Anti-Forensics of Median Filtered Images Using Non-Linear Optimization Techniques

Major Project submitted in partial fulfillment of the requirements for the award of degree of

Master of Technology

in

Information System

Submitted By:

Himanshu Singh

(2K11/ISY/09)

Under the Guidance of:

Ms. Ritu Agarwal

(Assistant Professor)

(Department of Information Technology)

Department of Information Technology

Delhi Technological University

(2011-2013)

CERTIFICATE

This is to certify that **Himanshu Singh (2K11/ISY/09)** has carried out the major project titled **"Anti-Forensics of Median Filtered Images Using Non-Linear Optimization Techniques"** as a partial requirement for the award of Master of Technology degree in Information System by **Delhi Technological University**.

The major project is a bonafide piece of work carried out and completed under my supervision and guidance during the academic session **2011-2013**.The matter contained in this report has not been submitted elsewhere for the award of any other degree.

(Project Guide)

Ms. Ritu Agarwal

Assistant Professor

Department of Information Technology

Delhi Technological University

Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I express my gratitude to my major project guide **Ms.Ritu Agarwal, Assistant Professor, Department of Information Technology** for the valuable support and guidance she provided in making this major project. It is my pleasure to record my sincere thanks to my respected guide for her constructive criticism and insight without which the project would not have shaped as it has.

I also extend my sincere regards to **Dr. O P Verma, HOD, Department of Information Technology** for being the guiding light throughout in my journey.

I humbly extend my words of gratitude to other faculty members, staff members and my friends for providing their valuable help and time whenever it was required.

Himanshu Singh

Roll No. 2K11/ISY/09

M.Tech (Information System)

E-mail: himanshu21588@gmail.com

Abstract

Digital image forensics is fast evolving field with many applications as the digitization is growing. Digital images have become more prone to tampering and forgery. Median filtering detection emerged as widely used tool against tampering in images. Recently a wide variety of techniques have been evolved for median filtering detection in digital images. This median filtering detection is done in blind fashion as the investigator is unaware of the post processing done on the image. Detection of median filtering is an important task in image forensics, since this operator is frequently used both for benign and malicious processing. It is also seen that median filtering detection is also used to hide traces from images so that preprocessing steps doesn't leave any footprints on the image. This is why counter forensics is evolved. In the proposed method particle swarm optimization is applied to minimize the function derived from statistics of detection of median filtering in images in the Yuan method. Yuan method is based on calculation of feature set comprised of five discriminate features used for detection of median filtering. Particle swarm optimization is a meta-heuristic tool used for the minimization of the objective function of the characteristic feature set.

Contents

List of Figures

Chapter 1

Brief Overview

1.1 Introduction

Ever since the evolution of technology, we heavily rely on the use of it. Technology is anything and everything which reduces human labor and work on the betterment of it. Since the time technology is imbibed for human use, there is a side in which it is misused and malpractices have been employed to forge people. This research focuses on studying the detection of median filtering in digital images which is a good forensic tool and applying anti-forensics on detection of median filtering. As median filtering is much of a research topic recently. Various methods have been developed to detect median filtering in digital images. Image forgery and image tampering has been growing since very long. Various methods have been deployed to overcome these malpractices. Image tampering, slicing etc. are some of the malpractices that have been occurred. It's been difficult to prove the identity of images. In the court of law, the evidences are forged and are a cause of acquittal of criminals. To stop these malpractices, it is been made sure that some effective steps are taken to eradicate image forgeries. Median filter is an effective forensic analyzer and it is been shown that image forgeries can be detected using this feature.

Anti-forensics has been developed to hide the traces left by median filter on digital images. It is used as a malicious practice by counterfeiters so that they can hide the traces after forgeries which make it difficult to detect median filtering. This causes a serious problem for forensic analyzers. The other way round is use of anti-forensic techniques by forensic investigators to hide traces of

median filtering thus making it difficult for counterfeiters to exploit the features for detection of median filtering. This approach is a benign practice of hiding traces used recently.

1.1.1 Forensic science

Forensic science is a field which is used to investigate and establish facts of interest against a criminal. Forensics comes from a latin word '*forensis'* which means "of or before the forum". In modern times forensics is the term widely used instead of forensic science. Forensics is sub-divided into different fields from which the field of our interest is digital forensics.

1.1.2 Digital forensics

Digital forensics is a branch of forensic science which deals with the recovery and investigation of material found in digital devices in relation to a computer crime. Digital forensics is divided into different branches relating to the investigation of various types of devices :-

- Computer forensics
- Mobile device forensics
- Network forensics
- Forensic data analysis
- Database forensics

1.2 Computer forensics

Computer forensics is a specialized branch pertaining to legal evidence found in computers. Computer forensics mainly deals with examining a digital media or computer forensically with an aim of identifying, preserving, recovering, analyzing and presenting facts and opinions about information.

1.2.1 Counter forensics

Counter forensics deals with the techniques used as counter measures to computer forensics. More widely used definition was given by Dr. Marc Rogers of Purdue University.

It is “Attempts to negatively affect the existence, amount and quality of evidence from a crime scene, or make the analysis and examination of evidence difficult or impossible to conduct.”

This thesis is a work in which we examine the techniques of computer forensics to detect the median filtering in digital images. Then we propose a counter forensic approach using particle swarm optimization.

1.3 Organization of thesis

In chapter 2, we look into the details of digital forensic science. We also understand median filtering and why it is relevant in image forensics. We also look into detailed techniques used for detecting median filtering. We also explain the process of counter forensics of median filtered images.

In chapter 3, we understand particle swarm optimization and its working with a flow chart.

In chapter 4, we propose a counter forensic technique for median filtering detection in digital images using particle swarm optimization. We also provide some results of already established median filtering detection techniques.

In chapter 5, we establish some experimental results of proposed method and compare it with already existing techniques.

Chapter 2

Forensics: A Theoretical Review

2.1 Formation of Digital Images

A digital image is made of a complex image generation process that converts a continuous scene to a discrete form. A scene can refer to any natural phenomena or can explain an arbitrary imaginary phenomenon that comes from human creativity.[1] As a result, the image conveys information about the depicted scene, which by human interpretation translates to a particular semantic meaning. The image generation process itself may consist of up to three principal stages:

- The image acquisition phase is the interface between the real and the digital world, where a scene can be represented having discrete projection. These projections can be obtained via an image acquisition device that has a sensor (e. g., a digital camera).
- In the processing phase, the acquired image is processed, leading to a improved version. At this stage, we distinguish between image processing and image manipulation. While processing in general refers to any change of the initial image, a manipulation is more specific and implies the change of the image's semantic meaning. In other words, a manipulation destroys the authenticity of a digital image.
- Finally, the digital image can undergo a re-digitization procedure, where the image is recaptured from the result of an output device such as printers etc. We note that re-digitization can be understood as a special form of post-processing: the resulting digital image is a modified version

of the image fed into the output device. We will make use of either perspective, depending on whether the explicit or implicit view is more convenient in a particular situation.

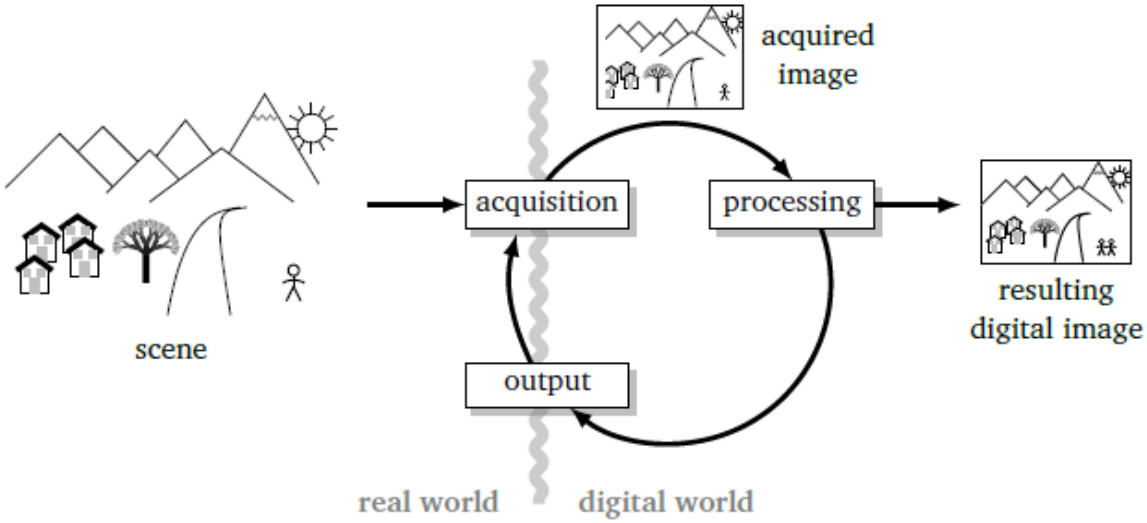


Figure 2.1 A digital image generation process showing the real world and digital world.

