustness (line 9). If an acceptable solution is reached using resolution k, there would be no need to calculate fitness for rest of resolutions (line 10–12) to reduce the complexity of fitness evaluations. For example, if robustness of NC = 1 is reached in application domains with high priority for robustness, then the fitness evaluations corresponding to other resolutions k are not required. Finally, solutions are ranked using watermark fitness and the optimal resolution k for each face is concluded (line 14).

3.2.1 Extract texture features

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

3.2.2 Classify image blocks for different k values

After extracting texture features, face image SI_g blocks are compared against cluster centroids recalled from BCM for different resolutions k. Each block b_i is associated with the nearest centroid for each value of k in texture space. As shown in Fig. 8, the block is compared with the recalled centroids for multi-resolution clustering k in texture feature space. In this example, the face image block b_i is associated with clusters 1, 3, 2, and 6 for number of clusters k equals to 3, 4, 5, and 6, respectively, in texture feature space.

Algorithm 4 Proposed BMRC generalization phase.

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

1: **for** $g = 1 \rightarrow M$ **do**
2: Extract texture features TF_g from stream face image SI_g.
3: **for** $k = K_{min} \rightarrow K_{max}$ **do**
4: Classify SI_g blocks in texture feature space TF_g using SC_k from BCM.
5: **if** Uneven embedding is used **then**
6: Identify face image SI_g blocks belonging to CL_k.
7: Calculate embedding capacity C for SI_g blocks using Algorithm 5.
8: **end if**
9: Recall $mfb_{(k,j)}$ from BCM for clusters in face image SI_g and calculate watermark quality and robustness fitness for value of k.
10: **if** Solutions are good enough with respect to application domain **then**
11: Break and find optimal embedding bands using the current value of k.
12: **end if**
13: **end for**
14: Rank solutions representing different k values and select the suitable value for k using threshold based on human vision as shown in Sect. 3.1.1
15: **end for**

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

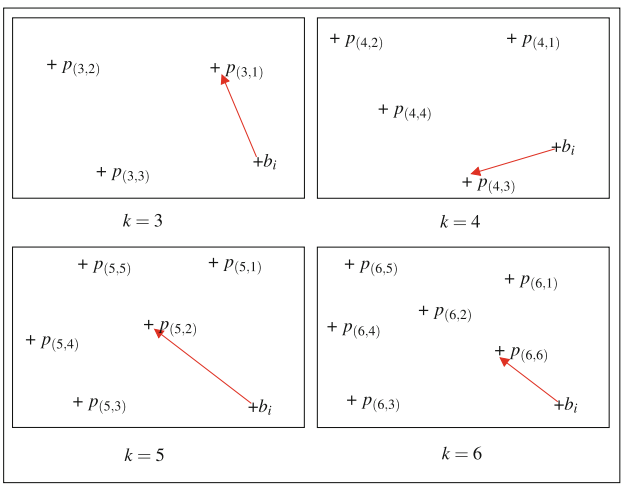

Fig. 8 Classifying block b_i for different k values in texture feature space

3.2.3 Calculate embedding capacity for different blocks C_i

For uneven embedding scheme, candidate embedding clusters based on robustness scores RS_k are recalled from BCM. These candidate clusters are ordered such that the cluster with highest $rs_{(k,j)}$ is used first and then less robustness scores clusters are used next for embedding. Classifying face image blocks into foreground and background is not efficient, as some of the foreground blocks have better robustness than others. These can be utilized for larger embedding capacities and other foreground blocks should be avoided during embedding as shown in Fig. 7 for smaller watermarks. Ordering candidate embedding clusters also ensures adaptability for different lengths watermarks.

Algorithm 5 Calculating watermark capacity for different blocks.

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

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

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

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

3.2.4 Recall MFB_k for clusters found and calculate fitness

The most frequent embedding bands $mfb_{(k,j)}$ associated with prototypes $p_{(k,j)}$ of clusters found in face image SI_g are recalled from BCM. These most frequent embedding bands

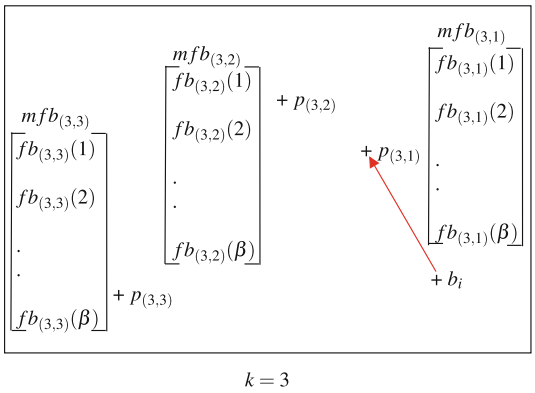

Fig. 9 Classifying block b_i in texture feature space and recall $\text{mfb}_{(k,j)}$ associated with prototypes $p_{(k,j)}$ for $k = 3$, where $\text{SC}_3 = \{p_{(3,1)}, p_{(3,2)}, p_{(3,3)}\}$, and $\text{MFB}_3 = \{\text{mfb}_{(3,1)}, \text{mfb}_{(3,2)}, \text{mfb}_{(3,3)}\}$

are used as optimal embedding bands for all face image blocks belonging to the same cluster of blocks based on their texture features. This recall is performed for different resolutions k to find optimal embedding bands for all blocks; then watermark quality and robustness fitness are calculated using these bands.

As shown in Fig. 9 for $k = 3$, each block b_i is classified using the distance to the prototypes $p_{(k,j)}$ of defined blocks clusters in texture feature space; then $\text{mfb}_{(k,j)}$ associated with $p_{(k,j)}$ are recalled. The embedding capacity C_i of b_i decides how many embedding bands $\text{fb}_{(k,j)}(f)$ are selected from $\text{mfb}_{(k,j)}$ to be used for embedding the watermark, such that if the capacity is equal to 1 bit-per-block, only the embedding band $\text{fb}_{(k,j)}(1)$ is selected for embedding; on the other hand if the capacity is equal to 3 bits-per-block, then the bands $\text{fb}_{(k,j)}(1)$, $\text{fb}_{(k,j)}(2)$, and $\text{fb}_{(k,j)}(3)$ are selected.

