Detection of Tampering in Digital Images Using Feature Based Hash Generation

by

Vinod Kumar Mall
200921001

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

October, 2014

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Doctor of Philosophy in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

VINOD KUMAR MALL

Certificate

This is to certify that the thesis work entitled “Detection of Tampering in Digital Images Using Feature Based Hash Generation" has been carried out by VINOD KUMAR MALL (200921001) for the degree of Doctor of Philosophy in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under our supervision.

ANIL KUMAR ROY & SUMAN KUMAR MITRA
Thesis Supervisors

Acknowledgments

More than five years have gone by since I joined Ph.D. program at Dhirubhai Ambani Institute of Information and Communication Technology. The process was intense but enjoyable. I came across some wonderful personalities during the work and would like to express my feelings for my association with them.

First, I would like to express my most sincere gratitude to my guides Prof. Suman K. Mitra and Dr. Anil K. Roy. Both spared their valuable time for motivating and guiding me to choose the area of research and providing direction whenever it was needed. I am particularly thankful to both of them for accommodating my professional commitment to Gujarat Police and stretching themselves to all possible limits to help me. Both the guides were friends and philosophers with whom I went through the process of academic evolution and in the process learned lot many things in addition to academic research work.

I am thankful to Prof. V. P. Sinha, my mentor in initial Ph.D. days, who helped me in orienting myself to research work by discussing various issues giving my broader framework to research. Listening to him was a great philosophical experience and meditative delight.

I would also like to thank Prof. Sanjay Srivastava and Prof. Manik Lal Das for their inputs and suggestions during the process of synopsis submission.

I also take cognizance of contribution in form of discussions with Shri Kedar Bhatt, Shri Shivanshu Shuka and Shri Vamshi Chenna (all bright students of B.Tech.)

with whom I had frequent discussions about different tools used in image processing area.

Students of M.Tech. with whom I did my course work durung $2010 - 12$, were always friendly and helpful despite huge age gap. They made me feel one of them and I express my feeling of gratitude towards them.

I am thankful to Shri Prabhunath Sharma of M.Des. Studio for teaching me how to use PhotoShop. This learning was of immense help in building our own database of tampered images and in carrying out the experiments.

This acknowledgment would not be complete unless I thank my family members Anuradha Mall (wife) and Kshitij and Rishabh (sons) who allowed me to devote time to my research work. I know they might have suffered quietly when I was juggling with bunch of text books and heaps of reference papers.

I am also aware of the fact that I failed to give desired attention and care to many of my friends due to my PhD work along with my job during this period. I apologize and thank all of them who shared this period with smiles and encouraged me throughout.

I owe special acknowledgment to Profs. Manish K. Gupta and Aditya Tatu who very playfully taught me the tools and tricks of LaTeX during writing of my thesis.

DA-IICT canteen was an integral part of life during this phase and I thank its entire staff for lovely tea and coffee they provided whenever I needed a break from work.

Vinod Kumar Mall
29 October 2014

Contents

Abstract

Recent years have witnessed an exponential growth in the use of digital images due to development of high quality digital cameras and multimedia technology. Easy availability of user-friendly image editing software has made modification of images a popular child's play. In this environment, the integrity of an image cannot be taken for granted. Malicious tampering has serious ramification in legal documents, copyright issues, photojournalism, celebrities' lifestyles, fashion statements, beauty and fitness products, entertainment sector, medical science, biometric images and forensic cases. The proposed work is based on features based hash generation of a digital image and thereby detection and localization of image tampering that is done with malicious intention, no matter howsoever small the tampering is. The hashing algorithm generates a short binary string by extracting the feature vectors of the image and mapping them into robust hash values. This hash mapping meets the two requirements of being sensitive to tampering in the image and being robust against content preserving manipulations. The first part of the work uses correlation coefficient to generate a hash representation of the image which is then utilized for tampering detection. This method is extended later to use properties of Singular Value Decomposition (SVD) to derive the hash values. SVD enables us to generate key based hash values. We show that this hash can be used in secured transmission of image information on the public network. We emphasize that the malicious tampering is generally done on the structural part of the image. Therefore, in the third part of our study, we have used Canny Edge Detector to extract the features of the image by detecting the edges present in it. Having the edges identified, tampering detection and localization are carried out. This method of tampering detection and localization has been found to be promising. The last part of the thesis discusses Comprehensive Image Index which is able to detect multiple types of tampering, viz., brightness, contrast, and structural, simultaneously. We observe that a structural tampering is always followed by brightness and contrast changes in the image. We also establish that even brightness or contrast or both change can be seen as a malicious tampering if exceeds a threshold value. The sensitivity of the technique and its robustness have been discussed quantitatively for each method.

List of Tables

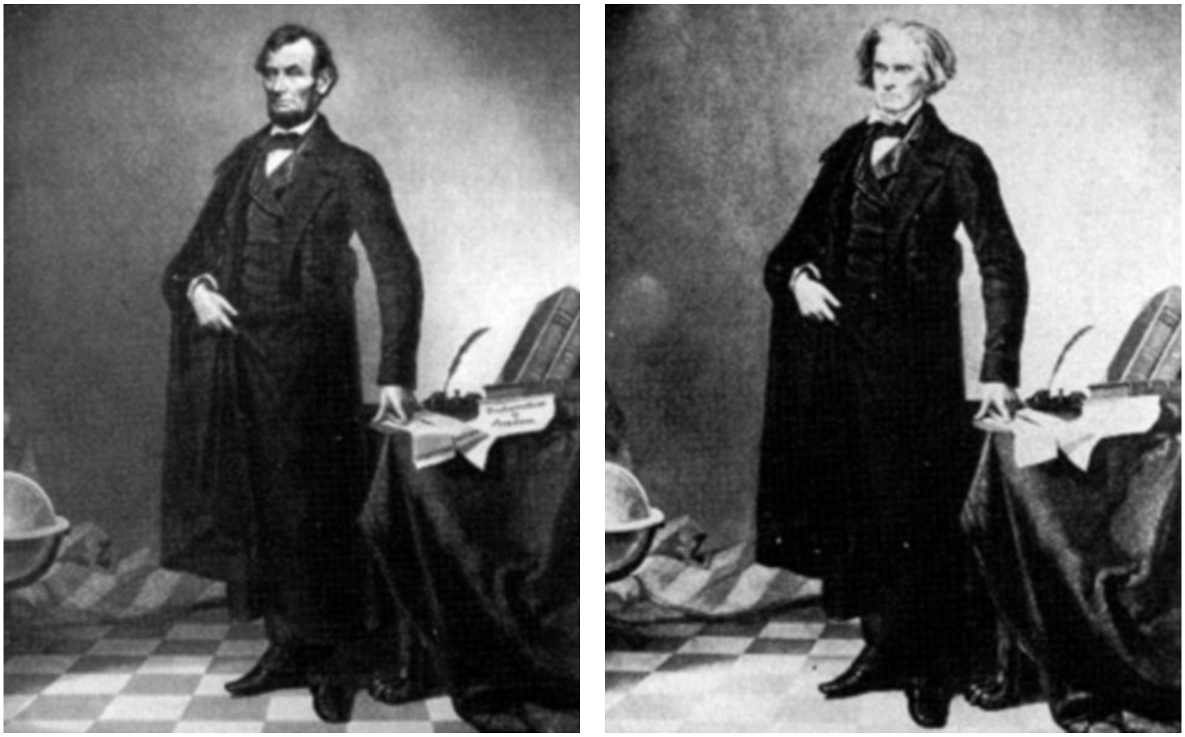
List of Figures

Chapter 1

Introduction

Capturing the moment is basic instinct of human being. It was in early 19^{th} century that the first photograph came into existence [1]. For the first time in 1827, Niepce produced a picture successfully using a material that hardened on exposure to light. The work on chemical substance and negative picture plate continued over the years. Hershel was interested in capturing and retaining images and could fix pictures using hyposulphite of soda in 1839. History of tampering of an image is almost as old as the history of image itself. It is widely known that iconic picture of Abraham Lincoln, shown in fig. (1.1), was a combination of Lincoln's head and southern politician John Calhoun's body [2]. It is also well known that Russian leader Stalin used to delete the pictures of persons from a photograph, whom he did not like. Some of these historic fake photos are shown in figs. (1.2) and (1.3). There have been various instances where scandalous stories were created around known politicians and socially important people.

Digital technology has provided easy tools to capture images or photographs, alter the contents of an image, and manage those images or photographs on the unlimited cyberspace. Over 250 million photos are uploaded daily on Facebook worldwide [3]. Instagram (http://instagram.com/#), the social photo app, that was launched in Oct 2010, has 16 billion photos shared with a rate of 45 million photos get uploaded per day [4]. Tumblr (https://www.tumblr.com/), Flickr (http://www.flickr.com/), Picasa (http://picasa.google.com/), Photobucket (http://www.photobucket.com/), Imgur (http://www.imgur.com/), DeviantART (http://www.deviantart.com/), etc. are some of the most popular social net-



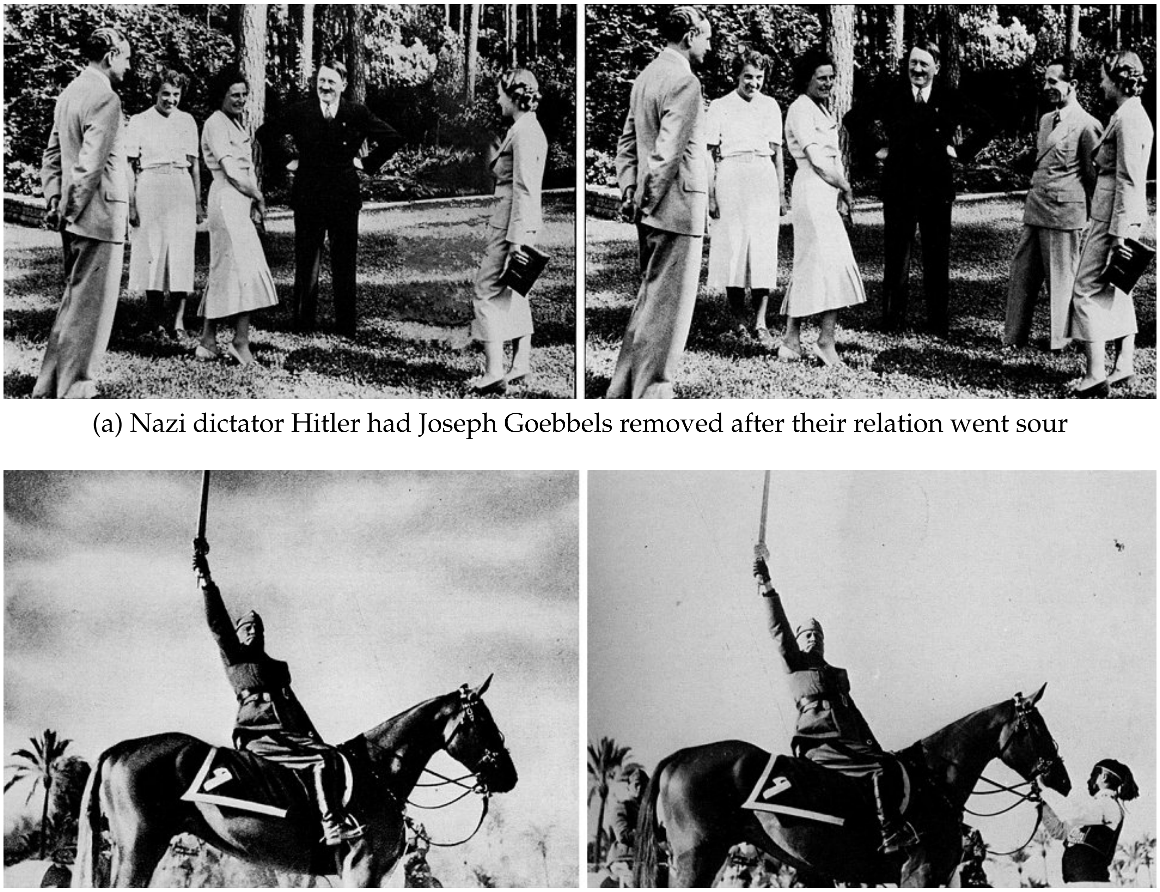
(a) Iconic picture of Abraham Lincoln

(b) Picture of politician John Calhoun

Figure 1.1: Once both pictures are given, one can easily make out that Abraham Lincoln's head is put on top of the body of John Calhoun. This is one of the earliest examples of generating fake photos.

working sites for managing and sharing photos.