2.2 Digital Image Forensics

2.2.1 Definition

Digital Forensic Research Workshop has defined digital forensics as “The use of scientifically derived and proven methods toward the preservation, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations."[2]

2.2.2 Description

The primary objective of forensic sciences is the systematic reconstruction of events as well as identification of entities involved. The forensic analysis of digital images (or digital imageforensics) then refers to the reconstruction of the generation process of a given digital image, where the main focus lies on inference about the image's authenticity and origin.[3]

Investigative process of digital forensics can be divided into several stages. According to [2][4][5][6], there are four major stages: preservation, collection, examination, and analysis.

- Preservation - Preservation stage corresponds to "freezing the crime scene". It consists of preventing any activities that can damage digital information being collected. Preservation involves various operations such as preventing people from using computers during collection of data and files, stopping ongoing deletion processes such as hard drives, and choosing the safest way to collect information.

- Collection - Collection stage consists in finding and collecting digital information that may be relevant to the investigation of the crime. Since digital information is stored in computers, collection of digital information means either collection of the equipment containing the information, or recording the information on some medium. Collection may involve removal of personal computers from the crime scene, copying or printing out contents of files from a server, recording of network traffic, and so on.

- Examination - Examination stage consists in an in-depth systematic search of evidence relating to the incident being investigated. The outputs of examination are data objects found in the collected information. They may include log files, data files containing specific phrases, timestamps, and so on.

- Analysis - The aim of analysis is to draw conclusions based on evidence found.

2.3 Variants of Digital Image Forensics

We have considered digital image forensics in a very general way. In particular, we have not made any assumption with regard to the access the forensic investigator has to the components of the image generation process or to their respective inputs and outputs. On the basis of nature of investigation different types of forensics are involved.

2.3.1 Passive vs. Active Image Forensics

Digital image forensics is called passive forensics if the forensic investigator cannot interfere with the image generation process and control type and appearance of identifying traces. The image generation process is considered as a 'read-only' procedure and the forensic investigator is confined to examine image characteristics that are generated by this process. Identifying traces in passive image forensics in general is divided into device characteristics and processing artifacts. [23] The device characteristics refer to inherent variations between different acquisition devices and thus allow to know about the origin of a given digital image. Such variations may exist because manufacturers use different components or adjust parameter settings for different devices. They can also be caused by unwanted inherent technological imperfections, such as sensor defects. Processing artifacts, on the other hand, relate to identifying traces that are introduced by post-processing the acquired digital image.

Hence, they are used as a means to assess image authenticity. Similar to device characteristics, different processing procedures may vary in the characteristics of resulting traces and thereby allow to know about the type of processing. Both device characteristics and processing artifacts can be tested for their presence and consistency, whereby inconsistent device characteristics are themselves a processing artifact.

Active approaches differ from passive approaches in that the generation process is purposely modified at an earlier stage to leave behind specific identifying traces. This auxiliary data, would—once being tied to the image—establish a link to the image's origin or ensure the image's authenticity, respectively. Typical instances of active approaches attach metadata to the image (e. g., a cryptographic signature [7] or a robust hash [8]) or embed a digital watermark directly into the image itself [9].

Identifying traces in active image forensics are designed to link the resulting image to its origin, or to be sensitive ('fragile') to (certain types of) image post-processing. By testing for their presence and consistency, these traces allow inference about the responsible component itself as well as subsequent components in the image generation process. In contrast to identifying traces in passive approaches, type and appearance of such traces can be chosen in anticipation of potential subsequent steps of the image generation process. They can also be designed based on cryptographic protocols to guarantee trustworthiness by means of mathematical proofs. Active approaches ideally need to be implemented directly in the acquisition device. It is not possible to infer parts of the generation process prior to the active insertion of respective traces. This reliance on special-purpose hardware is one of the major drawbacks of active image forensics: it does not allow to assess the trustworthiness of arbitrary images of unknown provenance. Moreover, recent examples of hacked active authentication systems of major digital camera vendors suggest that

such systems create a deceptive impression of trustworthiness when implementation errors cannot be ruled out.

2.3.2 Blind vs. Non-Blind Image Forensics

Digital image forensics is called blind if the forensic investigator is confined to examine the final output of the generation process which is the resulting image after post-processing. In particular, knowledge neither of the original scene nor any intermediate result of the generation process is available at the time of analysis. This includes the forensic investigator's uncertainty about whether the image under analysis has been subject to any kind of post-processing. A blind analysis however not necessarily implies that the forensic investigator does not have any knowledge of or assumptions about potential components of the image generation process. This applies in particular to active image forensics, because the image generation process has been purposely modified by the investigator at an earlier stage. But also passive forensic investigators may rely on such information. For instance, it is often known with reasonably high certainty, which device captured the image when it is analyzed for inconsistent device characteristics and potential manipulations [10].

Additional information about intermediate results helps non–blind forensic investigators to dis-entangle scene and generation process characteristics, respectively, and hence to make more informed decisions. Such data may be available from alternative sources (for instance, earlier versions of a processed image that have been published elsewhere), or could have been stored purposely in advance most likely the acquired image. Side information about the original scene may also be retrieved from other trustworthy images that exist of the same scene. Non–blind approaches in general have the advantage to mitigate some of the forensic investigator's

uncertainty. At the same time, they are often unviable in practical settings. In particular, non–blind forensics precludes the examination of arbitrary images of unknown provenance.

Swaminathan et al. [11] further divide non–blind forensics into semi–intrusive and intrusive approaches. A semi–intrusive analysis considers the image generation process as a black box, which is fed known input signals and allows to know about the overall input-output relation. In an intrusive analysis, the forensic investigator also has knowledge of the inputs and outputs of individual components of the image generation process and aims to determine their specific instantiation. This distinction becomes particularly relevant when the forensic investigator is less interested in assessing the trustworthiness of an image, but really in "reverse engineering" parameters of the image generation process to which she is granted access.

It is then possible to design special inputs, either to the complete generation process or to individual components, which are particularly suitable to analyze and tell apart identifying traces of different instantiations [12].

2.3.3 Passive–Blind Image Forensics

We focus on passive–blind image forensics [13] and use this term synonymously with digital image forensics. It does not follow from the above discussions that passive–blind forensic investigators can access or control components of the image generation process passive nor have knowledge of its inputs or about intermediate results.

Their analysis is solely based on inherent device characteristics and processing artifacts, which we extract from the image under investigation to infer particulars of the image generation process. Because the forensic investigator's view is restricted to the image under analysis, passive–blind image forensics will always remain an inexact science. With digital images being projections of the infinite set of all conceivable scenes, the forensic investigator can never fully know whether a

depicted scene is indeed a valid representation of the real world, or whether the image has been manipulated in some way. In general, there will always exist a residual probability that a particular scene imposes certain image characteristics that appear to the forensic investigator as device characteristic or processing artifact and thus lead to false decisions. Moreover, any image generation process inevitably involves quantization. By definition, quantization causes information loss and hence leaves the forensic investigator with an additional source of uncertainty. As a consequence, any result of passive–blind image forensics has to be understood as indication, which per se excludes absolute conclusions.

However, the advantage of passive–blind image forensics is its universal applicability. In particular, it is applicable to the analysis of arbitrary images of unknown provenance. Passive–blind image forensics does not rely on a closed infrastructure of trustworthy special-purpose acquisition devices that actively introduce identifying traces to the resulting images. While this might be a viable option for relatively small-scale applications, it is certainly too costly and most likely politically unviable to be implemented in typical consumer devices.