3.2.5 Rank solutions according to fitness to find value of k

Using the watermark fitness calculated for different values of k, solutions are proposed to be ranked to find the optimal number of cluster k to accomplish maximum watermark fitness. Ranking solutions based on two conflicting objectives like watermark quality and robustness would involve decision based on the priorities of the application domain. In this research we employ the same criteria of selecting solutions from full optimization module in training phase at Sect. 3.1.1. Thus, the number of clusters is determined using application-specific metrics rather than traditional clustering metrics like cluster accuracy (ACC) [10].

4 Experimental methodology

The database for face images used in experiments is proposed to be PUT [7] face database which consists of 100 individuals with 100 poses for each individual. Color face images of resolution 2,048 × 1,536 are converted to grayscale level. Using the first pose of each individual (face images of name pattern IIII1001.JPG where IIII is the individual number in four digits), the first 40 face images are used for verification, the next 10 individuals face images for training with full optimization to populate associative memory, and finally the last 40 face images is used as small testing set. Another larger testing set which consists of 100 for the last poses is used for extended testing for the proposed system (face images of name pattern IIII4001.JPG and IIII4002.JPG where IIII is the individual number in four digits). This large testing set consists of 198 face images using the poses 4001 and 4002.

Fig. 10 BancTec binary logo used as watermark to be embedded in face images

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

The watermark to be embedded is BancTec binary logo with different resolutions shown in Fig. 10. The watermark embedding/extracting algorithm used in experiments is an algorithm proposed by Shieh et al. [15] as illustrated in Sect. 2. The metrics used in experimentation for measuring watermark quality and robustness are wPSNR and NC, respectively, as defined in Sect. 2. Only robustness against JPEG compression of quality factor 80 % is considered in experimentation, the impact of attack intensity has been addressed before [13].

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

The first experiment compares the proposed system with the baseline system representing traditional methods with full optimization for all face image blocks. The optimization for baseline system is based on multi-objective PBIL proposed by Bureerat and Sriwaramas [2], with maximum iterations set to 40 iterations, and population size equal to 24.

The size of external archive is set to 20. The performance of the baseline system using mutli-objective PBIL is compared with traditional Multi-Objective Genetic Algorithm [3] (MOGA) using the same settings for both except for initialization. In multi-objective PBIL probability vectors are all initialized to 0.5 for all bits, while for MOGA initial bands are set to random embedding bands $\text{eb}_i \in [1, 63]$. Out of the resultant Pareto front, one solution is selected based on HVS criteria. Using this criteria, the solution of highest robustness with quality fitness equal to 42 dB measured using wPSNR metric is selected as described in Sect. 3.1.1. In this experiment, the minimum number of clusters $K_{\min}$ is set to 3 and maximum number $K_{\max}$ is set to 40. The maximum training capacity β is set to 20 bits-per-block, and the threshold for RS α is set to 89 %.

The second experiment evaluates the performance of the proposed system on validation set with uneven and even embedding as special mode of operation for the proposed system. In even embedding, embedding capacity calculation is not needed, nor ranking clusters of blocks using robustness scores. The quality of solutions produced using even and uneven embedding are compared against baseline system, and the time complexity is compared in both cases to evaluate the computational complexity reduction using the proposed system. The watermark length is 48.8 k-bits yielding to embedding capacity equal to 1 bit per block for even embedding scheme. The performance of BMRC with uneven scheme is analyzed on sample face images with different fitness levels.

In the third experiment, the performance of the proposed system for watermarks of different lengths is evaluated starting of the smallest logo whose resolution is 30×30 yielding to around 0.9 k-bits up to largest logo of resolution 221×221 yielding to 48.8 k-bits which is compared against the baseline system using Shieh method [15] using even embedding with equal embedding capacity equal to 1 bit per block. The training set for this experiment consists of one face image $N = 1$. From this experiment the optimal watermark length for using biometric traits to be embedded is concluded. Sensitivity analysis is performe for tunable parameters α and β for the proposed system. The functionalities of highest complexity are migrated into parallel implementation using GPU to evaluate its impact on the resulting complexity reduction.

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

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

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

5 Experimental results

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

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

CPU time	Training in ks			Generalization in ks		
	T_{optim}	$T_{\text{SC}} + T_{\text{MFB}}$	T_{RS}	T_{classify}	T_{capacity}	T_{fitness}
Even scheme	1,160	4	N/A	10	N/A	570
Uneven scheme	1,160	4	1	10	1	570

face image is 1,160 ks for the baseline with full optimization, compared with an average of 14.5 ks for the proposed BMRC during generalization. Also results show minimal impact of using uneven embedding scheme on the proposed system complexity.

Table 2 shows the complexity reduction measured by number of fitness evaluation, where results show significant reduction of 93.5 % when considering the training fitness evaluations with generalization for a stream of 40 face images. This complexity reduction is increased up to 95.5 % for larger streams of 198 face images. The fitness evaluations of the baseline and MOGA represent the fitness evaluations during the full optimization. The fitness evaluations for the proposed BMRC represent the fitness evaluations of the optimization during the training phase, and fitness evaluations for different number of clusters k to select the optimal value of k during the generalization phase. The fitness comparison of the development set of face images is shown in Fig. 11.