Once we have these billions of photos available in digital form on the Internet as well as on standalone computers, it has become a child's play to tweak and fiddle with these images. Easily available software such as Adobe Photoshop and Microsoft Paint has made it very simple to tamper with an image. Also there are a plenty of mobile Apps available [5, 6], such as Adobe Photoshop Touch, Camera+, CameraBag, CloneCamera, InstaCollage Pro, Instagram, iPhoto, Photogene, Toon Camera, ArtStudio, Foto Brush, PicPlz, Tiltshift, PhotoFX, FX Photo Studio, Adobe Photoshop Express, ACDSee Pro 6, ACDSee Photo Editor, Corel PaintShop Pro X5, DxO Optics Pro 8, GIMP 2.8, Paint.net, to add some photo effects and do adjustments like blurring, contrast enhancement, brightness alteration, colour variation and the like. Sometimes, alteration in an image is done in order to improve the visual quality of the image through operations such as

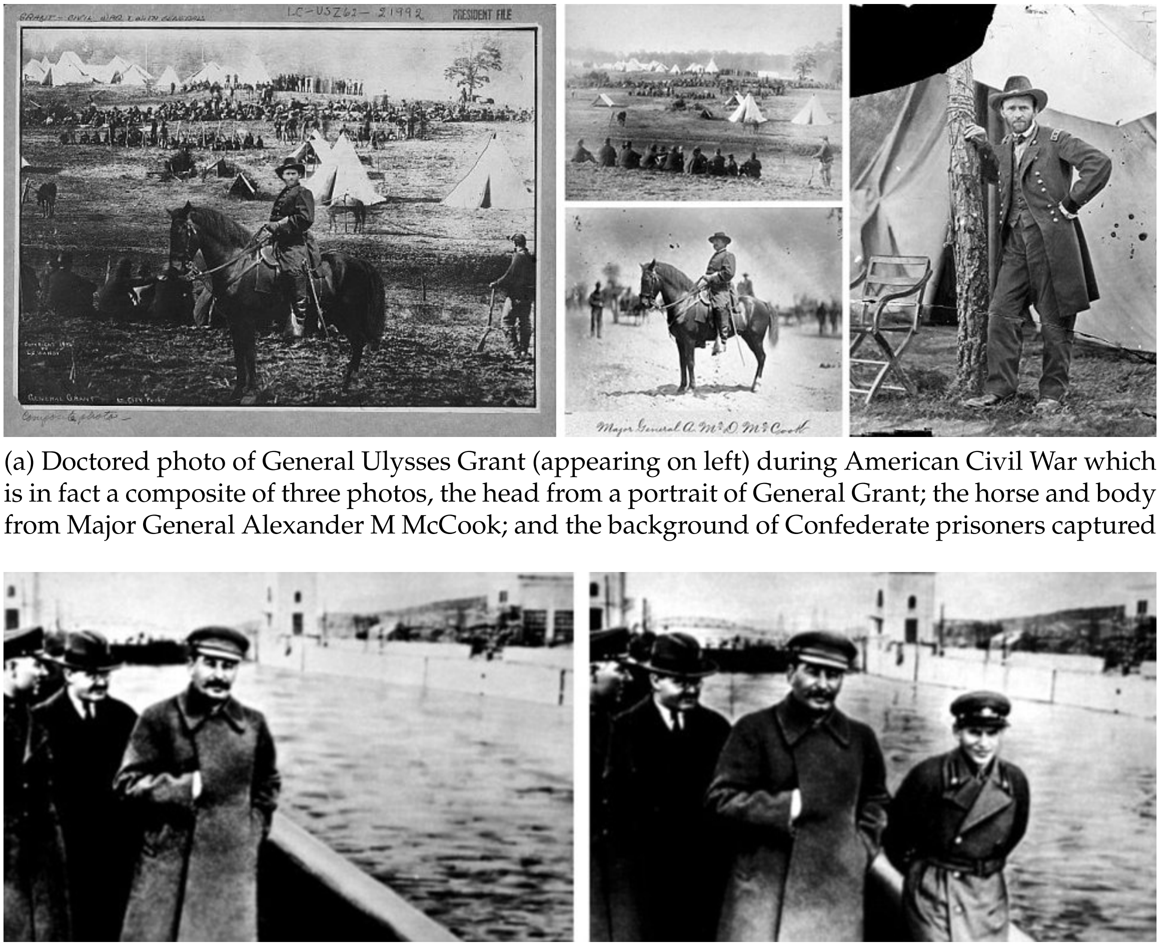


(a) Nazi dictator Hitler had Joseph Goebbels removed after their relation went sour

(b) Italian dictator Benito Mussolini removed his horse handler from the photo appearing on right to cast his more heroic image

Figure 1.2: In the above photos the historic characters Hitler and Mussolini are seen in photos which are tampered with.

contrast change, brightness change, stretching, low pass/high pass filtering etc. However, alteration in an image can also be done with malicious intention which may have bearing on legal cases, evidence in court, police investigation, defamation and fraud cases, copyright issues, politics [7], photojournalism [8, 9], celebrities' lifestyles [10], fashion statements, beauty and fitness products, entertainment sector, advertisements [11], crime against teens [12], medical science [13, 14, 15], biometric images and forensic cases. In the beauty product market a very interesting computer game has been developed [16, 17, 18, 19] to see and feel, in virtual reality, who has got the best makeover. In all such situations, verification of integrity of an image becomes a challenging task [20, 21, 22]. Efforts to design and develop more and more subtle and accurate techniques and tools to identify the original and to eliminate the duplicates or forged ones are increasing with



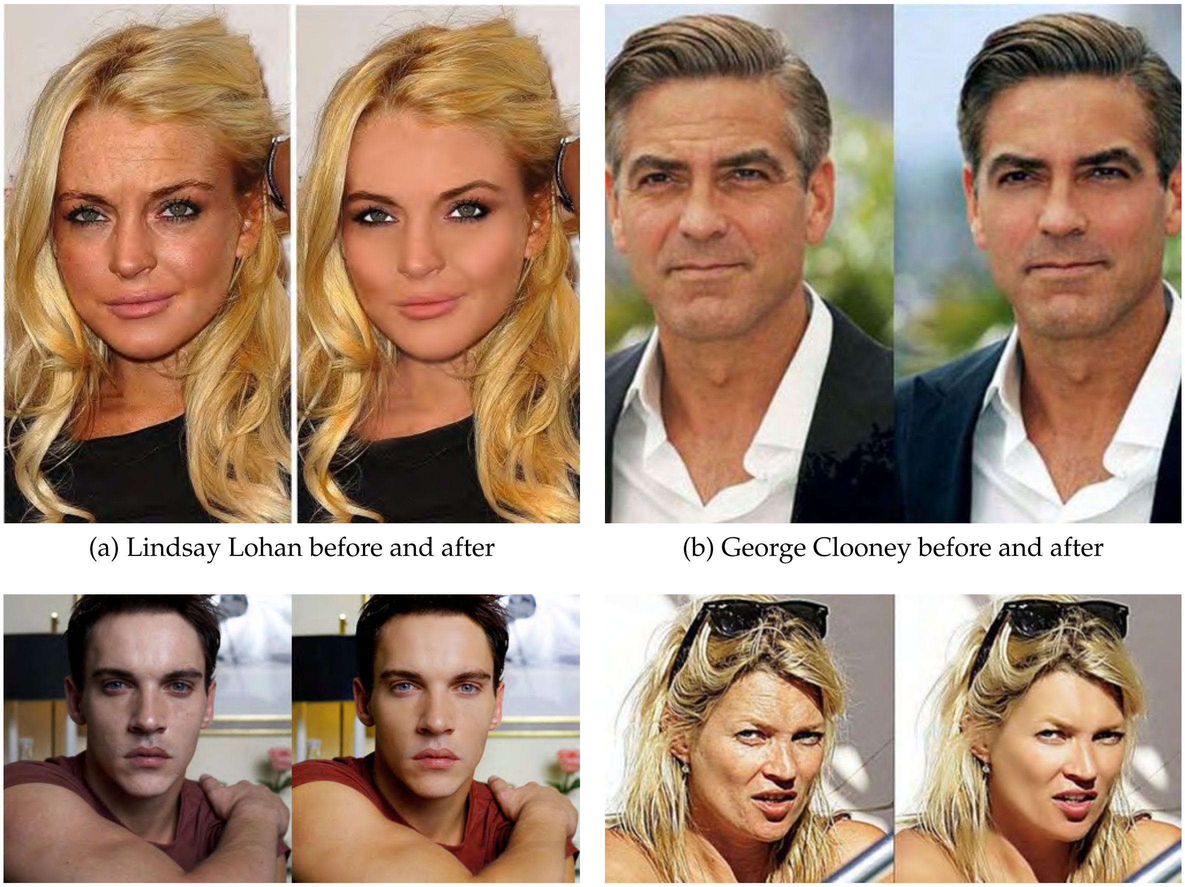
(a) Doctored photo of General Ulysses Grant (appearing on left) during American Civil War which is in fact a composite of three photos, the head from a portrait of General Grant; the horse and body from Major General Alexander M McCook; and the background of Confederate prisoners captured

(b) Russian dictator Josef Stalin was famous for doctoring his photographs. In this photo a commissar was removed from the original photograph (appearing on right for reason best known to Stalin

Figure 1.3: In the above photos the iconic characters Ulysses and Stalin are seen in doctored photos.

time [23]. Examples of photoshopping in photojournalism are in plenty. Some of them are reproduced in the following fig. (1.5).

Of late, large number of researchers have been working in the area of digital image forensics which focuses on detection of tampering in digital images [24, 25, 26, 27, 28, 29, 30, 31]. There are two possible categories when we talk about digital image tampering detection. In first category, the original image is not available. The techniques employed to detect tampering in absence of the original image are called *blind techniques* [32]. These techniques generally identify the statistical correlations generated following a tampering operation such as rotation, stretch-



(a) Lindsay Lohan before and after

(b) George Clooney before and after

(c) Jonathan Meyers before and after

(d) Kate Moss before and after

Figure 1.4: Celebrities are one of the main subjects of fake photos. They give particular attention on how should they appear in public no matter how they look like in private.

ing, JPEG compression etc. [33, 34, 35, 36]. The nature of statistical correlations depends upon a specific type of tampering operation [37].