2.4 Median filter

The median filter is normally used to reduce noise in an image, somewhat like the mean filter. However, it often does a better job than the mean filter of preserving useful detail in the image. This class of filter belongs to the class of edge preserving smoothing filters which are non-linear filters. This means that for two images A(x) and B(x):

$$median[A(x) + B(x]) \neq median[A(x)] + median[B(x)]$$

These filters smoothes the data while keeping the small and sharp details. The median is just the middle value of all the values of the pixels in the neighborhood. Note that this is not the same as the average (or mean); instead, the median has half the values in the neighborhood larger and half smaller. The median is a stronger "central indicator" than the average. In particular, the median is hardly affected by a small number of discrepant values among the pixels in the neighborhood. Consequently, median filtering is very effective at removing various kinds of noise. [15]

Like the mean filter, the median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the *mean* of neighboring pixel values, it replaces it with the *median* of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. Figure 2.2 illustrates an example calculation

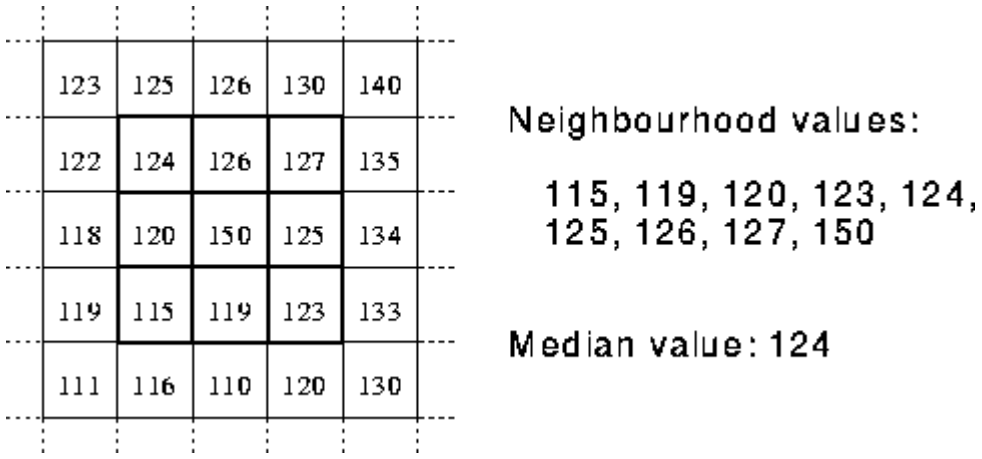


Figure 2.2 Calculating the median value of a pixel neighborhood. As can be seen, the central pixel value of 150 is rather unrepresentative of the surrounding pixels and is replaced with the median value: 124. A 3×3 square neighborhood is used here --- larger neighborhoods will produce more severe smoothing.

2.5 Counter-forensics

Counter-forensics is the collective term for techniques intended to complicate, inhibit, subvert, or delay forensic techniques for finding evidence. It can be seen as encompassing a broad range of techniques from subtle and highly sophisticated data altering techniques to methods as crude as smashing evidential hard drives with a hammer. The purpose of counter-forensics is to make sure that evidence is not discovered and subsequently disclosed to a court, arbitrator, or some other forum. Additionally, in most cases at least some attempt is made to disguise the fact that evidence is being altered or withheld. In the vast majority of cases such tampering with evidence will damage the interests of those using these counter-forensic techniques. [14]

Counter-forensics is sometimes seen as being mostly about evidence destruction or erasure, but this is not the whole story. In many instances, particularly in respect to evidence in civil cases, it may not be necessary for counter-forensic techniques to destroy or erase data on evidential media. It is enough if they make it more difficult for an investigator or analyst to recover the data.

A thorough assessment of the reliability of digital image forensics algorithms requires anticipating strategies of potential counterfeiters. Whenever an image is manipulated purposely (and with the intention to make it public to a certain group of entities), the counterfeiter will have at least a rough working definition of what is considered plausible and will try to conform to these expectations. Because knowledge of identifying traces of image generation functions is in general not limited to the forensic investigator, informed counterfeiters will eventually exploit their own knowledge (or, to be more precise: assumptions) to deceive forensic analyses, to influence the outcomes of digital image forensics algorithms. For a characterization of such attacks it is important to distinguish between the robustness and the security of digital image forensics algorithms. Because we expect counterfeiters to possess different skills and to have access to different resources, it seems feasible

to adopt the notion of adversary models to study the reliability of image forensics algorithms conditional to those expectations and to the potential strategies that counterfeiters can pursue.

2.6 Mean Squared Error (MSE)

The mean squared error (MSE) of an estimator is one of the ways to quantify the difference between values implied by an estimator and the true values of the quantity being estimated. MSE is a risk function, corresponding to the expected value of the squared error loss or quadratic loss. MSE measures the mean of the squares of the "errors." The error is the amount by which the value implied by the estimator differs from the true value of the quantity to be estimated. The difference occurs because of randomness or because the estimator doesn't account for information that could produce a more accurate estimate.

If $\hat{Y}_i$ is a dataset of n estimations, and Y_i is the dataset of the true values, then the (estimated) MSE of the estimator is:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{Y}_i - Y_i\right)^2.$$

2.6.1 Peak Signal-to-Noise Ratio (PSNR)

Peak signal-to-noise ratio is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is usually expressed in terms of the logarithmic decibel scale because many signals have a

very wide dynamic range. PSNR is most commonly used to measure the quality of reconstruction from sub-sample image data (e.g., for .image demosaicking). A higher PSNR generally indicates that the reconstruction is of higher quality.

PSNR can be easily defined via the mean squared error (*MSE*). Given a loss-less *m*×*n* monochrome image *I* with 255 as its maximum grey level, *PSNR* is defined as:

$$PSNR = 10.log\left[\frac{255^2}{MSE}\right]$$

whereMSE is the mean squared error, as explained in the above section.

2.6.2 CPU Time

CPU time (or CPU usage, process time) is the amount of time for which a central processing unit (CPU) was utilized for processing instructions of a computer program.The CPU time is often measured in clock ticks or seconds.

Chapter 3

Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Ebehart and Dr. Kennedy in 1995, [16] inspired by social behavior of bird flocking or swarm intelligence. PSO shares many similarities with evolutionary genetic computation techniques such as Genetic Algorithms (GA). [17][18][19] The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. The detailed information will be given in following sections. Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied.

3.1. BACK GROUND OF ARTIFICIAL INTELLIGENCE:

The term "Artificial Intelligence" (AI) is used to describe research into human-made systems that possess some of the essential properties of life. AI includes two-folded research topic.

- AI studies how computational techniques can help when studying biological phenomena
- AI studies how biological techniques can help out with computational problems

Actually, there are already lots of computational techniques inspired by biological systems. For example, artificial neural network is a simplified model of human brain; genetic algorithm is inspired by the human evolution. Here we discuss another type of biological system - social

system, more specifically, the collective behaviors of simple individuals interacting with their environment and each other. Someone called it as swarm intelligence. All of the simulations utilized local processes, such as those modeled by cellular automata, and might underlie the unpredictable group dynamics of social behavior. Some popular examples are bees and birds. Both of the simulations were created to interpret the movement of organisms in a bird flock or fish school. These simulations are normally used in computer animation or computer aided design. There are two popular swarm inspired methods in computational intelligence areas: Ant colony optimization (ACO) [20] and particle swarm optimization (PSO). ACO was inspired by the behaviors of ants and has many successful applications in discrete optimization problems. The particle swarm concept originated as a simulation of simplified social system. The original intent was to graphically simulate the choreography of bird of a bird block or fish school. However, it was found that particle swarm model could be used as an optimizer.

3.2 PARTICLE SWARM OPTIMISATION:

PSO simulates the behaviors of bird flocking. Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration. The effective way is to follow the bird, which is nearest to the food. PSO learned from the scenario and used it to solve the optimization problems. In PSO, each single solution is a "bird" in the search space. We call it "particle". All of particles have fitness values, which are evaluated by the fitness function to be optimized, and have velocities, which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles.

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values.

The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called g-best. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called p-best. After finding the two best values, the particle updates its velocity and positions with following equation (3.1) and (3.2).

$$V_i^{u+1} = w \times V_i^u + c_1 \times rand \times (pbest_i - P_i^u) + c_2 \times rand \times (gbest_i - P_i^u) \qquad (3.1)$$

$$P_i^{u+1} = P_i^u + V_i^{u+1} \qquad (3.2)$$

In the above equation, the term rand×$(pbest - P)$ is called particle memory influence. The term rand ×(gbest-P) is called swarm influence.

V_i^uwhich is the velocity of ith particle at iteration 'u', must lie in the range Vmin ≤ V_i^u ≤ Vmax

- The parameter Vmax determines the resolution, or fitness, with which regions are to be searched between the present position and the target position.
- If Vmax is too high, particles may fly past good solutions. If Vmin is too small, particles may not explore sufficiently beyond local solutions.
- In many experiences with PSO, Vmax was often set at 10-20% of the dynamic range on each dimension.
- The constants C1and C2 pull each particle towards pbest and *gbest* positions.
- Low values allow particles to roam far from the target regions before being tugged back. On the other hand, high values result in abrupt movement towards, or past, target regions.