Table 3 shows mean fitness for validation set where there is improvement in robustness fitness and slight degradation in quality fitness within the acceptable quality according to HVS using highly reduced computational resources. The baseline system based on PBIL has better robustness fitness than using MOGA because of the intrinsic property of PBIL of having probability vectors. This helps to improve the convergence properties compared with traditional GA. The complexity of MOGA is equal to the complexity of multi-objective PBIL as shown in Table 2, where the fitness

Table 2 Computational complexity reduction of the proposed BMRC system compared with baseline system and MOGA [3] for a stream of 40 face images using training set size $N = 1$, measured in number of fitness evaluations for the stream

Fitness evaluation	Baseline and MOGA [3]	Proposed BMRC
Training	N/A	960
Generalization	38,400	1,520
Complexity reduction		
Generalization only	96.0 %	
Overall reduction	93.5 %	

Table 3 Mean fitness for validation set of face images for MOGA [3] and baseline system compared with proposed system with even and uneven scheme using watermark of length WL = 48.8 k

Mean fitness	Quality	Robustness
MOGA [3]	58.41 ± 3.35	0.9393 ± 0.0061
Baseline	57.93 ± 2.71	0.9589 ± 0.0240
Even scheme	56.79 ± 3.76	0.9617 ± 0.0171
Uneven scheme	55.31 ± 2.24	0.9651 ± 0.0087

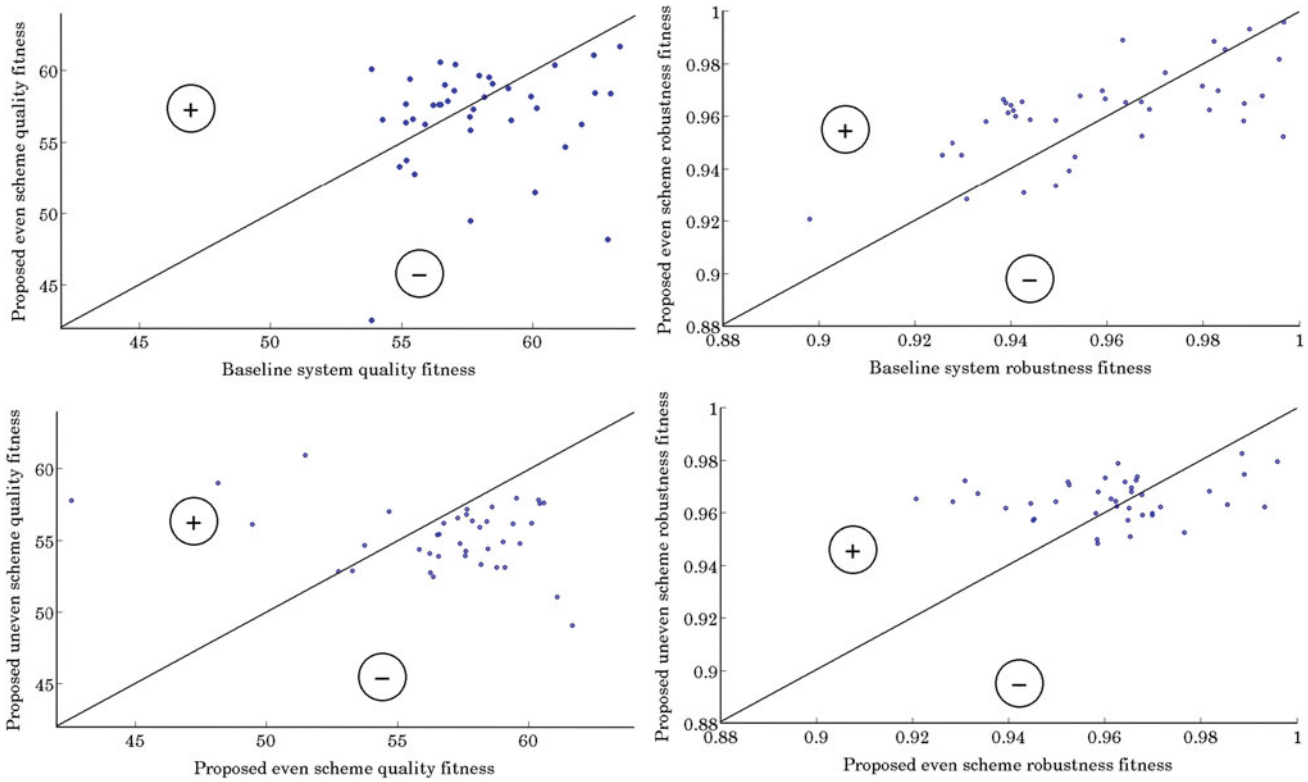

Fig. 11 Fitness comparison of the proposed system with even and uneven scheme compared with baseline system

evaluations of both methods are equal for the same population size and number of generations.

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

Figure 12 shows two face images with different fitness levels due to varying number of blocks belonging to similarly textured clusters. A histogram of the number of blocks belonging to each cluster and their relevant robustness scores RS for these two face images are shown in the figure. The robustness scores threshold is set to $\alpha = 0.89$, such that clusters of RS $< \alpha$ are excluded from embedding. Empirical embedding capacities C are calculated using Algorithm 5. The maximum empirical capacity C_{Max} of a blocks cluster is incremented until all watermark bits are distributed among the blocks belonging to the same cluster.

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

Table 4 shows the performance of the proposed system for different length of the watermark WL. The performance is represented using mean values of quality and robustness fitness for validation set of face images along with the maximum embedding capacity $C_{\text{Max}}(g)$ for face image SI_g, and optimal number of clusters k as shown in Fig. 14. Traditional approaches represented by baseline system and MOGA are shown for watermark length WL = 48.8 k only due to the formulation of these traditional methods. The extracted watermarks for different WL using the proposed system are shown in Fig. 13. The watermark fitness including quality and robustness is improved with watermarks of smaller lengths WL.