The second category techniques, where original digital image or its mathematical representation is available for the purpose of detection are called non-blind detection techniques. Non-blind techniques are further divided into two classes. These are (a) watermarking technique [38, 39, 40, 41, 42] and (b) image feature based hash generation technique [43, 44, 45, 46, 47]. In watermarking technique, a code called watermark is embedded in the image by the source [48, 49, 50, 51, 52]. The tampering in the image is detected by the receiver at a later stage by finding the changes, if any, in the watermark [53, 54, 55, 56, 57, 58]. The disadvantages of watermarking technique are:

1. Pre-possession of the image is required.
2. The size of the watermark is limited by a size of the image in which it is being embedded.
3. Perturbation caused by the watermark may not be ignorable in all situations [59, 60].

To overcome the shortcoming of watermarking techniques, hash generation method based on features of the image is used for integrity verification. This technique creates a short binary string which is an efficient representation of the image. The hash representation should have maximum possible information about the image. At the same time, it should not be too long to adversely affect the processing speed and memory requirement.

As mentioned earlier, there are large numbers of image manipulation techniques. No single technique is capable of detecting all of them. We will discuss a few such techniques here. There have been cases where digital images from a camera were replaced by computer generated pictures. Detection of such tampering is based on a fact that every camera is associated with a unique noise pattern [61]. This pattern is identified during the detection process. However, knowledge of noise pattern of digital camera is prerequisite for detection which becomes a limitation in many situations.

It has recently been reported that digital image processing tool such as Photoshop has lured many biologists to do tweaking with their data. It has generated a loud debate for fixing the guidelines for deciding what is an acceptable image quality enhancement and what is a scientific misconduct [13]. Tampering in the contrast of the image is carried out using gamma correction given by $P = CI^{\gamma}$. To detect this non-linear luminous manipulation, Inverse Gamma Correction method has been used. The contrast enhancement introduces higher order correlations in frequency domain. These correlations are detected using polyspectral analysis tools [39]. Re-sampling processes which cause scaling and rotating operations on an image or part of it, are responsible for generating specific statistical correlations. The re-sampling artifacts can therefore be detected by identifying these correlations. However, similar periodic patterns can be generated by different re-sampling processes which become a serious limitation in uniquely determining a type of re-sampling [39].

It is observed that image tampering operations leave behind detectable fingerprints which are often unique to the performed operation. Tampering operations are basically pixel value mappings which result into statistical traces which are mapping's intrinsic fingerprints. Various tampering operations such as local and global contrast enhancement, histogram equalization, noise addition in previously JPEG compress images etc. have been detected using this concept [33, 43]. In some cases, part of an image is removed and replaced by part of another image and resaved. Most of images in the digital environment are already JPEG compressed. Therefore while resaving the image after tampering operation, the tampered area will exhibit single compression while the remaining part of the image will show double compression. In such cases, tampering detection is done by estimating the primary quantization matrix for initial JPEG compression [62, 63]. The doubly compressed area will show changes in the elements of quantization matrix.

In another work [38], fluctuations in localized estimates of SNR is used to detect the artifacts which are created due to image tampering. However, a smart attacker adds global noise due to which SNR variations get subdued and can not be detected. In order to deal with this problem, an innovative method was developed which introduces a predefined mapping into the image. This mapping is associated with a specific fingerprints and it changes whenever some alteration is done in the image [33].

Going back to the motivation behind this research work, we reiterate that human beings are trained in thinking in images. Images have everlasting impression on our memory. With the advent of powerful tools for digital imagery, we never know how the Dr. Jekyll or Mr. Hyde in somebody would use that tool to alter the image. That alteration may have malfide intention which may or may not be detectable by human eye. Therefore we thought, parallel to developments of these digital tools to photoshop an image, we must be ready with some tool that would tell us about the fidelity or authenticity of the image. This tool must be plug-and-play type of tool which can be used by any amateur even.

Now one may ask that if some tampering is not noticeable by human eye, how come this be so important that its authenticity can be challenged. We have given several examples earlier in this chapter in this regard. The major area under severe threat could be lifestyle related, because everybody wants to look beautiful, handsome, presentable, adorable to catch others' attention. The statistics of Facebook is glaringly self-evident where people wait passionately for hours for their "friends" to "like" their just uploaded images. And this is just what we see for the images that are in the public domain. We suspect that around 90% of worldwide data don't reside in public domain at all. Malpractices there might be much more subtle. But we do believe that cases like fig. (1.4) need to be verified before deciding for any cosmetic product if they are brand-ambassadors of any such beauty products. Public faces in private places are different and in public places are presented differently. So how to authenticate? This question perplexed us to look into this problem where technology had deep rooted social impact.

Another example could be to make a judgmental view on police atrocities or brutality which is very difficult to establish otherwise. Still there is a common public opinion worldwide that police investigations may many times violate human rights. The suspects undergo ruthless brutality. The mug-shots presented by police in the courts may be made up photographs of the undertrials. The reality could be different. So how to authenticate?

A new area of forgery has recently emerged and that is medical images. These images could be of a patient who is fighting for a health insurance claim or interestingly it may be of medical research also. The articles published in Nature [13, 14, 15] are self-evident for possibility of such conspiracies.

The latest fad of technological boon is the birth of an immersive technology called Augmented Reality. In this a person may take a real-life image or video, tags it with explanatory data overlaid on that. The lenses of the device converts it into an expansive $3-D$ panorama.This would help you better understand what's going on, or who the people in the scene are, or how to get to where you want to go. So far so good, but who knows how this technology could be misused in future if some tampered image would be used as the key element of this $3-D$ panorama. The complete story will change.

This real cum futuristic trend of natural human instinct of playing with images motivated us to find out in simplest way how to detect tampering in any digital image, how to identify the exact location where that tampering has been done with mischievous goal.

1.1 Organization of the Thesis

For doing so, we start with the mathematical representation of an image that is to find its hash representation and it should be able to help in tampering detection. Our research work uses image feature based hash generation technique which has distinct advantage of robustness against content preserving manipulations [64, 65, 66]. Cryptographic hash function is very powerful tool to protect the integrity of data. These functions are key dependent and very sensitive. Even if a single bit in the input data changes, the hash value changes significantly. Though cryptographic hash functions serve very well to protect the integrity of text messages, they are not very suitable to safeguard the integrity of image as authentication of image input data is not straightforward. Cryptographic hash functions do not meet the requirement of robustness against certain content preserving manipulations. Due to this reason, cryptographic hashes are not suitable for image integrity check. Instead, feature based image hashes are used for this purpose. To meet the twin requirements of detection sensitivity and robustness, in the ***chapter 2*** we propose a general algorithm for feature based hash generation and related techniques to provide efficient tampering detection and localization (TDL). Following this a number of algorithms will be discussed which have their own distinct features.

Besides the chapter of 'General algorithm using feature based hash generation', the presentation of our research work will be divided into following chapters in this thesis:

1. Correlation coefficient based hash generation techniques
2. Singular value decomposition based hash generation for detection of structural tampering
3. Hash technique using Canny Edge Detector for structural tampering detection
4. Comprehensive Image Index for detection of multiple tampering using 3-tupled hash function

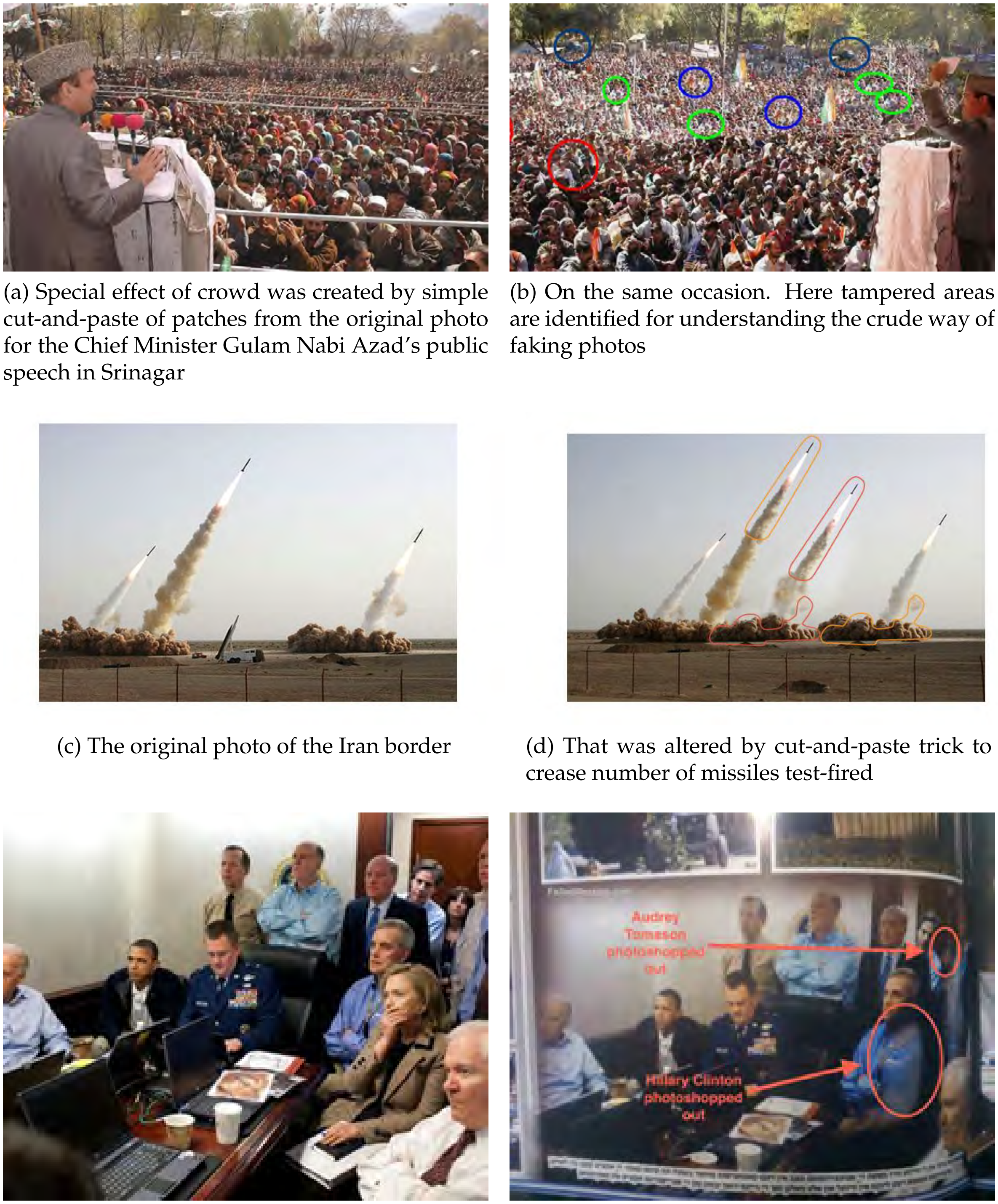
In our work, a number of feature based techniques have been attempted and their relative merits have been studied through experiments. First method uses correlation coefficient based hash generation technique which is explained in the ***chapter 3***. This feature based hash function can be obtained using correlation coefficient of two images in consideration. One of these images we call the original image and the other we call the suspect image. We will show that it completely represents an image, is compact in size, robust against content preserving manipulations such as brightness change, alteration in contrast etc., and is sensitive towards even a minutest of structural change. Using this method, tampering detection and localization (TDL) has been achieved efficiently. Accuracy of localization of tampered area has been shown to improve by changing the sampling block size. An index called Similarity Value is introduced to be used to qualify the amount of tampering to measure amount of tampering in the image.