- The acceleration constants C1 and C2 are often set to be 2.0 according to past experiences.
- Suitable selection of inertia weight w provides a balance between global and local explorations, thus requiring less iteration on average to find a sufficiently optimal solution.
- In general, the inertia weight *w* is set according to the following equation,

$$W = W_{max} - \left[\frac{W_{max} - W_{min}}{ITER_{max}}\right] \times ITER$$

Where W - is the inertia weighting factor

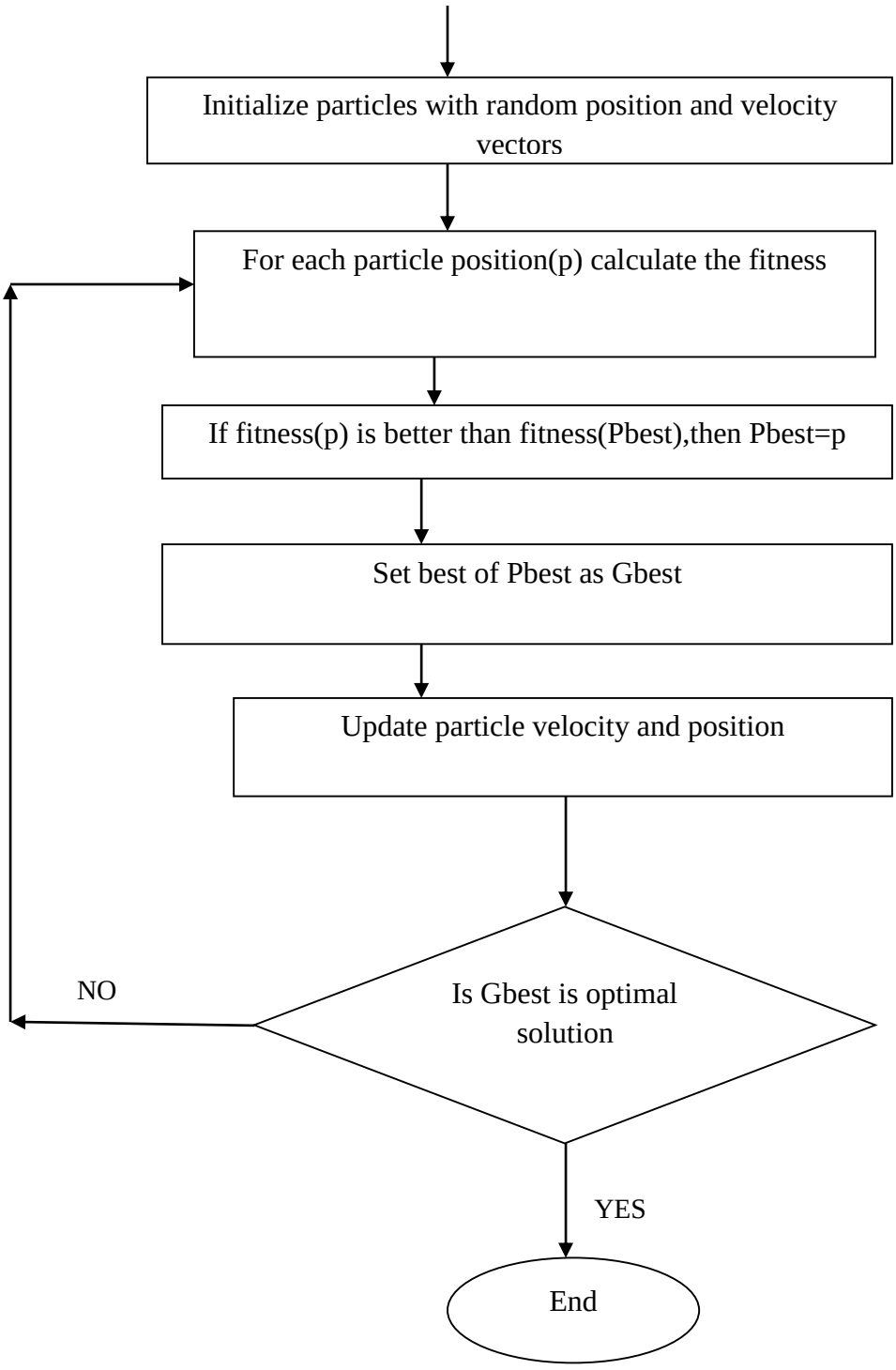
Wmax - maximum value of weighting factor

Wmin - minimum value of weighting factor

$ITER_{max}$ - maximum number of iterations

ITER - current number of iteration

3.2.1 FLOW CHART


Initialize particles with random position and velocity vectors
For each particle position(p) calculate the fitness
If fitness(p) is better than fitness(Pbest),then Pbest=p
Set best of Pbest as Gbest
Update particle velocity and position
Is Gbest is optimal solution
NO
YES
End

Chapter 4

Proposed scheme

4.1 Literature Review

Median filter is a non-linear filter widely used to remove noise and smoothening of images. It is used in the post processing of images. The forensics involves detection of this median filtering and exploit the fingerprints left on the digital images. [22][23] Several techniques had been developed for detection of median filtering in the recent past.

4.1.1 Kirchner et. al method

Kirchner et. al [21] proposed a median filtering detection technique based on first order differences. They present two measures based on first order difference $d_{i,j}^{(p,q)}$ denote the first order difference with lag (p,q)∈{(1,0),(0,1),(-1,0),(0,-1)} then,

$$d_{i,j}^{(p,q)} = y_{i,j} - y_{i+p,j+q}$$

And

$$H^{(p,q)} = \{\dots, h_{-1}^{(p,q)}, h_0^{(p,q)}, h_1^{(p,q)}, \dots\}$$

is corresponding histogram differences. It is been seen that median filtering is increased in h_0 than in $h_{+1/-1}$. Considering this as the discriminating feature, the ratio

$$\rho^{(p,q)} = {h_0^{(p,q)}}\Big/{h_1^{(p,q)}}$$

where$\rho \gg 1$ indicates median filtering.

The strong saturation effects in the original image will render the detection of median filtered images by means of ρ unreliable. To obtain a more robust discriminating feature, we divide the image under investigation into the set B of non-overlapping blocks of dimension B × B. By determining ρ_b as the ratio of histogram bins h0 and h1 from the b-th block, the influence of saturated image blocks can be reduced by taking the weighted median

$$\hat{\rho} = median(w_b \rho_b)$$

as a robust detection feature. Here, the weights wb function as an attenuation factor for saturation effects.[24] In the course of this paper, we set wb to

$$w_b = 1 - (\frac{h_0}{B^2 - B})$$

giving less weight to strongly saturated blocks.

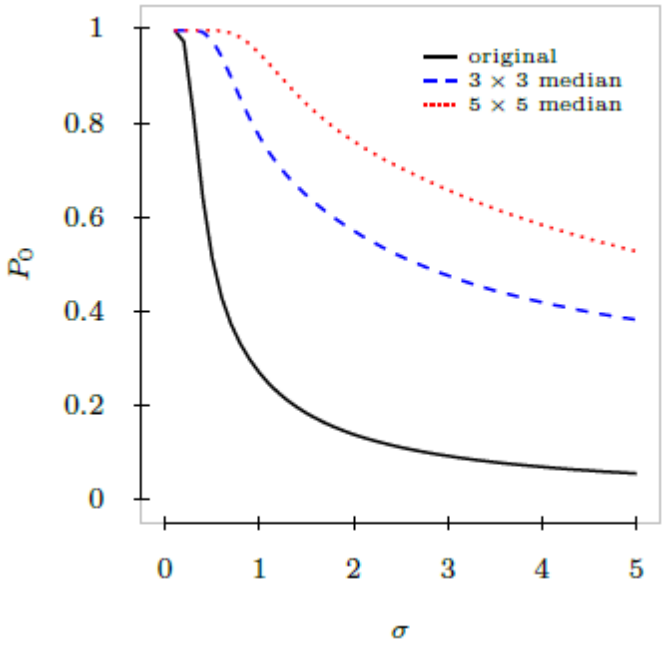


Figure 4.1. Streaking probabilities P0 (direct vertical or horizontal neighbors) for quantized i.i.d. Gaussian samples input samples with variance σ^2

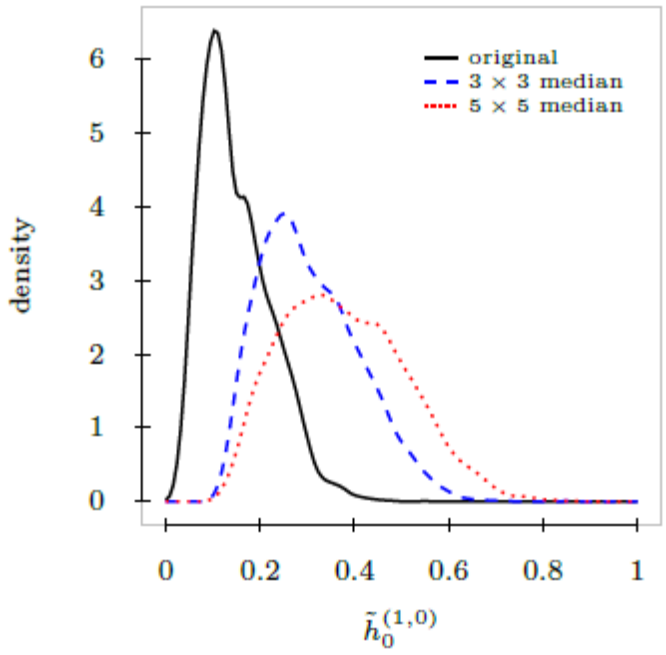


Figure 4.2. Density estimates for relative frequencies $\widetilde{h_0^{(1,0)}}$ from 6500 original images and their 3×3 median filtered versions, respectively.