Figure 14 shows that the optimal message length for the best robustness is 1.6 k bits, where the average robustness for a stream of 40 face images is over 99.9 %. This message length ensures the highest robustness for such high-resolution grayscale face images. Also in Fig. 14, the number of clusters k is fluctuating reflecting the variance of k along the stream of validation set of face images. The time complexity for fitness evaluation T_{fitness} is 190 ks for 506 fitness evaluations compared with 570 ks for 1,520 fitness evaluations as shown on Tables 1 and 2, where the fitness evaluation is not needed once a robustness fitness for k of 1 is reached, yielding to additional 67 % complexity reduction. The performance of the proposed system using watermarks of different lengths up to 48.8 k-bits is compared against the baseline system performance for the same length.

Table 5 shows the impact of the tunable parameter β on the size of associative memory required for BCM Mem_{BCM}. The total memory Mem_{BCM} consists of memory required to store set of clusterings SC, robustness scores RS, and most frequent embedding bands MFB defined as Mem_{SC}, Mem_{RS}, and Mem_{MFB}, respectively. The memory size is dependent on the binary representation of float and integers defined as $\text{Rep}_{\text{float}}$ and Rep_{int}, respectively, where Mem_{SC} equal to $\text{Rep}_{\text{float}} \times 39 \times \sum_{k=K_{\min}}^{K_{\max}} k$, Mem_{RS} equal to $\text{Rep}_{\text{float}} \times \sum_{k=K_{\min}}^{K_{\max}} k$, and Mem_{MFB} equal to $\text{Rep}_{\text{int}} \times \beta \times \sum_{k=K_{\min}}^{K_{\max}} k$. As shown in Table 4, the maximum capacity for the stream $\text{Max}(C_{\text{Max}}(g))$ shows that β could be reduced to 10 instead of 20 for watermarks of length 6.4 k. However, changing β has minimal impact on the memory size, especially the memory size is around only 1 Mega bit which can be afforded easily with ordinary computers.

Table 6 shows the impact of the robustness scores threshold α to identify candidate embedding clusters on the proposed system with watermarks of different lengths WL. For small watermarks of length 1.6 k, the quality and robustness fitness are not affected because the mean maximum capacity $C_{\text{Max}}(g)$ is 2 bits-per-block. This implies that only the two highest robustness scores clusters are used for embedding, and thus increasing the threshold up to 94 % would not affect the fitness nor the clusters used for embedding. For medium watermarks of length 6.4k, the quality fitness is decreased slightly and robustness fitness is increased slightly when increasing α. This can be explained by the maximum embedding capacity used for the stream of the 40 face images $\text{Max}(C_{\text{Max}}(g))$ of 9 bits-per-block, compared with 7 bits-per-block for α equal to 0.89. This implies that increasing α