In the following ***chapter 4*** another method using Singular Value Decomposition (SVD) in conjunction with correlation coefficient will be dealt with. We will show that Singular Value Decomposition (SVD) offers a good detection tool for identification of structural tampering in digital images [67, 68]. SVD gives a strong representation of structural features as well as luminous values of the image. The orthogonal matrices in singular value decomposition contain the structural features of the image while the singular value matrix is essentially the luminance component [59]. Breaking up of image matrix into orthogonal and singular value matrices, which represent structural and luminance component respectively, enables researchers to use them in various ways for the purpose of tampering detection. In our work also, this technique will be used for key dependent structural tampering detection. Second method deals with key-enabled hash generation using Singular Value Decomposition technique. This method provides increased security to the hash. It will be shown that it provides higher level of security to the hash. We have also calculated the collision probability of hash functions and proved it to be extremely small which is required for a good function.

The structural information in an image is contained in the edges existing in it. Any alteration, removal or insertion of an edge amounts to structural tampering in the image. This can be affected through bringing part of a different image into the original one or by removing the part of original image or sometimes removing a part followed by inserting segment of a different image. The utmost required property of feature based hash generation is its robustness against Content Preserving Manipulations (CPM). Therefore, in the next ***chapter 5***, we have used Canny Edge Detector to extract features which are basically edges in any image. The property of this detector is double thresholding along with local maxima suppression and hysteresis tracking which has been efficiently used to reject CPMs while picking up features needed to generate the hash. As expected, sensitivity and robustness is better than method using correlation coefficient as hash value. This method defines a new index called Average Edge Index for the first time in this area of research. The properties of this detector provide a high degree of sensitivity of detection along with extreme robustness against content preserving manipulations.

During the research work it was realized that to understand the type of tampering in more complete manner, a multiple parameter index is required which captures different types of tampering operations in an image in a comprehensive manner. In ***chapter 6***, we will be defining this new index called Comprehensive Image Index which will be useful to detect multiple tampering carried out in an image simultaneously. It uses three parameters namely structural index, brightness index and contrast index simultaneously and helps the forensic expert to conclude about the motive of the attacker by classifying the type of tampering operation or a combination of them. This has been attempted by us for the first time.

At the end, we will sum up with our design of a plug-and-play type of TDL software. We will call it Image Tampering App which is included in the ***chapter 7***. We will show that this software could be a right plug-in for Adobe Photoshop, whereby one immediately gets the result if or not the suspect image is tampered with. It also gives the measure of tampering by displaying the similarity value (defined later) of the suspect image with respect to the original image.


Audrey Tomason photoshopped out
Hillary Clinton photoshopped out

(a) Special effect of crowd was created by simple cut-and-paste of patches from the original photo for the Chief Minister Gulam Nabi Azad's public speech in Srinagar

(b) On the same occasion. Here tampered areas are identified for understanding the crude way of faking photos

(c) The original photo of the Iran border

(d) That was altered by cut-and-paste trick to increase number of missiles test-fired

(e) This is the original Bin Laden situation room photo from the White House in which two ladies are seen including Hillary Clinton, the Minister of State of USA

(f) The same photo was altered in a fundamentalist national daily Hasidic Newspaper who believed that showing women involved in war scene was inappropriate

Figure 1.5: Phtojournalism is another profession where rampant cases of image tampering are observed.

Chapter 2

Principles of Proposed Technique of Generation of Feature Based Hash Function

Lot of work has been done by various researchers using hash generation techniques [48, 69, 70, 71, 72, 73, 74]. Here, we will give a brief summary of work done in this area. One group of researchers used Radon Transform and Principal Component Analysis to extract image features and generate digital signature which is robust against geometrical transformations, viz., rotation and scaling, and image processing attacks [75]. Special Image Digest function has been used [76] which returns same bit-string for images with some amount of CPM but derived from same original image. Another group generated a reference image from a monitor image and extracted feature information to compute hash values [77]. Method of generating reference image and permissible error decides accuracy and robustness. In another work, 2-directional cosine transform was used to extract feature vectors. An intermediate hash was created from sign bit of the vector which was incorporated into a security mechanism to yield final hash [78]. Most of these techniques are good at detecting existence of tampering in an image but not in attempting at achieving higher accuracy in the localization of tampering area [79, 80]. Issue of robustness of detection method against Content Preserving Manipulation (CPM) has also been dealt widely by different researchers [81]. However, critical value of CPM against which a particular method is robust, is seldom found out. It is observed that content preserving manipulations are normally global in nature while tampering is limited to a smaller area in the image. Therefore, if CPM is summed up for the whole image, it may increase the amount of malicious tampering which is limited to small area. In such case, CPM might be mistaken for a tampering operation. Therefore it becomes necessary to quantify the value of extent of CPM below which a technique is robust.

Image feature based hash generation is popular technique for tampering detection. Image hash based on Fourier transform features along with controlled randomization was found to be resilient to CPMs such as moderate geometric and filtering distortions [82]. Generated hash provides excellent security also. Another algorithm [83] used randomized signal processing strategies for non-reversible compression of images into random binary strings and is shown to be robust against alterations due to compression, geometric distortions and other attacks. To enhance the security, the pixels in spatial domain are randomly modulated through a secret key before applying wavelet transform. Process is termed by researches as randomized pixel modulation (RPM). Approximate Image Message Authentication Code (AIMAC) was also reported [84] in which mean of image block was extracted to generate the hash. This method has limitation that a block could be drastically changed without changing the mean. This method is therefore not so good from collision probability point of view. Another reported work [85] used Scale Interaction Model in wavelet domain to extract the image features to generate hash for checking the integrity of suspected image. Wrap around effect of wavelet transform puts a limitation on accurate localization of tampering in the image. Invariance property of a radon transform was used [86] for hash generation followed by verifying the integrity of suspect image. Nonnegative Matrix Factorization [NMF] was used for hashing [72] in which randomized dimensionality reduction was used. NMF has also been used [87] for analysis of security of perceptual image hash. The technique is claimed to be computationally simple and minimizes error probabilities. Robustness of polar harmonic transforms and feature selection based hash function of an image is also discussed and found to be distortion-resistent [88]. In another method, image after being transformed from RBG to YCbCr scheme, is mapped to unit circle through conformal mapping followed by calculation of Zernike moments and permuting the intermediate hash sequence to obtain final hash [89, 90]. Hashing based on dimensionality reduction in compressive sensing technique was used to detect image tampering [91].

2.1 General algorithm for feature based hash generation

Hash generation technique creates a short binary string called hash vector which contains suitably extracted features of the image. First, the image features are extracted using a suitable method and then it is converted into a mathematical representation using a hash function. The hash function should generate exactly one hash value corresponding to each image. Required properties of hash function can be mathematically expressed as below:

1. For any image I, it should be easy to compute its hash value $h(I)$ using hash function h at sufficiently fast speed.

2. The reverse computation should be extremely difficult. It should be almost impossible to compute I from its hash value $h(I)$.

3. One digital image I should result into only one hash value $h(I)$, i.e., function h should be a one to one mapping.

4. For every $h(I)$ there should be exactly one I in the image domain. Equivalently, if $h(I) = h(I')$ then $I = I'$. This property can also be expressed by saying that the collision probability of the hash function should be extremely low [92].

The detection technique using feature based hash generation should have following properties:

1. **Sensitivity**: The algorithm should be sensitive meaning that it should be able to detect even very minute tampering.

2. **Robustness**: The algorithm should be able to ignore content preserving manipulations such as JPEG compression, contrast enhancement, blurring and brightness change etc. carried out without any malicious intention.

3. **Low Collision Probability**: There should be a strict one to one correspondence between an image and its hash representation. It has been mathematically explained above while describing the required properties of hash function.

4. **Compact Size**: Hash used should have maximum content information of its image but should not be too long to adversely affect memory and processing time considerations.

5. **Irreversibility**: It should be impossible to find out the image from its hash value.

Image Hash is an image feature based digital signature. Security of the hash can be enhanced by using a secret key in the process of hash generation. Hash can be sent to the receiver along with the image or separately. There are two contradictory requirements for a good hash function. The size of the hash should be big enough to have maximum possible information about the image. At the same time, it should not be too long to handle from system memory and processing speed considerations. Therefore, a trade-off between the two requirements is to be ensured. Quality of hash function is evaluated around the parameters listed above [93].

Tampering detection method consists of following three steps:

1. Feature extraction

2. Hash generation using image features. Image feature vector **F** is N dimensional vector in real number space.

3. Verification through comparison of hash output corresponding to original and suspect images. Image feature vector $F \in \Re^N$ is derived from the im-

age using hash function, $f : F \rightarrow h$. Hash values h are mapped to binary sequence $\{0,1\}^N$. $\{0,1\}^N$ represents bit sequence of size N.

These three steps are shown in fig. (2.1).

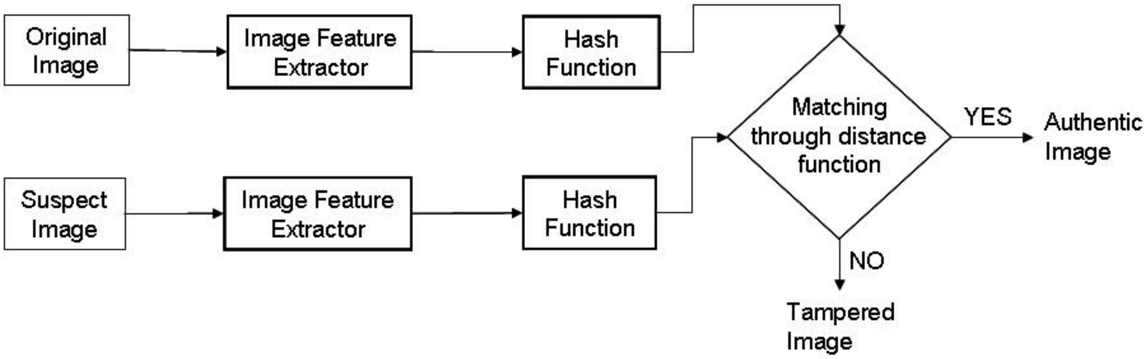


Figure 2.1: Simple flowchart of tampering detection

To find out the hash representation, the image is divided into a number of blocks. Size of the blocks will be decided by the accuracy of tampering detection, localization and affordable length of the hash vector. Hash values are calculated for each block and are arranged at their respective block locations. This representation is called hash matrix $H(I)$. Absolute difference of hash matrices corresponding to original and tampered image gives the tampered region in the image. All the rows of the hash matrix when put together sequentially in form of a single row, constitute the hash vector $\mathbf{H}$ for the image. It will be shown in the following chapters how the hash vector $\mathbf{H}$ is used for the mathematical quantification of tampering area in the digital image.