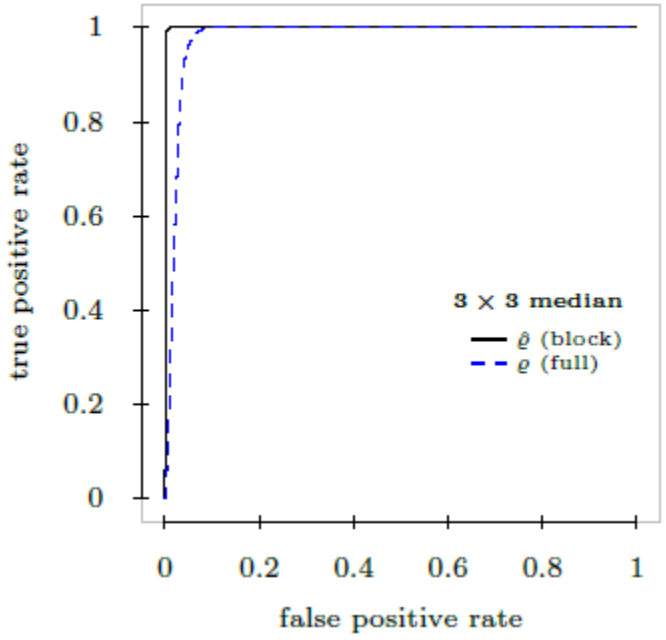


Figure 4.3. Detection results for 3×3 median filtering. ROC curves for $\rho$and $\hat{\rho}$, (B = 64). The block based approach is more robust to false alarms.

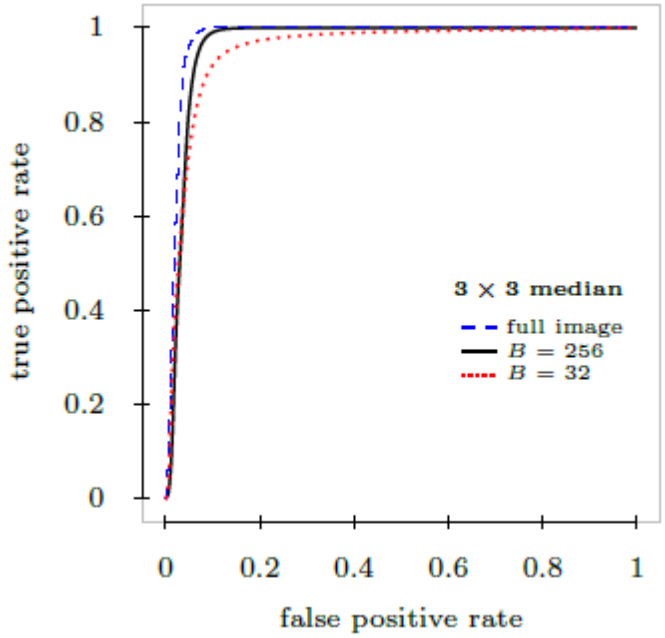


Figure 4.4. Detection results for 3×3 median filtering. ROC curves for ρ and ρ_b for varying block sizes. Smaller blocks are less robust to local image characteristics.

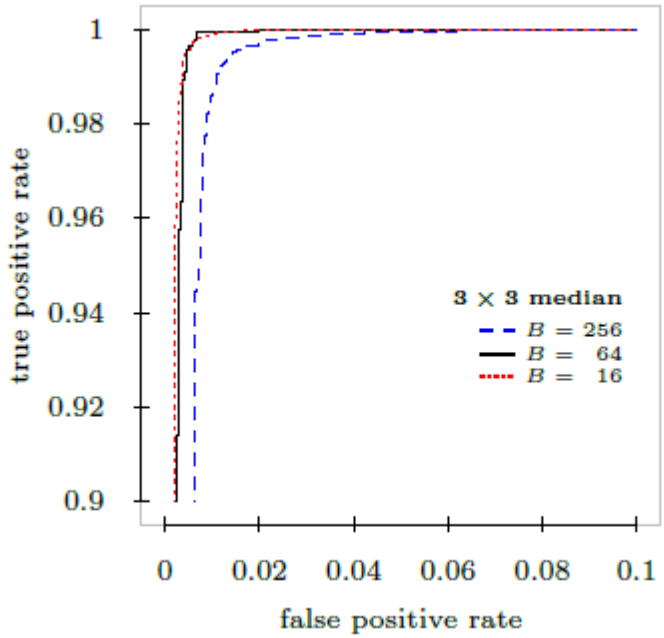


Figure 4.5. Detection results for 3 × 3 (left) and 5 × 5 (right) median filtering, respectively. ROC curves for different block sizes B. Vertical differences with lag (k, l) = (1, 0). Smaller block sizes increase detection performance.

4.1.2 Cao et. al method

A similar consideration leads the work by Cao et al [25]. The authors observe that, in presence of median filtering, it is much more likely that the difference between two adjacent pixel is exactly zero. To explore the presence of this footprint, they compute and binarize the row-based first order difference ΔIr as follows:

$$\Delta I_r(i,j) = \begin{cases} 1 & \text{if } \mathrm{I(i+1,j)} - \mathrm{I(i,j)} = 0 \\ 0 & \text{if } \mathrm{I(i+1,j)} - \mathrm{I(i,j)} \neq 0 \end{cases}$$

where I is the image under analysis. In the same way, they also compute column-based difference ΔIc(I, j) for each pixel. Obviously highly textured regions will rarely show equal adjacent pixels, independently of median filtering, and this must be taken into account: a map V (i, j) is computed evaluating for each pixel the variance of the surrounding region. Using the first order difference and the variance map, the actual features of the scheme are computed as:

$$f_r = \sum_{i,j} \frac{\Delta I_r(i,j) \cdot V(i,j)}{V(i,j)}$$

The final scalar feature ρ is obtained as

$$\rho = [f_r, f_c] \cdot [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$$

where· denotes the dot product.

4.1.3 Yuan method

Yuan [26] proposed a more elaborate approach for median filter detection which takes non overlapping blocks of size s×s(s is odd) to exploit the local dependence of artifacts of overlapping blocks in median filtering. To achieve this he computes a feature set consisting five individual discriminating features

- Distribution of block median denoted as h^{DBM} indicates that in median filtered image gray levels in small block tends to equal to the block median.
- The occurrence of the block center gray level denoted as h^{OBC} indicates that in a median filtered image the block center gray level occur frequently in the block.
- Quantity of gray levels in a block denoted as h^{QGL} indicates that the number of gray levels is decreased in a block because median filter smoothes the image without introducing new gray levels.
- Distribution of block center gray level in the sorted gray levels denoted as h^{DBC} takes into account the sorted gray levels and indicates the frequency of block center gray level.
- First occurrence of the block center gray level in the sorted gray levels denoted as h^{FBC} considers the first occurrence of block center in sorted gray levels.

Based on these set of features a merged feature set is calculated as

$$f = \frac{h_5^{DBM} h_2^{OBC} h_6^{QGL} (h_3^{DBC} + h_7^{DBC} - h_2^{DBC} - h_8^{DBC}) h_3^{FBC}}{h_1^{OBC} h_9^{QGL} (h_2^{DBC} + h_8^{DBC} - h_1^{DBC} - h_9^{DBC}) h_2^{FBC} h_9^{FBC}} \quad 4.1$$

4.1.4 Fontani and Barni counter forensic technique

Fontani and barni [27] developed a counter forensic technique which takes into account the features calculated from yuan method and then tries to minimize the artifacts on which the median filtering in images is detected. They formulate the optimization problem to search for a processing p for a median filtered image M which when applied to M produces another image W such that W=p(M). this method tries to minimize the cost function of feature set f from yuan method denoted as f_M for the median filtered image M and generate f_W features for the image W. to measure the similarity between two images PSNR is selected. The optimization problem can be given as

$$\min\left(c(f_W) - PSNR(M, W)\right) \quad 4.2$$

For p ∈ P, W=p(M)

The optimization problem is solved using Nelder Mead optimization algorithm [28] which takes starting point a 3*3 filter window as a 9 dimensional vector.

4.2 Proposed scheme

we start from a median filtered image M, from which the algorithm by Yuan extracts the feature f_M (computed as in eq. 4.1). We define a cost function c : R → R that maps each value of the feature f to a cost, which grows as f increases. Then, given M, we are looking for a processing p ∈ P that produces an image W = p(M) whose extracted feature f_W has a cost c(f_W) as low as possible, while introducing as low distortion between M and W as possible. We choose PSNR as a measure of similarity, and define the following optimization problem as in eq. 4.2.