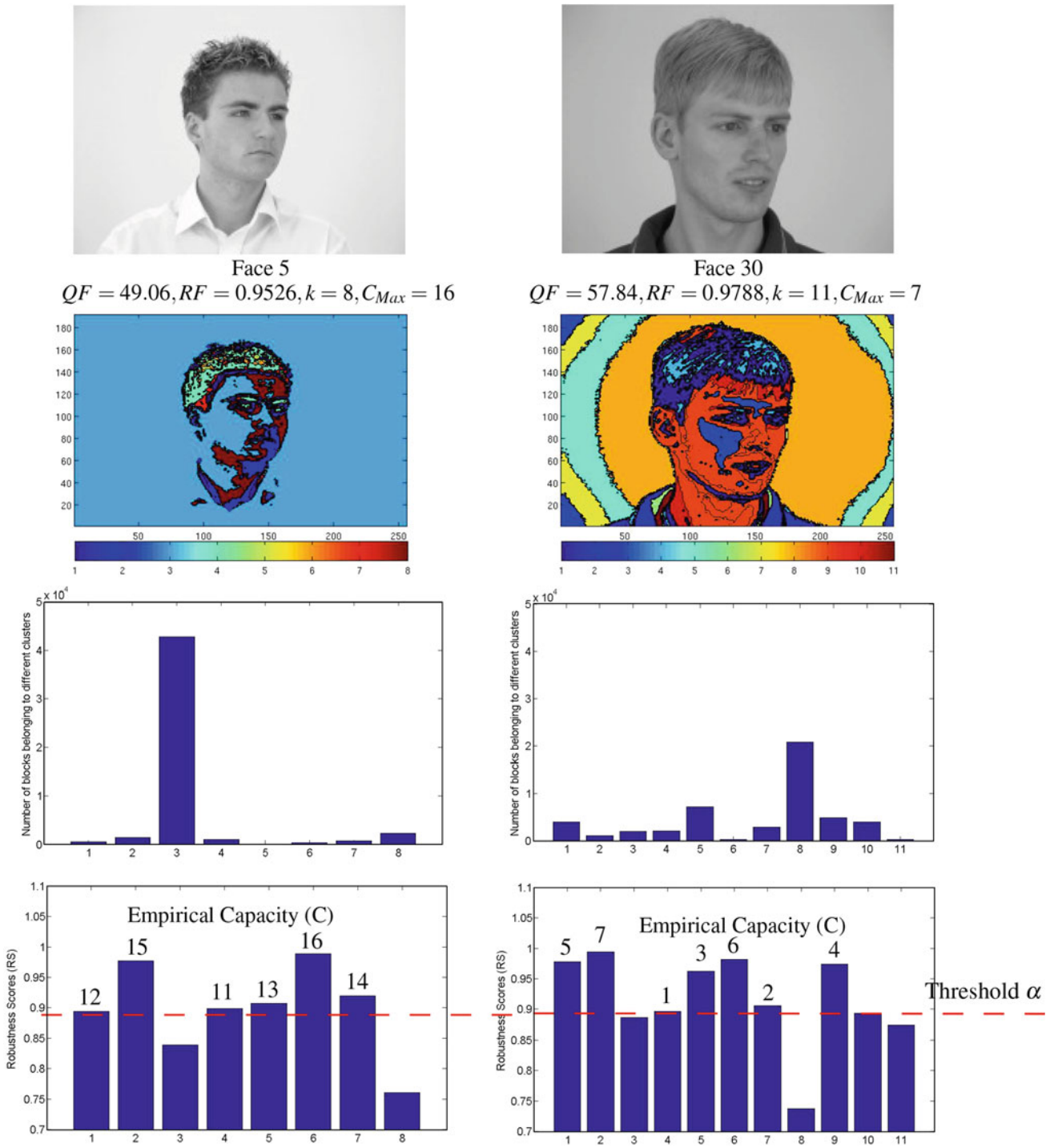

Fig. 12 The impact of variations of the number of blocks belonging to similarly textured clusters on the proposed BMRC performance using uneven scheme

for this watermark length WL would exclude some embedding clusters and increase the capacity of other clusters. For large watermarks of length 22.5 k, the impact is similar to medium watermark; however, it is more significant increase in mean robustness fitness, decrease in mean quality fitness, and increase in mean maximum capacity $C_{\text{Max}}(g)$.

Table 7 shows the complexity comparison between the fitness evaluation for both watermark quality and robustness using matlab and migrating DCT transform to GPU implementation and the watermark embedding/extraction to C. The results show minimal root mean square error 0.1496 and 0.0054 for quality and robustness fitness, respectively, due to

Table 4 Proposed system performance using uneven scheme with different watermark length

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

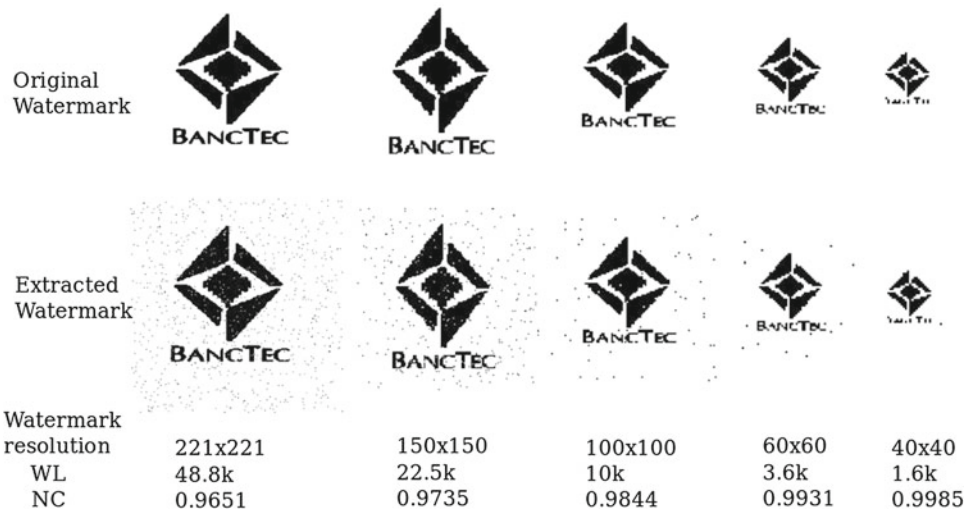

Fig. 13 Extracted watermarks of different length WL using the proposed system with uneven scheme

different representation for numeric formats between GPUs and matlab data structures. The experiment shows huge computational complexity reduction of cpu time for fitness evaluation of 40 face images stream, where GPU implementation is more than 100 times faster than CPU implementation. Time complexity is measured using the time to compute fitness of watermark quality T_{QF} and robustness T_{RF} for one face image. Watermark quality and robustness fitness evaluation for one face image involves 3 DCT transforms, and 2 inverse DCT transforms for the high-resolution face image.

However, still, the optimization iteration computation $T_{\text{Iteration}}$ in traditional methods is highly complex due to the constraints handling to avoid embedding on DC coefficient and embedding multiple bits in the same coefficient of the 8×8 block as shown in Eq. 7. And thus the proposed framework is well suited for GPU implementation more than traditional methods. This yields to a watermarking system throughput of 12 high resolution face images per hour compared to 0.0667 images per hour for traditional methods, with migrating only DCT costly transform into GPU implementation.

Table 8 shows the impact of increasing training set size from 1 training face image to 2, 3, and 7 face images using different embedding capacities 1, 2, 4, 8 bits per block for training face images. The optimal performance is accomplished using 3 face images for training with capacity 8 bits per block with mean robustness equal to 0.9993; using more training face images would result in over-training. The cpu time for single iteration $T_{\text{Iteration}}$ for the full optimization step of the training is shown in the table as well. Figure 15 shows the impact of increasing training set size N and embedding capacity on the watermarking fitness produced by the proposed system.