As we have discussed earlier, proposed method is based on non-blind detection technique. It is suitable when the authenticity of the image under question is to be established and either image or its mathematical representation is available for the purpose of comparison. Here, we are focusing on detection of structural tampering and tampering like low pass filtering, brightness change, minor contrast change, JPEG compression etc. will be ignored. It implies that technique has to be robust against such content preserving manipulations and therefore it ignores them. Proposed method is especially suitable to cases of image tampering relating to celebrity life style, political leaders and sometimes when such attempts are aimed at vitiating law and order situation by raising communal and ethnic passion. Adding or deleting pictures, morphing images, superimposing two different pictures may have serious implications in cases mentioned above. Forensic science laboratories will find this technique very useful for two purposes:

1. Detection of structural tampering as mentioned above which impact on public opinion and law and order situation.

2. It will help in providing opinion to prosecution as expert opinion under law such as Indian Evidence Act.

2.1.1 Selection of threshold value of $\mathcal{D}$

A distance function $\mathcal{D}$ is defined to find out the "distance" between two matrices, i.e., original image matrix and the suspect image matrix, which determines whether tampering has taken placed or not and if so, what is the extent of tampering. The distance function $\mathcal{D}$ may be an Euclidian distance, hamming distance or an absolute distance etc. In present case, absolute difference between matrices has been used as the distance function. Let I_O, I_H and I_T be original image, image with content preserving manipulation and tampered image respectively. If we describe the hash matrices of these images as $HM(I_O)$, $HM(I_H)$ and $HM(I_T)$ respectively then distance between $HM(I_O)$ and $HM(I_H)$ should be always less than a critical value $\mathcal{D}_{C1}$, i.e.,

$$D[HM(I_H), HM(I_O)] < D_{C1}, \tag{2.1}$$

and distance between $HM(I_O)$ and $HM(I_T)$ should be always more than critical value $\mathcal{D}_{C2}$, i.e.,

$$D[HM(I_T), HM(I_O)] > D_{C2}, \tag{2.2}$$

where $\mathcal{D}_{C1}$ is the highest possible distance between original image and image with CPM, $\mathcal{D}_{C2}$ is the lowest possible distance between original and tampered images. If $\mathcal{D}_{C1}$ and $\mathcal{D}_{C2}$ are sufficiently separated then a threshold $\mathcal{D}_T$ can be decided which will discriminate between I_O, I_H and I_T efficiently with extremely low false alarms. $\mathcal{D}_{C1}$ decides the critical value of tampering below which the detection method is robust. Thus the combined condition becomes

$$D[HM(I_H), HM(I_O)] \ll D_{C1} < D_T < D_{C2} \ll D[HM(I_T), HM(I_O)]. \tag{2.3}$$

If $\mathcal{D}_{C1}$ and $\mathcal{D}_{C2}$ are sufficiently separated then we can conclude that the technique is robust against content preserving manipulations as well as it is sensitive to malicious tampering.

2.2 Detection of structural tampering

Various steps involved in the tampering detection process are as below:

1. Image of size $M \times M$ is divided into N number of blocks of size $q \times q$ by discrete movement of sampling block $q \times q$ horizontally and vertically over the image. Sampling started from left most edge of the row. Every time it is shifted by $q/2$ pixels to right for generating next block till it reaches the right most edge of the image. Total of t blocks are generated in each row. It should be noted that M is an integral multiple of t so that blocks on the boundary of the image are not prematurely truncated. Here we note that

   $$t = \frac{2M}{q} - 1 \tag{2.4}$$

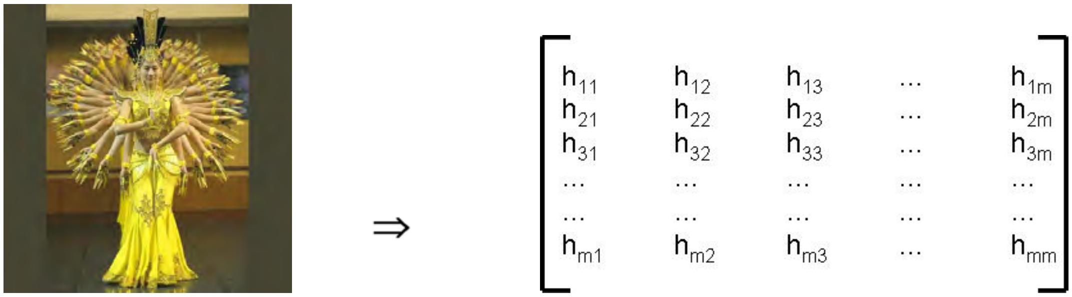
   After first row is done, we shift sampling block $q/2$ pixels below the first row and the same process is repeated as was done for the first row. The process is continued till the bottom most row to get $t^2 (= N)$ blocks.

2. For all N blocks in the image, the structural content is extracted and represented through a suitable value.

3. The structural indices of the blocks at location (i, j) are used to generate suitable hash values h_{ij} using a hash function which meets requirements discussed in sub-section (2.1).

4. Hash values h_{ij} are arranged at their respective block position to constitute hash matrix H of the image. Hash matrix is computed for original and suspect image.

5. Tampering detection and localization (TDL) is achieved by computing the absolute difference $|H_o - H_t|$ and suitably normalizing and converting it into a grey scale matrix which gives the tampering area in the image.



$$\Rightarrow \begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1m} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2m} \\ h_{31} & h_{32} & h_{33} & \cdots & h_{3m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ h_{m1} & h_{m2} & h_{m3} & \cdots & h_{mm} \end{bmatrix}$$

(a) An Image (b) (c) Conversion of image into hash matrix

Figure 2.2: This is how an image can be converted into corresponding hash matrix.

The corresponding algorithm that converts a digital image into a hash matrix, as represented schematically in fig. (2.2), can be described by the following flow chart of fig. (2.3):

2.3 Similarity Value

We propose a mathematical index called Similarity Value S, which is a measure of similarity between two hash matrices or equivalently two corresponding images. As the name implies S will have high value for similar images and low value for dissimilar images. Two images with corresponding hash matrices H^a and H^b are first converted into respective hash vectors $\mathbf{H}^a$ and $\mathbf{H}^b$ by concatenating the rows of the matrices in a single row. Ratio R_i, a measure of similarity between two hash vectors, is defined as:

$$R_i = \frac{exp\left[min\left(\mathbf{H}_i^a, \mathbf{H}_i^b\right)\right]}{exp\left[max\left(\mathbf{H}_i^a, \mathbf{H}_i^b\right)\right]}, \tag{2.5}$$

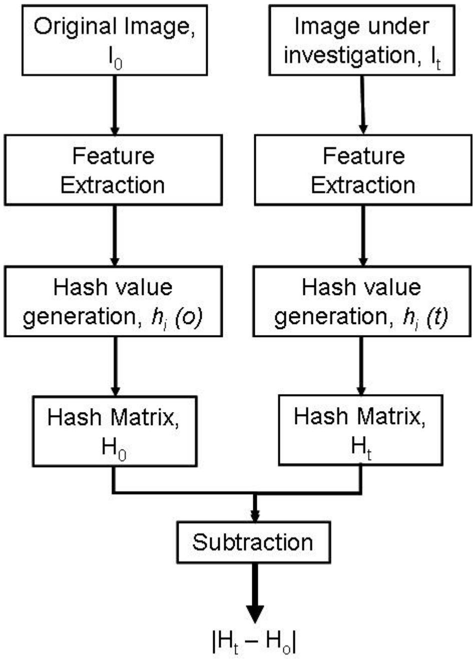


Figure 2.3: Flow chart shows the steps to (a) calculate hash value, (b) generate hash matrices and (c) carry out tampering detection and localization.

where $i = 1, 2, 3, ..., q^2$. From above equation, it can be inferred that R_i moves closer to 1 as $\mathbf{H}^a$ moves closer to $\mathbf{H}^b$. Now Similarity Value S is defined as:

$$S\left(\mathbf{H}^a, \mathbf{H}^b\right) = \frac{\prod_{R_i \in R_S} R_i}{\prod_{R_i \in R_L} R_i}. \tag{2.6}$$

In this equation, R_L and R_S consist of m largest and smallest R_i values where m is any selected integer. It can be seen that S varies between 0 and 1. It is closer to 1 for similar images and it moves towards 0 if the amount of tampering increases. It becomes 0 for completely dissimilar images.

If we look at R_i carefully, we observe that it gives an idea about (i) whether a particular block (i,j) in an image has been changed (tampered with) or not and (ii) if tampering has been done, R_i will give information about amount of tampering in that particular block. It can be seen that if there is no tampering in the block, value of R_i for the block under consideration will be 1. However, if there is some change, value of R_i will be less than 1 and will move towards 0 with increase in amount of tampering.

While R_i gives an idea about tampering in a block, S_i gives quantification of tampering in complete image. As expected value of S_i will reflect aggregate of all blocks in some way. Here, m should be selected in such a way that it is more than the number of tampered blocks. The most ideal selected value of m is equal to the number of tampered blocks in order to account for all tampered blocks accurately. If the tampering area is small, a value of m equal to 10, for example, will suffice.

CHAPTER 3

Correlation Coefficient Based Hash Generation Techniques

Structural content of the image is represented through pixel values. The pixel values of the image can be expressed as a random variable X. In the proposed method, correlation coefficient between random variable X and Y is found out where Y is random variable representing pixel values of a reference image [94]. The correlation coefficient thus becomes a measure of structural features of image X and is given by:

$$\rho_{xy} = \frac{Cov(X,Y)}{\sigma_x \sigma_y} = \frac{1}{N-1} \frac{\sum_{i=1}^{N} (X_i - \bar{X})(Y_i - \bar{Y})}{\sigma_x \sigma_y}, \tag{3.1}$$

where N is the number of pixels in the images A and B. Also X_i and Y_i are the random variables representing the i^{th} pixel value, $\bar{X}$ and $\bar{Y}$ represent the mean pixel value of images A and B respectively. σ_x and σ_y are standard deviations of X and Y.

$$\sigma_x = \sqrt{\frac{\sum_{i=1}^{N} (X_i - \bar{X})^2}{N-1}}, \tag{3.2}$$

and

$$\sigma_y = \sqrt{\frac{\sum_{i=1}^{N} (Y_i - \bar{Y})^2}{N-1}}. \tag{3.3}$$

To calculate correlation coefficient images of size 400 × 400 are taken and they are divided into overlapping block of size 50 × 50. Overlapping blocks are created to make sure that even minor tampering located near the boundary of a block is not missed out. A very small tampering located on both sides of a particular block boundary will be picked up by two overlapping blocks, ruling out the possibility of being left out. At least in one block complete tampering will be picked up. This results in higher sensitivity of the detection. Hash matrix is computed using algorithm given in section (2.2). In step 2, correlation coefficient is calculated for each block which becomes its structural index. Before going further, we will discuss reason for selection of correlation coefficient as feature extractor with regard to required property of irreversibility of hash function as listed in earlier chapter. We will discuss as to why some other common functions are not chosen for this purpose. Later, we will also discuss what kind of reference image is chosen and why.

3.1 Irreversibility of hash function

One of the requirements for a hash function is that it should be irreversible. It means that it should not be possible to back calculate the input message from its hash value. Qualitative explanation is as follows.

1. Let us consider a hash function $h(x)$ of input message x defined as,

   $$h(x) = ax + b, \tag{3.4}$$

   where x is pixel value and a, b are constants.

   Hash function defined above is not a good hash function because of the fact that from given $h(x)$ values, we may be able to estimate the values of x. So for this hash function, it will be easy for the attacker to calculate/estimate values of x and input message can be obtained.

2. We can discuss another example where we take mean value of the image block pixel values at its hash value, i.e.,

   $$h(x) = \frac{\sum_{i=1}^{N} X_i}{N}, \tag{3.5}$$

   where X_i is random variable representing the pixel values in the block and

N is total number of pixels in the block.

Here, it is not possible to back calculate the individual pixel values from the mean value of image block. Therefore, hash function is irreversible to an extent. However, if we arrange the mean values at their respective block locations, we will get a reasonable idea about the image. This conclusion is supported by the fact that neighboring image pixels are strongly related to each other. So the attacker might use this interdependency feature of pixel and hash values (mean) to get approximate description of the image.

In the proposed method, we have used correlations coefficient to generate hash value which is calculated using the formula,

$$\rho_{xy} = \frac{1}{N-1} \frac{\sum_{i=1}^{N} (X_i - \bar{X})(Y_i - \bar{Y})}{\sigma_x \sigma_y}, \tag{3.6}$$

where, X is random variable representing pixel values in the test image A and Y is random variable representing pixel values in reference image B.