Instead of weighting the two components of the objective function with scalars, we directly choose the function c in such a way that a good tradeoff between footprint concealment and quality is retained; since very small displacements of f from typical values for unfiltered images allow the detector to discriminate well, we need a cost that grows rapidly when f moves away from a desired value f_0. Therefore, we choose an exponential cost function c(f):

$$c(f) = \exp(f - f_0)$$

Where f_0=2.2

We solve this optimization problem to reduce the cost function using particle swarm optimization. The starting point is taken as a 9 dimensional vector in the search space. The position and the velocity update are given by

- Velocity update

$$v_{id}^{t+1} = w.v_{id}^{t} + c_1.\alpha_1(p_{id}^{t} - x_{id}^{t}) + c_2.\alpha_2(p_{gd}^{t} - x_{id}^{t})$$

- Position update

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1}$$

v_{id}^{t} : Component in dimension d of the i^{th}particle velocity in iteration t.

x_{id}^{t} : Component in dimension d of the i^{th} particle position in iteration t.

c_1, c_2 : Constant weight factors

p_{id} : Best position achieved so long by particle i.

p_{gd} : Best position found by the neighbors of particle i.

α_1, α_2 : Random factors in the interval [0,1].

w : inertia weight

Chapter 5

Experimental results

UCID: This image database contains 1338 uncompressed RGB images [29]. The resolution of each image is 512×384. Many of the images have significant regions of either saturated pixels, or largely uniform intensity patches. Furthermore, many of the images are out-of-focus, or exhibit camera shake blur. Such images pose a challenge in distinguishing between the median filter and other smoothers. Before testing, the images are converted to gray scale.

BOWS2: This image database contains 10 000 gray-scale images with fixed size 512×512 coming from rescaled and cropped natural images of various sizes.

The experimental results of Yuan method taking images from these two databases have been shown.

Figure 5.1. Image taken from ucid database for yuan method

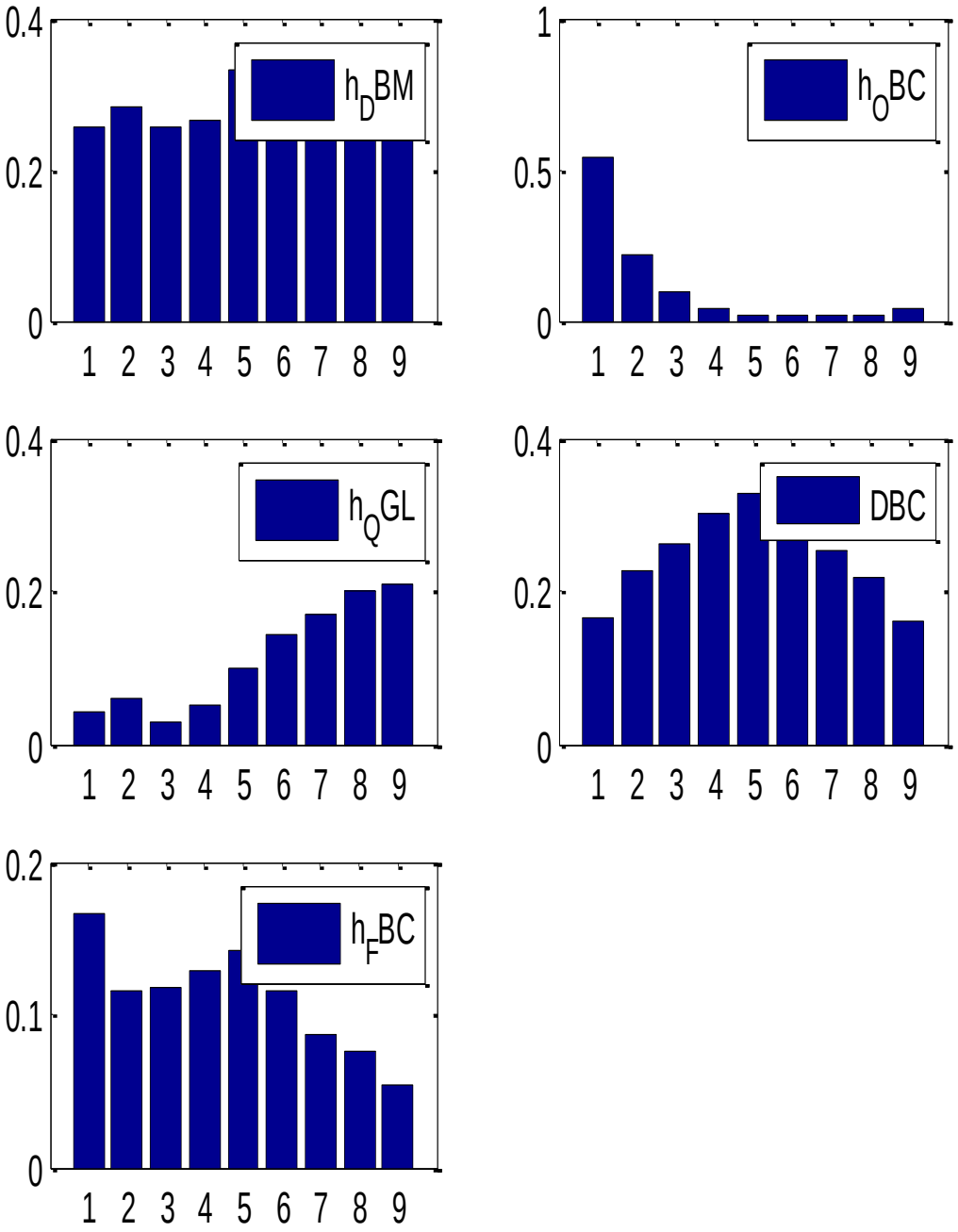


Figure 5.2. The histogram plot of the feature set calculated from yuan method for figure 1

Figure 5.3. Image taken from Bows2 database for Yuan method

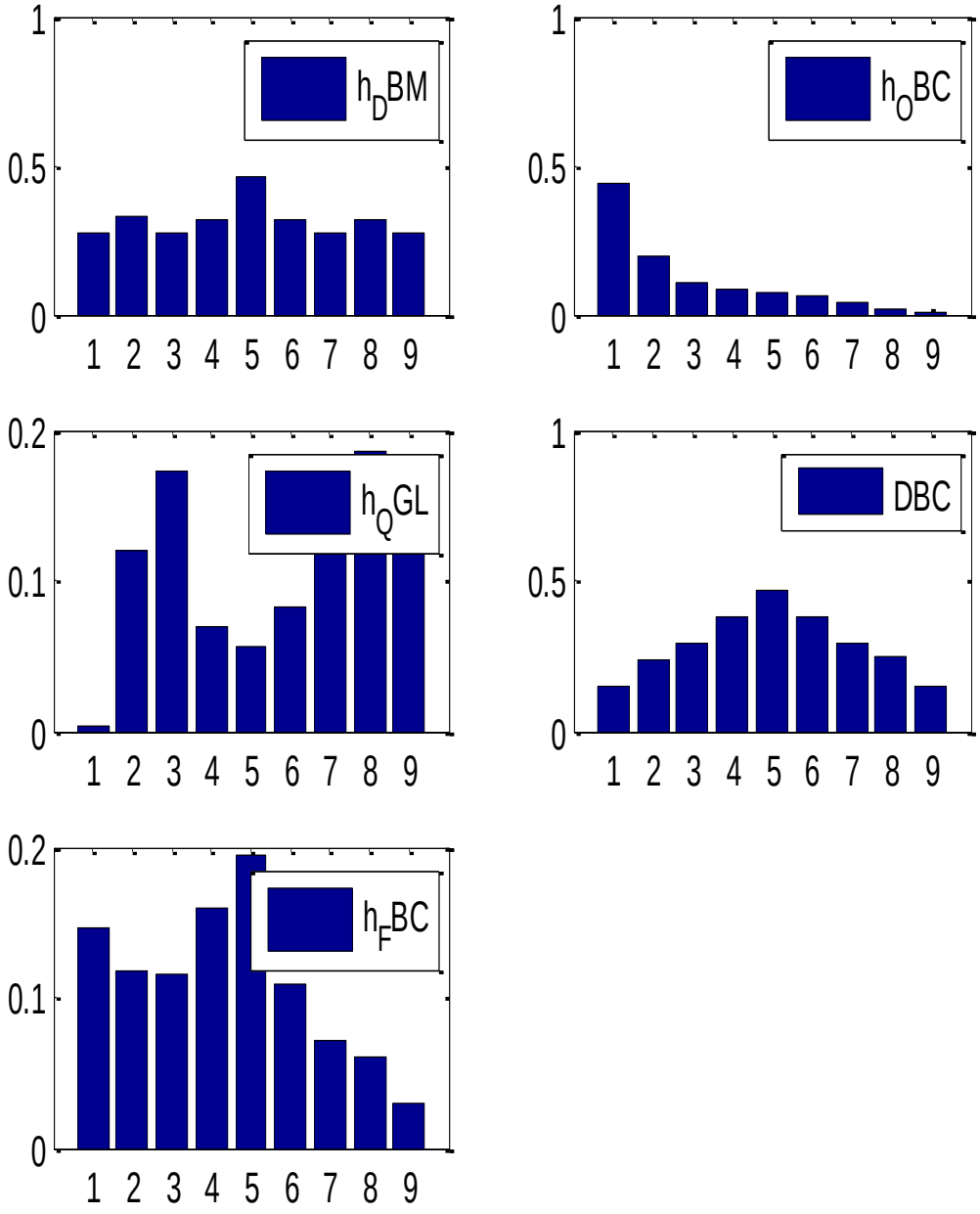


Figure 5.4. Feature set calculated from Yuan method

The proposed method is implemented on a database of 12 images taken from UCID database. The particle swarm optimization is implemented taking 10 particle swarm and with 10 iterations. 12 images taken are given below.