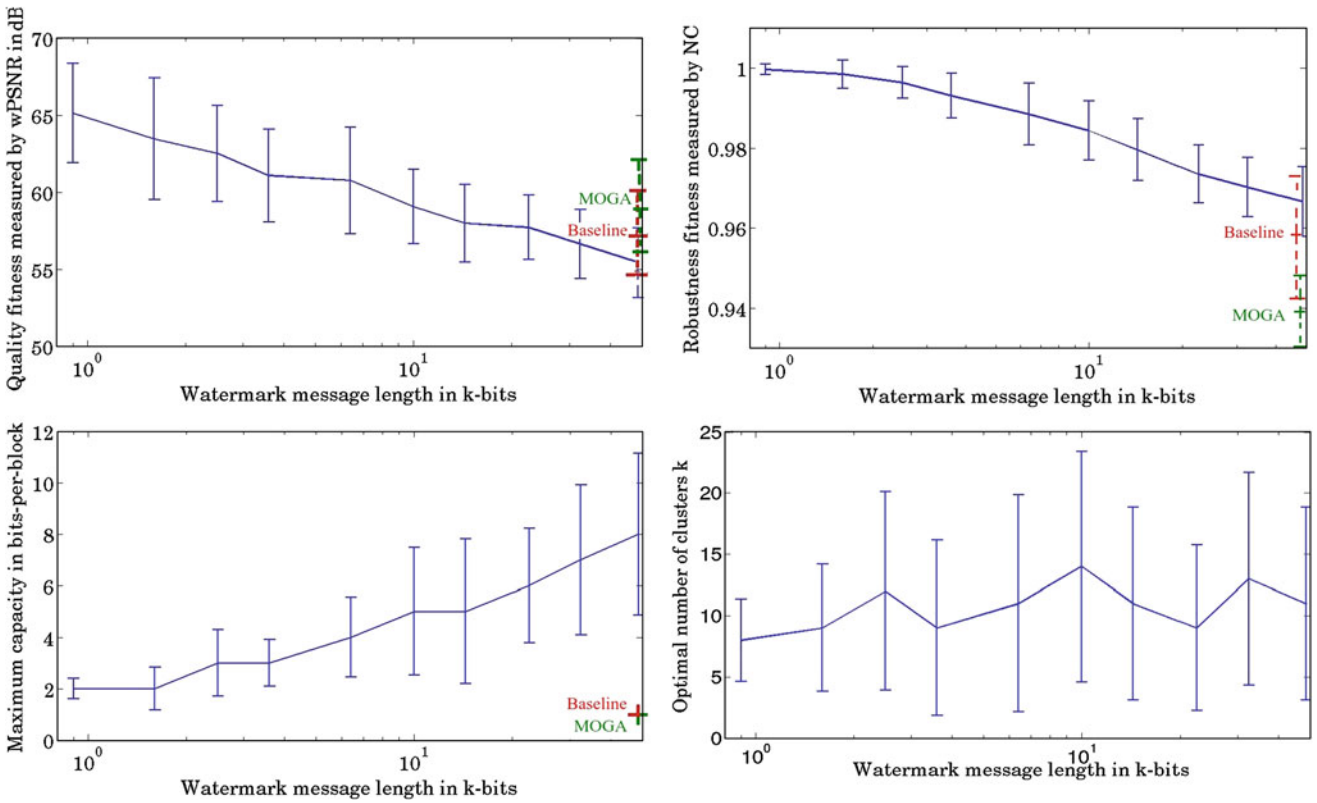

Fig. 14 Proposed system performance with uneven scheme for different watermark lengths WL

Table 5 Impact of β on the total memory size of BCM Mem_{BCM} in k-bits, where $\text{Rep}_{\text{float}} = 32$-bits as per IEEE 754 and $\text{Rep}_{\text{int}} = 6$-bits to represent index of embedding coefficients $a \in [0, 1, \ldots, 63]$

β	Mem_{SC}	Mem_{RS}	Mem_{MFB}	Mem_{BCM}
30	1,019	26	147	1,192
20	1,019	26	98	1,143
10	1,019	26	49	1,094

Table 9 shows the performance of the proposed system using training set of size $N = 3$ and watermark length WL $= 1.6$k concluded from previous experiments for two different testing sets. The stream of face images size is $M = 40$ and 198 faces, respectively, for the two testing sets. The quality of solutions are almost the same for training set face images, where the mean quality is slightly degraded from 64.08 dB to 63.96 dB for larger testing set, and the mean robustness fitness is still around the robustness fitness threshold of 99.9 % for both sets.

6 Conclusions

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

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

Table 6 Impact of robustness scores threshold α on the proposed system performance for watermarks of different lengths WL for stream of 40 face images

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

Table 7 Impact of using GPU to perform DCT transform, and C for watermark embedding/extraction on fitness evaluation complexity $T_{\text{QF}} + T_{\text{RF}}$, accuracy for 40 face images, and optimization iteration complexity $T_{\text{Iteration}}$ in traditional approaches [15]

	GPU	CPU
Mean ($T_{\text{QF}} + T_{\text{RF}}$) (s)	2.03 ± 0.84	245.52 ± 1.66
Stream ($T_{\text{QF}} + T_{\text{RF}}$) (s)	81.10	9,820.77
Quality RMSE	0.1496	
Robustness RMSE	0.0054	
$T_{\text{Iteration}} \times 10^3$ (s)	29 ± 1.63	24 ± 2.12

for embedding and enable handling watermarks of different lengths. This metric identifies the optimal embedding blocks grouped according to their texture.

Simulation results show a significant complexity reduction measured in number of fitness evaluations including the training phase of BMRC. This complexity reduction is 93.5 % for a stream of 40 face images, and it is increased up to 95.5 % for a stream of 198 face images. The quality of solution produced by the proposed framework is almost of the same accuracy of full optimization. Sensitivity analysis shows the optimal message length of 1.6 k when consider-

Table 8 Proposed system performance with different training set size N and capacity C