We can observe from the formula for correlation coefficient that it is impossible to back calculate the values of individual pixels X_i and Y_i from given hash value. Also, we will not get any idea about the image from individual hash value plotted at respective block locations. We may recall that attacker might get some idea about the shape of image by plotting mean pixel value as hash value. In that sense, hash generation using correlation coefficient is far superior to one using mean value.

3.2 Selection of reference image

It has been mentioned that textured image is chosen as reference image to calculate correlation coefficient which is in turn used to generate hash values. Texture gives the information about pixel intensity in an image. Natural outdoor scenes make good textured images. Certain man made objects with patterns also constitute texture. However, “texture” is not very rigidly defined. Common examples of textured image are grass, stone pattern, sand, bricks, leaves etc.

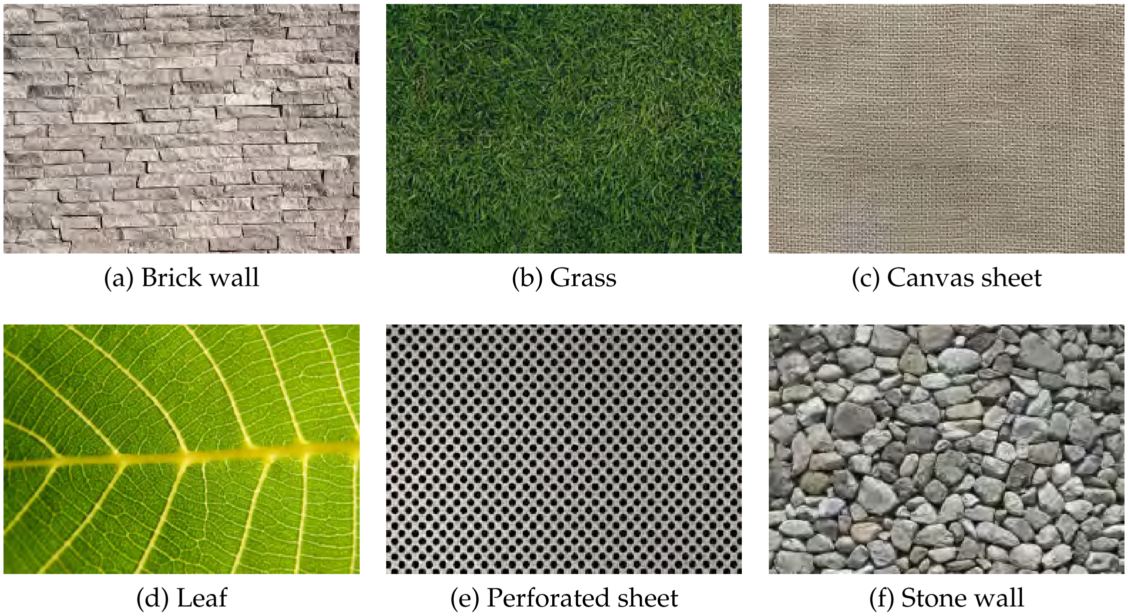

(a) Brick wall
(b) Grass
(c) Canvas sheet
(d) Leaf
(e) Perforated sheet
(f) Stone wall

Figure 3.1: Natural textured images.

Textured image is chosen as reference image as it has element of periodicity and predictability. A textured image which is low pass filtered and is devoid of sharp structural elements serves very well as reference image. An image with high frequency structural elements such as sharp edges, will make correlation coefficient vary in very unpredictable manner making it unsuitable as reference image.

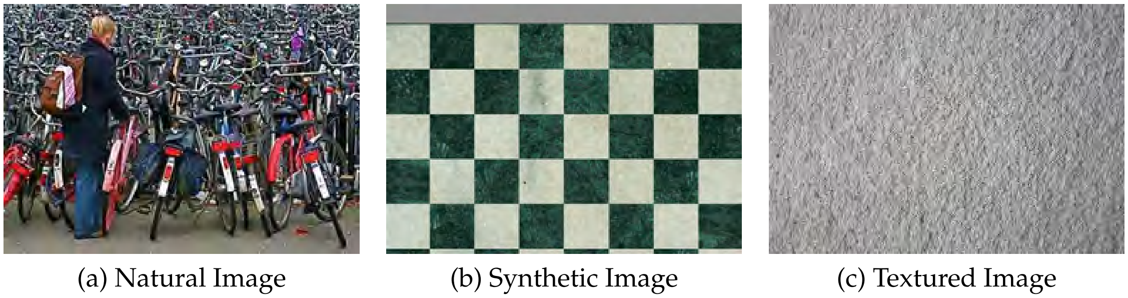

(a) Natural Image
(b) Synthetic Image
(c) Textured Image

Figure 3.2: Natural, Synthetic and Textured images for the purpose of understanding of suitability for a reference image.

To support this proposition, an experiment was carried out on a set of hundred images. Correlation coefficient of these images is found out with three different types of reference images namely normal natural image, synthetic image and textured image. Synthetic image is formed by arranging chosen periodic numbers representing pixel values at different pixel locations.

Correlation Coefficient	Textured	Normal Coloured	Synthetic
0.0-0.01	60	20	6
0.01-0.02	19	12	6
0.02-0.03	4	7	3
0.03-0.04	3	7	3
0.04-0.05	2	3	6
0.05-0.06	3	5	5
0.06-0.07	0	9	1
0.07-0.08	4	8	5
0.08-0.09	2	11	5
0.09-0.1	0	4	3
0.1-0.15	1	12	17
0.15-0.2	2	2	19
0.2-0.25	0	0	9
> 0.25	0	0	12

Table 3.1: This table explains why we chose a textured image as a reference image.

The number of images (out of 100 images) having correlation coefficient falling in different ranges is arranged in Table (3.1) which shows that the correlation coefficient obtained using textured image is consistent. It is expected that two arbitrary images (the suspected image and reference image in this case) will not be similar and therefore the similarity value of the two images will be very small.

In case of textured image as reference, all the similarity values are lying in the range 0.0 to 0.1, i.e., there similarity is very less. This pattern is not observed for synthetic and normal image being used as reference image. It is seen that in some cases using synthetic and natural images, the similarity value comes out to be more than 0.20 which means that there exists an element of randomness in correlation coefficients. Random pattern will not help us draw a conclusion. This justifies our selection of textured image as reference. The above result is also illustrated in fig. (3.3).

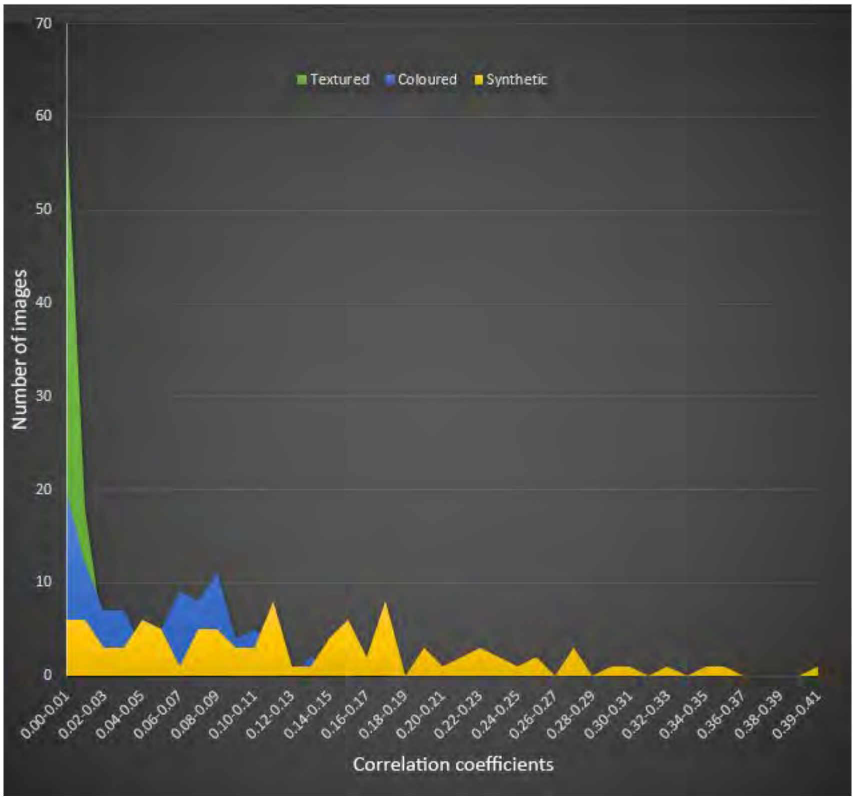


Figure 3.3: Line graph shows that our selection of textured image as reference image in valid.

3.3 Sensitivity and Robustness

Experiment was carried out on a set of 100 images of 50 × 50 size each. Reference image of same size was chosen with negligible structural content. The reference image is low pass filtered to suppress any structural features present.

Edges and sharp intensity transitions contribute significantly to high frequency components in the Fourier transform of the image. Hence blurring (low pass filtering) is required to suppress these high frequency components. This is done in order to ensure that feature vector values are coherent and don't exhibit arbitrary changes. Any of the following low pass filters can be used for this purpose generally.

3.3.1 Selection of low pass filter

At pre-processing stage image is low pass filtered for two reasons. 1) It reduces high frequency components corresponding to minor image modifications and 2) it also ensures that these minor modifications do not have any significant effect on hash values. There are a number of low pass filters used in image processing applications, such as, Ideal Low Pass Filter (ILPF), Butterworth Low Pass Filter (BLPF) and Gaussian Low Pass Filter (GLPF) etc. First two have very sharp cutoff, hence not suitable for our purpose to suppress high frequency components. GLPF is suitable for pre-processing the image as it has gradual cutoff and it gives smooth suppression of undesired high frequency components. Two dimensional GLPF is defined as

$$H(u,v) = e^{-D(u,v)^2/2\sigma^2} \tag{3.7}$$

where $D(u,v)$ is the distance from centre of frequency rectangle and D_0 is the cutoff frequency. A typical GLPF is shown in fig. (3.4)

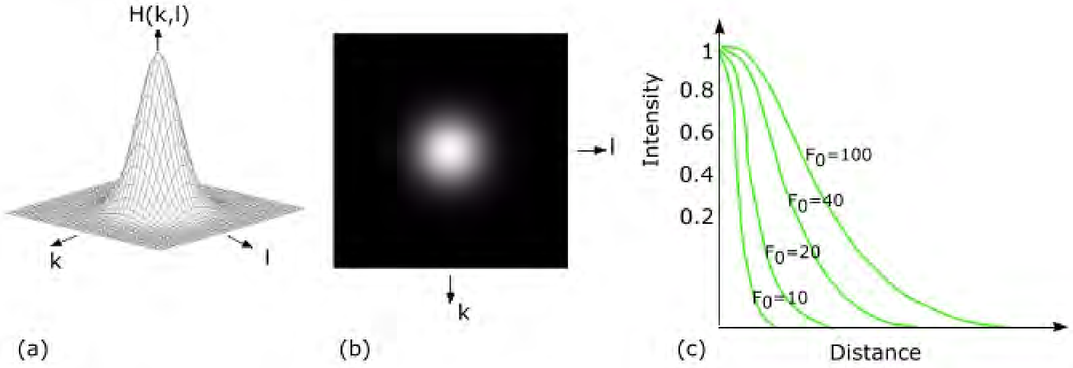


Figure 3.4: A typical Gaussian Low Pass Filter

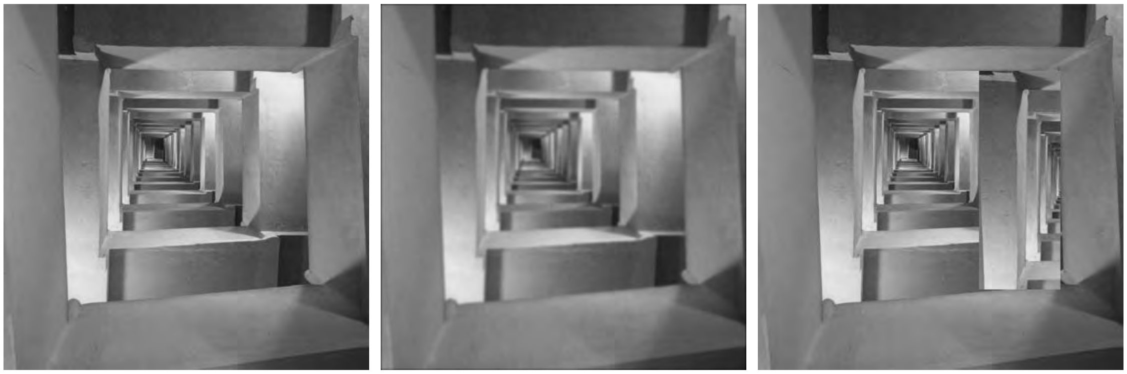
Above three categories cover range from very sharp (ideal) to very smooth (Gaussian) filtering. By changing the order of Butterworth filter, we can achieve desired type of filtering. For higher order values, it works as ideal filter while for lower order values, it approaches a Gaussian filter.