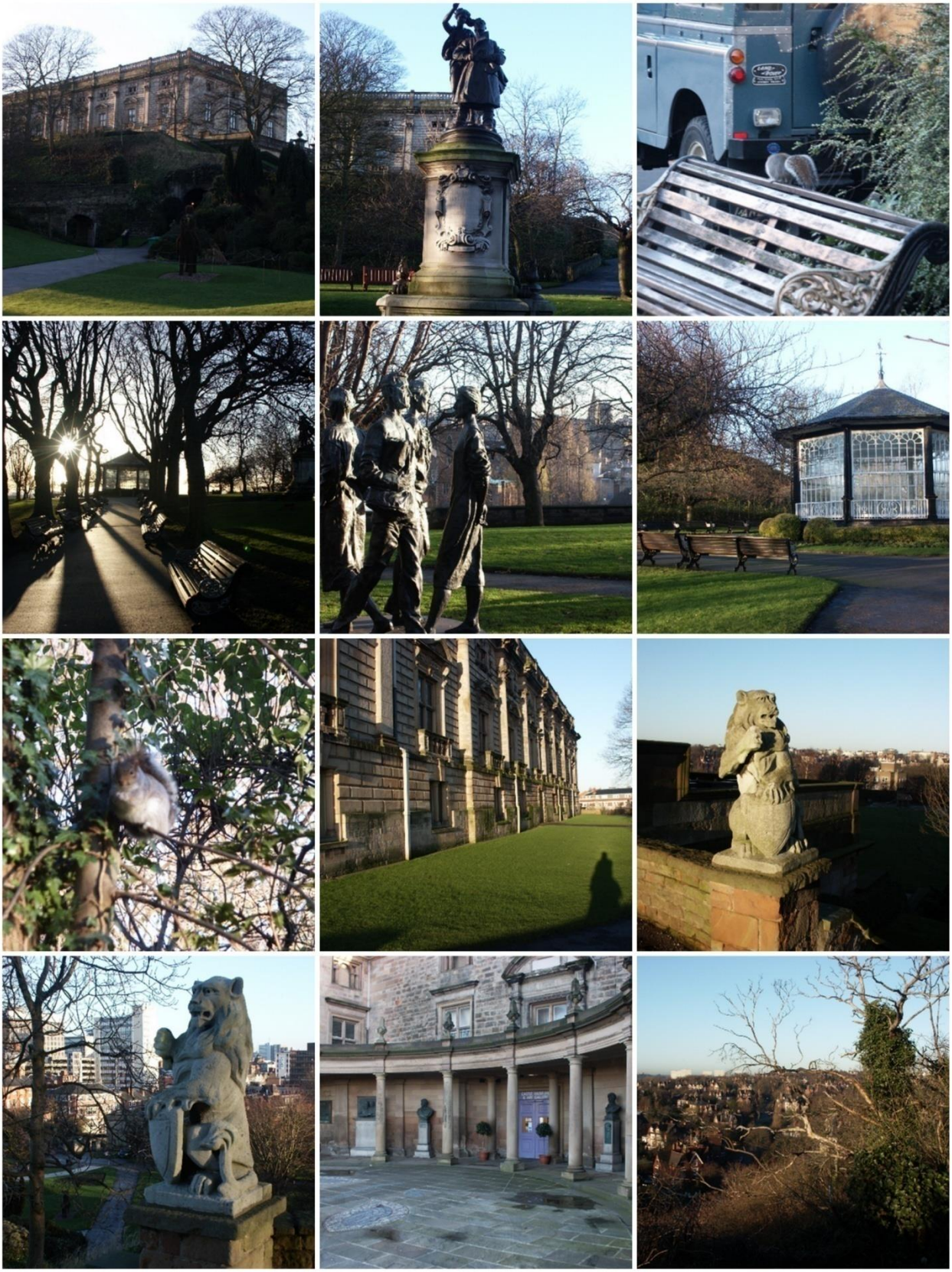


Figure 5.5. UCID database images on which the proposed scheme is implemented

The median filtered image M is as shown below. This image is now been used to find out an anti-median filtered image W with a processing P and the cost function is minimized using particle swarm optimization.

Figure 5.6. Median filtered image

The parameter values taken for PSO algorithm are:

Inertia weight w = 1.0

C1 = 2.05,C2 = 2.05

Figure 5.7. Anti-median filtered image

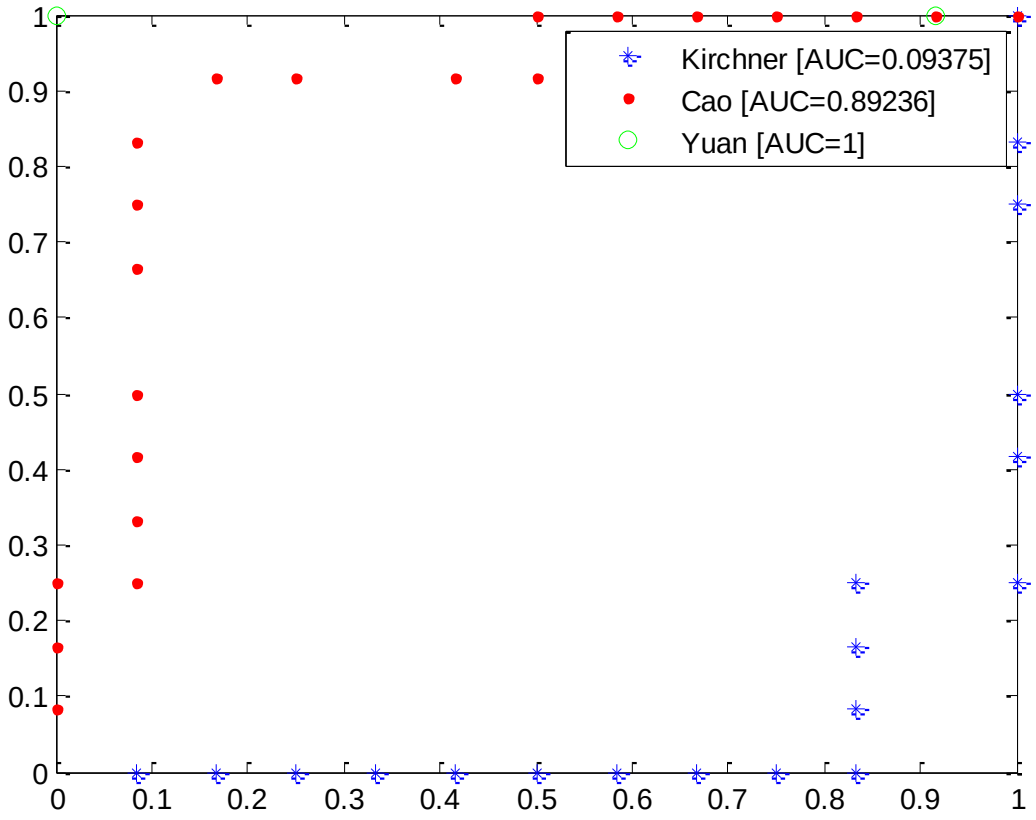


Figure 5.8. ROC characteristics graph shown for 12 images taken as input

The graph in figure 8 is a plot of Yuan Method, Cao et. al method, Kirchner et. al method. This graph is plot of true positive rate(x-axis) and false positive rate(y-axis)

5.1 Comparison with existing scheme

Fontani and Barni in their approach had solved the optimization problem using Nelder Mead optimization algorithm. The Nelder Mead optimization is a simplex method approach which takes input and move towards a solution which is minimized. The experimental results of optimization using Nelder Mead optimization for 12 images database taken previously same as for proposed approach are as shown below.