N	Quality	Robustness	k	$C_{\text{Max}}(g)$
Capacity = 1-bit per block, $T_{\text{Iteration}} = 1{,}361 \pm 13.98$				
1	66.56 ± 8.72	0.9734 ± 0.0139	17 ± 9.19	3 ± 0.62
2	80.29 ± 2.73	0.9673 ± 0.0176	19 ± 10.35	3 ± 0.90
3	80.03 ± 2.85	0.9663 ± 0.0192	18 ± 8.66	3 ± 0.73
7	78.05 ± 3.39	0.9641 ± 0.0172	18 ± 9.84	3 ± 1.26
Capacity = 2-bit per block, $T_{\text{Iteration}} = 3{,}524 \pm 8.10$				
1	80.14 ± 2.10	0.9675 ± 0.0164	21 ± 9.82	3 ± 1.17
2	80.67 ± 1.76	0.9687 ± 0.0176	23 ± 11.26	3 ± 1.07
3	81.08 ± 1.57	0.9645 ± 0.0211	19 ± 8.04	2 ± 0.68
7	78.73 ± 2.99	0.9637 ± 0.0169	19 ± 9.83	3 ± 1.10
Capacity = 4-bit per block, $T_{\text{Iteration}} = 6{,}513 \pm 573.76$				
1	69.32 ± 3.57	0.9982 ± 0.0031	14 ± 8.70	3 ± 0.90
2	80.37 ± 2.35	0.9677 ± 0.0202	21 ± 9.58	3 ± 0.78
3	79.80 ± 2.49	0.9668 ± 0.0215	17 ± 9.67	3 ± 0.87
7	77.92 ± 2.65	0.9685 ± 0.0166	24 ± 10.72	3 ± 1.08
Capacity = 8-bit per block, $T_{\text{Iteration}} = 28{,}978 \pm 1{,}628.56$				
1	63.46 ± 3.94	0.9985 ± 0.0035	9 ± 5.18	2 ± 0.83
2	63.48 ± 4.99	0.9988 ± 0.0026	9 ± 3.78	2 ± 0.60
3	**64.08 ± 4.81**	**0.9993 ± 0.0016**	**10 ± 7.29**	**2 ± 0.62**
7	66.70 ± 3.34	0.9978 ± 0.0038	13 ± 6.64	3 ± 0.98

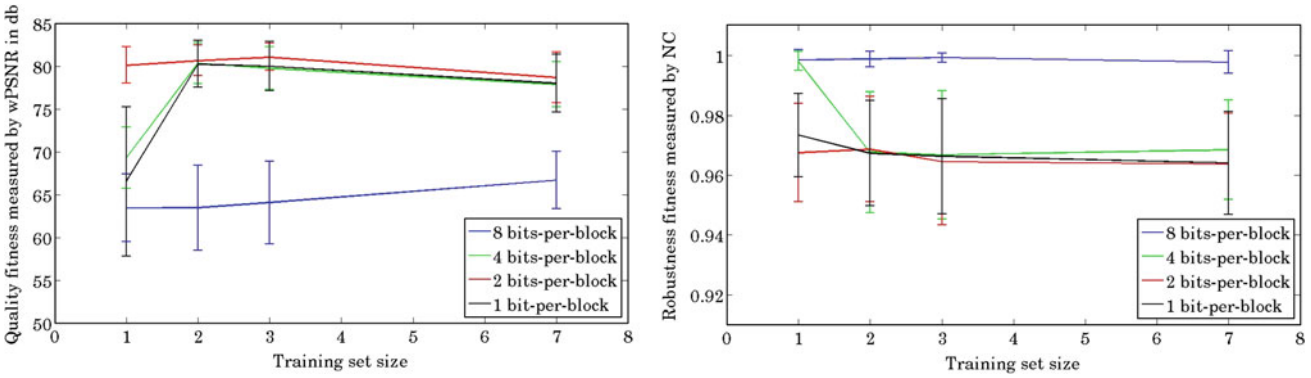

Fig. 15 Impact of increasing training set size N on the proposed system performance

Table 9 Proposed system performance with two different testing sets of size M equal to 40 and 198 face images, respectively, using training set size N equal to 3 face images

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

ing robustness fitness of 99.9 % threshold and evaluates the impact of other tunable parameters on framework performance and associative memory size.

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

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

Acknowledgments This work was supported by the Natural Sciences and Engineering Research Council of Canada and BancTec Inc.

References

1. Baluja, S.: Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning. Tech. rep, CMU (1994)
2. Bureerat, S., Sriworamas, K.: Population-based incremental learning for multiobjective optimization. Soft Comput. Ind. Appl. **39**, 223–232 (2007)
3. Deb., K.: Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, New York (2001)
4. Diaz, D.S., Romay, M.G.: Introducing a watermarking with a multiobjective genetic algorithm. In: Genetic and Evolutionary Computation Conference (GECCO 2005), pp. 2219–2220 (2005)
5. Haouzia, A., Noumeir, R.: Methods for image authentication: a survey. MultiMed Tools Appl. **39**, 1–46 (2008)
6. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31**(3), 264–323 (1999)
7. Kasinski, A., Florek, A., Schmidt, A.: The put face database. Image Process. Commun. **13**(3–4), 59–64 (2008)
8. Lee, Z.J., Lin, S.W., Su, S.F., Lin, C.Y.: A hybrid watermarking technique applied to digital images. Appl. Soft Comput. **8**, 798–808 (2008)
9. Licks, V., Jordan, R.: Geometric attacks on image watermarking systems, pp. 68–78. IEEE Multimedia, IEEE Computer Society (2005)
10. Nie, F., Zeng, Z., Tsang, I.W., Xu, D., Zhang, C.: Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering. IEEE Trans. on Neural Networks (T-NN) **22**(11), 1796–1808 (2011)
11. Pereira, S., Voloshynovskiy, S., Madueno, M., Marchand-Maillet, S., Pun, T.: Second generation benchmarking and application oriented, evaluation. pp. 340–353 (2001)
12. Rabil, B.S., Sabourin, R., Granger, E.: Intelligent watermarking with multi-objective population based incremental learning. In: IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing(IIH-MSP 2010), pp. 131–134. Darmstadt, Germany (2010)
13. Rabil, B.S., Sabourin, R., Granger, E.: Impact of watermarking attacks on biometric verification systems in intelligent bio-watermarking systems. In: IEEE Workshop on Computa-