For our purpose, we will choose Gaussian filter for low pass filtering as it provides smooth filtering.

3.3.2 Correlation coefficient as feature extractor

To test the robustness against blurring operation, the set of images is blurred using a 7×7 Gaussian low pass filter. The correlation coefficient ρ_{br} of blurred images with reference image is then found out.

The same set of images was tampered using Photoshop. The approximate tampered area was of the size $200 - 300$ pixels. A typical set of original image, its blurred version and the tampered image is shown in fig. (3.5). Blurring is done by passing the image through the Gaussian low pass filter once. In the image (a) tampering is done by inserting the left part of the gallery on the right part of the image too.



(a) Original (b) Blurred (c) Tampered

Figure 3.5: Typical representation of (a) an original, (b) its blurred and (c) the tampered images.

To test the sensitivity and robustness of the algorithm, following correlation coefficients were found using eqn. (3.6).

1. Correlation coefficient between original and reference image ρ_{or}

2. Correlation coefficient between blurred and reference image ρ_{br}

3. Correlation coefficient between tampered and reference image ρ_{tr}

A graph representing three correlation coefficients ρ_{or}, ρ_{br} and ρ_{tr} for 30 images has been plotted in fig. (3.6). Points Blur1 and Blur2 represent the correlation coefficients for images which were blurred once and twice respectively.

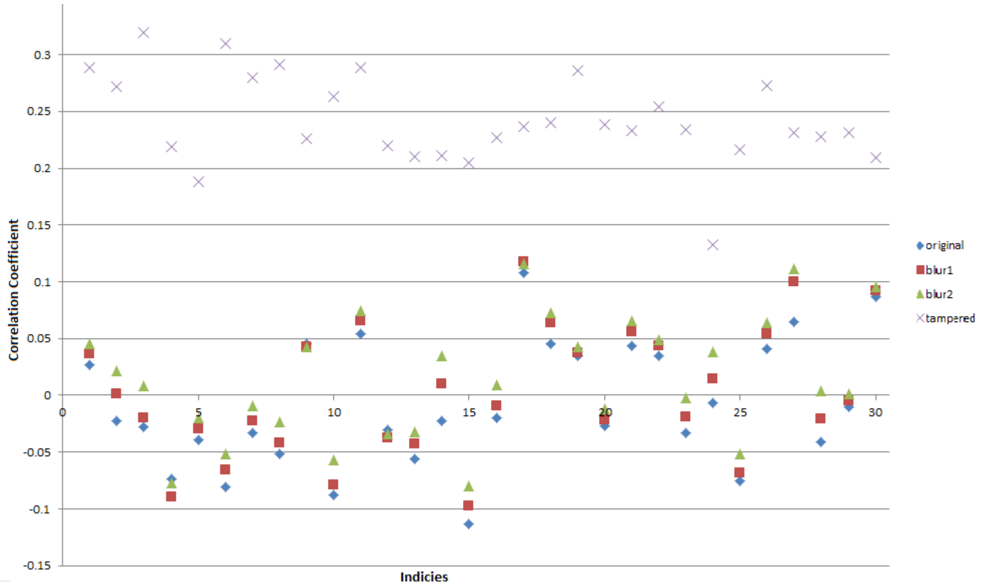


Figure 3.6: Correlation Coefficients for the 30 images shown for original, blur1 (low pass filtered once), blur2 (low pass filtered twice) and the tampered image.

The graph shows that the correlation coefficient ρ_{or} is very closed to ρ_{br} but there is a significant difference between ρ_{or} and ρ_{tr}. This observation will be used to distinguish between original and tampered images. A very small or negligible difference between ρ_{or} and ρ_{br} implies that a content preserving manipulation such as blurring by low pass filter has only a little impact on the structural index (correlation coefficient in this case). It means that the proposed algorithm is robust against blurring operation. It is further observed that the difference between ρ_{or} and ρ_{br} increases for doubly blurred images as compared to images blurred only once. If the image is repeatedly blurred, then it enters the category of tampered images. Hence we conclude that robustness against content preserving manipulations is not absolute and it depends on degree to which manipulation has been

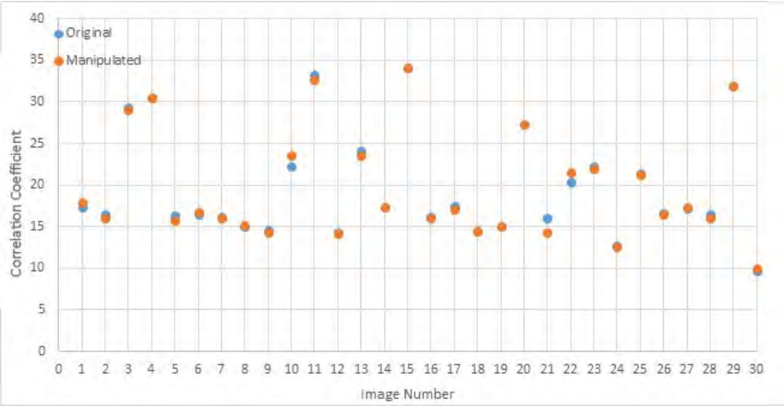


Figure 3.7: Correlation Coefficients for the 30 images shown for original and the manipulated images. The manipulations are inserted by using simple commands of Photoshop on original images.

done.

The above experiment to prove the robustness of proposed algorithm was done relatively on "controlled set" in the sense that at any given time all the images were made to undergo a particular type of content preserving manipulation and change in correlation coefficient was observed. Same results will be obtained if we carry out "normal" CPMs instead of controlled ones mentioned above. This was done by randomly applying CPMs, i.e., brightness and contrast change in varying amount. The correlation coefficient of such a set of images was found out. Graph of fig. (3.7) shows that high degree of robustness was exhibited by the experiment. The technique shows that it can effectively reject possible positive false alarms caused by CPMs.

3.4 Calculation of hash values and generation of hash matrix

As mentioned in chapter (2) square image of size $M \times M$ is divided into overlapping blocks of size $q \times q$. It should be noted that sampling block size is same as size of the reference image. Overlapping blocks are created by shifting the sampling block by $q/2$ pixels each time from left to right starting from left top corner of the image. The same procedure is repeated for the second row which is $q/2$

pixels below the first row. For avoiding incomplete blocks on the extreme right and the extreme bottom of the image, M should be integral multiple of q. The process is carried out for the whole image and we get a total of t^2 blocks where

$$t = (2M/q) - 1. \tag{3.8}$$

By using the algorithm mentioned in chapter (2), the correlation coefficient ρ_{ij} of every block at location (i, j) is found out where $i, j = 1, 2, 3, ..., t$. To avoid negative values of correlation coefficients and to scale it up, we define a new variable p_{ij} as below:

$$p_{ij} = 4 \times (\rho_{ij} + 1), \tag{3.9}$$

where $i, j = 1, 2, 3, ..., t$ and ρ_{ij} is the correlation coefficient of $(i, j)^{th}$ block with the reference image. Hash value of a block (i, j), denoted by h_{ij}, is given by 4-neighbourhood sum as below:

$$h_{ij} = p_{i-1,j} + p_{i,j} + p_{i+1,j} + p_{i,j+1} + p_{i,j-1} \tag{3.10}$$

4-neighbourhood sum is used to calculate hash value so that even minor tampering existing across the block boundary is not missed. By summing, it is ensured that even minute tampering is detected efficiently. However, it is at the cost of increased tampering detection area shown by the algorithm. By positioning hash values h_{ij} so obtained at their respective block positions, we get a hash matrix H

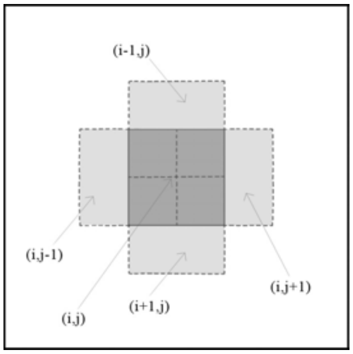


Figure 3.8: The central block is at i^{th} row and j^{th} column of the blocks of the image. The four neighbouring blocks are obtained shifting by half block width to the left (i,j-1), right (i,j+1), up (i-1,j) and down (i+1,j).

of size $t \times t$. Once the hash matrices for original and suspect images have been calculated, the tampering area can be detected by finding out the absolute difference $|H_o - H_t|$ as shown in Table (3.2).

3.5 Experiments and Results

Algorithm was tested for a set of 100 images of 400×400 size. Adobe Photoshop CS2 was used to tamper these images. Image was divided into 225 blocks by using sampling block size of 50×50. Hash values were calculated following the algorithm mentioned in section (3.4). Hash matrices and hash vectors were created to carry out tampering detection and find out similarity value. The similarity values were calculated for varying amount of tampering. The result are shown in Table (3.2). It can be seen that similarity value reduces with the increase in tampering. As amount of tampering is increased, value of S moves away from 1 towards 0.

3.5.1 Sampling block size v/s accuracy of localization

As we are assigning single hash value to each block, tampered area which is much less than a block size will be seen over the full block. Therefore, it makes sense to reduce the size of sampling block to locate the tampered region more accurately. This reduction in size of the sampling block will depend on how accurately the tampering localization is to be done. If only the existence of tampering or otherwise is required to be established, sampling size of 50×50 will serve the purpose. However, for more accurate localization, sampling block size may be reduced to 25×25 or even 20×20. A comparative localization by 50×50 and 25×25 sampling is shown in Table (3.3). Precise localization of tampering becomes the unique selling point of this algorithm. This is one of the major advantages over other existing methods [60, 95, 96]. During extensive literature survey on the subject, nowhere it was found where variation in sample size was used to improve the optimality in accuracy of tampering size detection.

It has been shown experimentally that accuracy of tampering detection can be improved by reducing the size of sampling block in Table (3.3). A general algorithm which will automatically select the appropriate sampling block size is as follows:

First, sampling block size is taken to be $q \times q$ and tampering area is found out. If area found is greater than the area of single block size at each location in the image, our selection of block size is appropriate and that is the minimum sampling block size which will adequately serve our purpose. If tampered area is less than single block size even at one location in the image, the algorithm reduces the block size to $q/2 \times q/2$ and tampering detection is carried out. Again, if tampered area is greater than $q/2 \times q/2$ size at all locations, the size $q/2 \times q/2$ is appropriate. If tampered area is less than $q/2 \times q/2$, even at single location, process is repeated with reduced sampling block size of $q/4 \times q/4$ size. This repetitive method is carried out till tampered area size at each location becomes more than selected sampling block size.