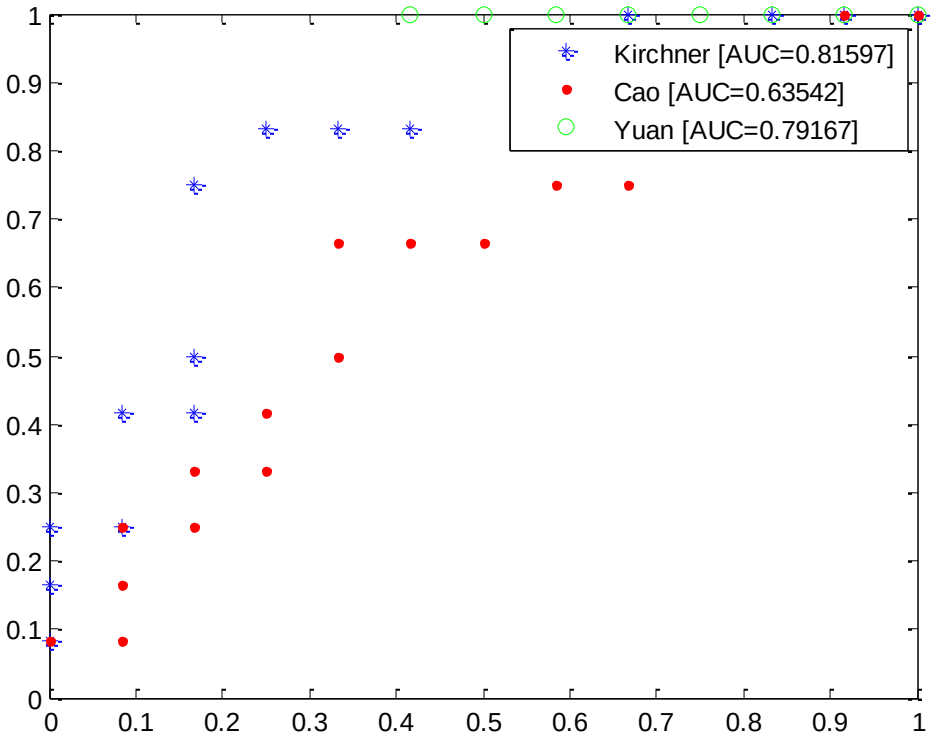


Figure 5.9. ROC characteristics of image database taken using Nelder Mead optimization

5.2 Advantages of using particle swarm optimization over existing technique

PSO have no overlapping and mutation calculation. The search can be carried out by the speed of the particle. During the development of several generations, only the most optimist particle can transmit information onto the other particles, and the speed of the researching is very fast.

The calculation in PSO is very simple. Compared with the other developing calculations, it occupies the bigger optimization ability and it can be completed easily.

PSO is a stochastic approach which works iteratively to reach for a best solution. The Nelder Mead optimization algorithm used can only search for a local minima as it only move towards the minimum value starting from a point, whereas PSO search for the global minima as it can move in either direction to search for the minimum. Inertia weight factor used in PSO plays an important role in finding out the global minima. Considering the value of inertia weight w in the range [0.8,1.2] finds a global minima as higher values move the swarm in a larger area hence, more possibility of reaching a global minimum. Considering lower values of w fine tune the global minimum solution and can refine the results.

Conclusion and Future work

We propose a method for concealing traces of median filtering in uncompressed digital images. The method exploits knowledge of features used by existing techniques for median filtering detection [21] [25] [26]: an optimization problem is devised that, for a given image, yields a linear filter that allows to remove footprints while keeping the fidelity between the processed image and the counter processed one as high as possible. The proposed method has shown a considerable improvement over existing technique. PSO as an iterative method and a meta-heuristic approach finds a global minima which lacks in existing technique.

Future work may consider the use of composed objective functions for the optimization problem, so to obtain a filter that simultaneously conceal traces searched by different detectors. It may also consider using a mixture of heuristic and numerical methods for the optimization problem.

References

[1] Rainer Bohme. *Advanced Statistical Steganalysis*. Berlin, Heidelberg: Springer-Verlag, 2010.

[2] A road map for digital forensic research. Report from the First Digital Forensic Research Workshop (DFRWS), 7–8 August 2001. *DFRWS Technical Report.* Utica, New York: Digital Forensic Research Workshop (DFRWS 2001).

[3] Casey, E (ed.) 2002b. *Handbook of Computer Crime Investigation. Forensic tools and technology.* London: Academic Press.

[4] John, J L 2008. Adapting existing technologies for digitally archiving personal lives. Digital forensics, ancestral computing, and evolutionary perspectives and tools. *iPRES 2008 Conference. The Fifth International Conference on Preservation of Digital Objects.* The British Library, London.

[5] Kruse, W G, II and Heiser, J G 2002. *Computer Forensics. Incident response essentials.* Boston, Addison-Wesley.

[6] Sammes, T and Jenkinson, B 2007. *Forensic computing. A practitioner's guide*, London, Springer-Verlag. 2nd edn.

[7] Gary L. Friedman. "The Trustworthy Digital Camera: Restoring Credibility to the Photographic Image". In: *IEEE Transactions on Consumer Electronics* 39.4 (Nov. 1993), pp. 905–910.

[8] Ramarathnam Venkatesan, S.-M. Koon, Mariusz H. Jakubowski, and Pierre Moulin. "Robust Image Hashing". In: *IEEE International Conference on Image Processing, ICIP2000* (Vancouver, BC, Sept. 10–13, 2000). Vol. 3. 2000, pp. 664–666.

[9] Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, 2008.

[10] Jan Lukáš, Jessica Fridrich, and Miroslav Goljan. “Digital Camera Identification from Sensor Pattern Noise”. In: *IEEE Transactions on Information Forensics and Security* 1.2 (June 2006), pp. 205–214.

[11] Ashwin Swaminathan, Min Wu, and K. J. Ray Liu. “Nonintrusive Component Forensics of Visual Sensors Using Output Images”. In: *IEEE Transactions on Information Forensics and Security* 2.1 (Mar. 2007), pp. 91–106.

[12] Ashwin Swaminathan, Min Wu, and K. J. Ray Liu. “Optimization of Input Pattern for Semi Non-Intrusive Component Forensics of Digital Cameras”. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2007* (Honolulu, HI, Apr. 15–20, 2007). 2007, pp. II–225–228.

[13] Tian-Tsong Ng, Shih-Fu Chang, Ching-Yung Lin, and Qibin Sun. “Passive-blind Image Forensics”. In: *Multimedia Security Technologies for Digital Rights*. Ed. by Wenjun Zeng, Heather Yu, Ching-Yung Lin. Academic Press, 2006, chap. 15, pp. 383–412.

[14] Kuncik N & Harbison A. A Brief Introduction to Counter-Forensics, Digital Forensics Magazine, Issue 1, September 2009

[15] R. Boyle and R. Thomas *Computer Vision: A First Course*, Blackwell Scientific Publications, 1988, pp 32 - 34.

[16] Kennedy, J. and Eberhart, R. C. (1995). Particle swarm optimization. *Proc. IEEE Int'l. Conf. on Neural Networks, IV*, 1942–1948. Piscataway, NJ: IEEE Service Center.

[17] J.H. Holland (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, Michigan; re-issued by MIT Press (1992).

[18] C.R. Reeves (1997) Genetic algorithms for the Operations Researcher. INFORMS Journal on Computing, 9, 231–250.

[19] M. Mitchell (1996) An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA.

[20] M. Dorigo et L.M. Gambardella, *Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem*, IEEE Transactions on Evolutionary Computation, volume 1, numéro 1, pages 53-66, 1997.

[21] M. Kirchner and J. Fridrich, "On detection of median filtering in digital images," in SPIE Conference Series, Feb 2010, vol. 7541 of SPIE Conference Series

[22] Farid, H.: Exposing Digital Forgeries in Scientific Images. In: Proc. of ACM MMSec. pp. 29–36 (2006)

[23] Matthias Kirchner and Rainer Bohme, "Tamper hiding: Defeating image forensics.," in Information Hiding'07, Jun. 2007, pp. 326–341.

[24] Popescu A.C., Farid, H.: Exposing Digital Forgeries by Detecting Traces of Resampling. IEEE Trans. on Signal Processing 53, 758–767 (2005)

[25] G. Cao, Y. Zhao, R. Ni, L. Yu, and H. Tian, "Forensic detection of median filtering in digital images," in Multimedia and Expo (ICME), 2010, Jul. 2010, pp. 89 –94.

[26] H. Yuan, "Blind forensics of median filtering in digital images," Information Forensics and Security, IEEE Transactions on, vol. 6, no. 4, pp. 1335 –1345, Dec. 2011.

[27] M. Fontani, M. Barni "Hiding Traces of Median Filtering in Digital Images", EURASIP European Signal Processing Conference (EUSIPCO'12), Bucarest, Romania

[28] J. A. Nelder and R. Mead, "A simplex method for function minimization," Computer Journal, vol. 7, pp. 308–313, 1965.

[29] G. Schaefer and M. Stich, "UCID: an uncompressed color image database," in SPIE Conference Series, M. M. Yeung, R. W. Lienhart, & C.-S. Li, Ed., Dec 2003, vol. 5307, pp. 472–480.