tional Intelligence in Biometrics and Identity Management 2011. IEEE Symposium Series On Computational Intelligence 2011, pp. 13–20. Paris, France (2011)
14. Rabil, B.S., Sabourin, R., Granger, E.: Watermarking stack of grayscale face images as dynamic multi-objective optimization problem. In: International Conference on Mass Data Analysis of Images and Signals in Medicine, Biotechnology, Chemistry and Food Industry, pp. 63–77. New York/USA (2011)
15. Shieh, C.S., Huang, H.C., Wang, F.H., Pan, J.S.: Genetic watermarking based on transform-domain techniques. Pattern Recogn. **37**, 555–565 (2004)
16. Sorwar, G., Abraham, A.: Dct based texture classification using a soft computing approach. Malays. J. Comput. Sci. **17**(1), 13–23 (2004)
17. Vellasques, E., Granger, E., Sabourin, R.: Intelligent Watermarking Systems: a Survey, Handbook of Pattern Recognition and Computer Vision, 4th edn., pp. 1–40 (2010)
18. Vellasques, E., Sabourin, R., Granger, E.: A high throughput system for intelligent watermarking of bi-tonal images. Appl. Soft Comput. **11**(8), 5215–5229 (2011)
19. Vellasques, E., Sabourin, R., Granger, E.: Gaussian mixture modeling for dynamic particle swarm optimization of recurrent problems. In: Genetic and Evolutionary Computation Conference (GECCO 2012), pp. 73–80. Philadelphia, USA (2012)
20. Voloshynovsky, S., Herrigel, A., Baum, N.: A stochastic approach to content adaptive digital image watermarking. In: Proceedings of the 3rd International Workshop on Information Hiding 1999, pp. 211–236. Dresden, Germany (1999)
21. Wang, Z., Sun, X., Zhang, D.: A novel watermarking scheme based on pso algorithm. In: Proceedings of the Life System Modeling and Simulation 2007, (LSMS'07). International Conference on Bio-Inspired Computational Intelligence and Applications, pp. 307–314 (2007)
22. Wu, M.: Multimedia Data Hiding. Ph.D. thesis, Princeton University (2001)
23. Yang, Y., Xu, D., Nie, F., Yan, S., Zhuang, Y.: Image clustering using local discriminant models and global integration. IEEE Trans. Image Process. (T-IP) **10**, 2761–2773 (2010)
24. Yu, F., Oyana, D., Hou, W.C., Wainer, M.: Approximate clustering on data streams using discrete cosine transform. J. Inform. Process. Syst. **6**(1), 67–78 (2010)

Author Biographies

Bassem S. Rabil obtained B.Sc. and M.Sc. in computers and systems engineering from Ain Shams University, Cairo, Egypt at 1998 and 2007, respectively. He held different roles in software development industry in research and development organizations at Mentor Graphics and Ericsson from 2000 till 2009. He joined Laboratoire d'imagerie, de vision et d'intelligence artificielle (LIVIA) at École de Technologie Supérieure (ÉTS), Université du Québec, Montréal, Canada in 2009 to pursue his Ph.D. His research interests are in the areas of machine learning, evolutionary computation, multiobjective optimization, incremental learning, clustering, coevolution, intelligent watermarking, and biometrics.

Robert Sabourin joined the physics department of the Montreal University in 1977 where he was responsible for the design, experimentation and development of scientific instrumentation for the Mont Mégantic Astronomical Observatory. His main contribution was the design and the implementation of a microprocessor-based fine tracking system combined with a low-light level CCD detector. In 1983, he joined the staff of the École de Technologie Supérieure, Université du Québec, in Montréal where he co-founded the Dept. of Automated Manufacturing Engineering where he is currently Full Professor and teaches Pattern Recognition, Evolutionary Algorithms, Neural Networks and Fuzzy Systems. In 1992, he joined also the Computer Science Department of the Pontifícia Universidade Católica do Paraná (Curitiba, Brazil) where he was, co-responsible for the implementation in 1995 of a master program and in 1998 a PhD program in applied computer science. Since 1996, he is a senior member of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI, Concordia University). Since 2012, he is the Research Chair holder specializing in Adaptive Surveillance Systems in Dynamic Environments. Dr Sabourin is the author (and co-author) of more than 300 scientific publications including journals and conference proceeding. He was co-chair of the program committee of CIFED'98 (Conférence Internationale Francophone sur l'Écrit et le Document, Québec, Canada) and IWFHR'04 (9th International Workshop on Frontiers in Handwriting Recognition, Tokyo, Japan). He was nominated as Conference co-chair of ICDAR'07 (9th International Conference on Document Analysis and Recognition) that has been held in Curitiba, Brazil in 2007. His research interests are in the areas of adaptive biometric systems, adaptive surveillance systems in dynamic environments, intelligent watermarking systems, evolutionary computation and biocryptography.

Eric Granger obtained a Ph.D. in Electrical Engineering from the École Polytechnique de Montréal in 2001, and from 1999 to 2001 he was a Defence Scientist at Defence R&D Canada in Ottawa. Until then, his work was focused primarily on neural network signal processing for fast classification of radar signals in Electronic Surveillance systems. From 2001 to 2003, he worked in R&D with Mitel Networks Inc. on algorithms and ASICs to implement cryptographic functions in IP-based communication platforms. In 2004, Dr. E. Granger joined the École de technologie supérieure (Université du Québec), where he presently holds the rank of Assistant Professor in the département de génie de la production automatiseé. Since joining ÉTS, he has been a member of the Laboratoire d'imagerie, de vision et d'intelligence artificielle (LIVIA), and his main research interests are adaptive classification systems, incremental learning, ambiguity and novelty detection, neural and statistical classifiers, and multiclassifier systems, with applications in biometrics

(authentication from signatures and faces), military surveillance (recognition of radar signals), and computer and network security (intrusion detection). To date, he has authored/co-authored over 50 open literature papers with peer reviews, and supervised/co-supervised 20 graduate students in these areas of research. In 2011, he has been named member of the Task Force on Biometric of IEEE CIS ISATC.