Table (3.3) shows that by reducing the sampling block size, the tampering localization achieved is more accurate in case of small size tampering. It shows the comparison of results obtained using 50×50 and 25×25 sampling block sizes. Results in Table (3.2) show how similarity value varies with variation in amount of tampering. It shows that if tampering area increases, value of S decreases.

To the best of our literature survey, we found that most of the reported tampering detection algorithms were experimented on the images having large tampered area that could even be detected by naked eyes if the original images were available to compare with. Our objective of study was different. In our case the structural changes made in the image are not at all noticeable by human eye. We found that the proposed algorithm was able to detect structural tampering of any size (howsoever small it was) in the image. However, it comes at the cost of hash size. In the course of study, we represented the amount of tampering through a mathematical index called Similarity Value. We also successfully detected the tampered area and localized it accurately. It was found that the technique was not only able to detect and localize the tampered part of the image but also correctly measure the amount of tampering. The algorithm was tested for large tampering, small tampering and tampering at multiple places in a single image. Results were found encouraging. Robustness of this method against the content preserving manipulation parameters were also observed.

3.6 Conclusion

Correlation coefficient was used as feature extractor to generate hash in this chapter. The hash function was tested for sensitivity and robustness and was found to be efficient in detection of tampering. Variation in the similarity value was shown with change in amount in tampering in the image. Accuracy of localization was achieved through change in sampling block size.

Use of correlation coefficient ensures that eavesdropper is not able to get any idea about the image from its hash representation. However, security of the hash was not evaluated mathematically in this chapter. This will be done in the next chapter, i.e., chapter 4, where we will use Singular Value Decomposition technique in conjunction with correlation coefficient feature extractor for hash generation.

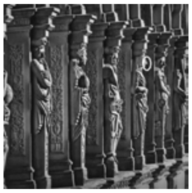
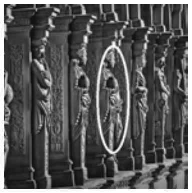
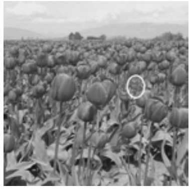
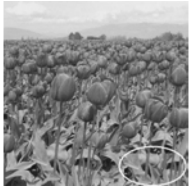
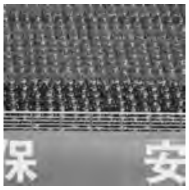
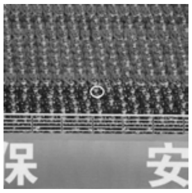
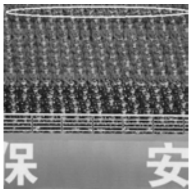
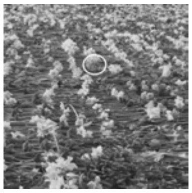
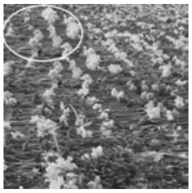
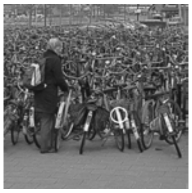
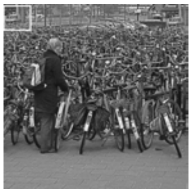
No.	Original Image	Image with small tampering	Image with large tampering	Similarity value for small tampered area	Similarity value for large tampered area
1				0.6405	3.30E-04
2				0.449	1.10E-04
3				0.2454	4.86E-03
4				0.1338	1.91E-05
5				0.363	2.19E-01

Table 3.2: The similarity value changes according to the size of tampering in the images.

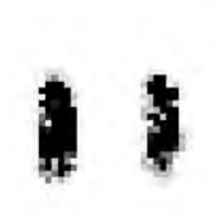
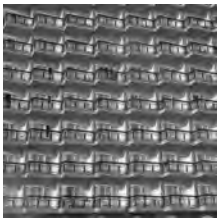
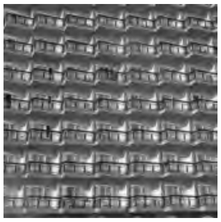
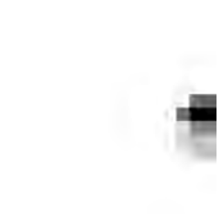
No.	Original Image	Tampered Image	Localization of tampering using 50 × 50 block	Localization of tampering using 25 × 25 block
1				
2				
3				
4				
5				
6				

Table 3.3: Accuracy in localization of tampering in the images is achieved by reducing the sampling block size.

Chapter 4

Singular Value Decomposition Based Hash Generation for Detection of Structural Tampering

In the previous chapter, correlation coefficient was used as feature which was extracted and used for hash generation. As discussed earlier, correlation coefficient function is extremely irreversible and it is impossible to back-calculate the image matrix from hash values. Therefore, it is quite secure. Meaning, if at all an eavesdropper gets the hash value while it is being transmitted, he or she cannot make out, from that value, the image under forensic investigation. However, security of hash is very important and researchers all over the world have been working to make hash functions more and more secure. Like in online banking transactions the security features migrated from 32-bit to 64-bit to the current 128-bit encryption. Similarly, in case of transaction through credit cards, CVV number is not the only security cover. One may be asked for OTP (one time password) and also is advised to use a second tier security by using chip enabled card.

Use of reference image for calculating correlation coefficient in previous chapter can be treated as key based hash generation where reference image is the key. In the method proposed in this chapter, we will be using Singular Value Decomposition (SVD) of image matrix and add another level of key to generate the hash. Consequently, the method will be a 2-level key based hash generation which is more secure than the earlier method.

Two-level key based hash generation uses Singular Value Decomposition (SVD) of the image matrix. SVD gives an efficient representation of the features of the image [97] and thus helps in designing a very suitable hash matrix. This matrix is used for detection of tampering in the image.

4.1 Singular Value Decomposition

For an $m \times n$ matrix A with rank r, there exists an $m \times n$ matrix Σ and there exist an $m \times m$ orthogonal matrix U and an $n \times n$ orthogonal matrix V such that

$$A = U \, \Sigma \, V^T \tag{4.1}$$

where,

$$\Sigma = \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix}$$

and D is a diagonal matrix with singular values on the diagonal as shown below:

$$D = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_k \end{pmatrix}.$$

Here σ_1, σ_2, ..., σ_k, are the singular values of A. These singular values are the square roots of eigenvalues $\lambda_i s$ of $A^T A$. First few singular values contain the maximum information about the image while the later values are relatively insignificant. This property enables us to use reduced rank approximation method to represent the image in much lesser space. D is an $r \times r$ diagonal matrix for some r not exceeding the smaller of m and n. Any factorization of the form $A = U \Sigma V^T$ with orthogonal matrices U, V and Σ as defined in eqn. (4.1) is called a Singular Value Decomposition (SVD) of the matrix A. Matrices U and V are not uniquely determined but diagonal values of Σ are necessarily unique and are called singular values of A. The columns of orthogonal matrix U are called left singular vectors and columns of V are called right singular vectors of A. The structural features of the image are mainly described by matrices U and V [98].

Singular Value Decomposition is a very useful tool to represent an image. The decomposition allows us to separate an image into two components, one represents the structural features of image and second the luminance component of the image. The two components are represented through orthogonal matrices and singular value matrix respectively.

4.2 Proposed scheme for key based hash generation

SVD based hash generation method is inherently secure. This is because of two reasons: (1) the method of division of the image in the number of blocks is unique and is known only to the person who uses a particular algorithm for purpose of division and (2) some features of a reference image (which is discussed in following paragraphs) are imported and introduced in the image, which is not known to the attacker.

The use of reference image matrix in place of original Σ matrix amounts to introducing a key in hash generation process. For the same image, its hash representation can be changed by changing the key which is reference image in this case. Due to this, the attacker will not be able to read the hash even if he gets hold of original image. He needs to have both, the original and reference image to be able to decipher the hash. The introduction of key introduces one more level of security to the algorithm.

4.3 Steps of SVD based algorithm

The structural details of image are contained in the luminance values of the concerned pixels. Therefore while going for tampering detection, only the luminance values, i.e., Y components of YCrCb description is considered and relevant experiments are conducted. The proposed algorithm consists of following steps:

1. A reference image which is predominantly a textured image, is selected. This image acts as reference for calculating the correlation coefficient and also provides the key for corresponding hash generation.

2. Sampling block of the size of reference image is moved discretely length and breadth-wise across the image to divide it in N overlapping blocks as discussed in chapter (3).

3. Singular value decomposition $U\Sigma V^T$ is obtained for each of the N blocks of the image.

4. The Σ matrix in above representation is replaced by Σ_{ref} matrix from SVD of reference image. It has been mentioned earlier that change in Σ matrix of the image modifies the image only marginally.

5. The image is reconstructed by computing $U\Sigma_{ref}V^T$.

6. Correlation coefficients are calculated for each of the N blocks.

7. From correlation coefficient, hash value is calculated using a suitably formulated set of eqns. (3.9, 3.10). This is done to avoid negative values of correlation coefficients.

8. Hash values for different blocks when arranged in a matrix form, constitute a hash matrix.

9. The above calculations are repeated for original image, image with CPM and tampered image.

10. The absolute difference of the hash matrices corresponding to original and tampered images shows the tampered region and its localization.

The correlation coefficient mentioned in above algorithm is defined as

$$\rho_{xy} = \frac{1}{N-1} \frac{\sum_{i=1}^{N} (X_i - \bar{X})(Y_i - \bar{Y})}{\sigma_x \sigma_y}. \tag{4.2}$$

In this equation X_i and Y_i are the random variables which represent the pixel intensity values of the i^{th} pixel of the images A and B. $\bar{X}$ and $\bar{Y}$ represent the mean pixel values of the images A and B respectively. Similarly σ_x and σ_y are corresponding standard deviations of A and B. N is the number of pixels in the image.

4.4 Robustness against content preserving manipulation

Correlation coefficient of an image X with reference image Y is given by

$$\rho_{XY} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} = Q_1(say). \tag{4.3}$$

Having already discussed effect of low pass filtering on correlation coefficient and found it to be robust, we will discuss effect of change in brightness level and contrast on the same. For image X under consideration, we will select a reference image Y and define two indices Q_2 and Q_3 representing brightness level and contrast followed by the study of the effect of change in them on Q_1. To find out the effect of change in mean brightness level on Q_1, we define brightness level index Q_2 as:

$$Q_2 = \frac{2\bar{x}\bar{y}}{\bar{x}^2 + \bar{y}^2}, \tag{4.4}$$

where $\bar{x}$ and $\bar{y}$ are average mean brightness values of image X and Y (reference image) respectively. Similarly, to evaluate the effect of change in contrast variation on Q_1, we define a contrast index Q_3 given by

$$Q_3 = \frac{2{\sigma_X}^2{\sigma_Y}^2}{{\sigma_X}^2 + {\sigma_Y}^2}, \tag{4.5}$$

where σ_X and σ_Y are standard deviations of image X and Y respectively.

Change in brightness level index (given by Q_2) and change in contrast index of the image (given by Q_3) are content preserving manipulations. It will be shown that change in Q_2 and Q_3 changes Q_1 very marginally. It means that the proposed method which uses correlation coefficient (Q_1) for hash generation is immune to changes in mean brightness level and contrast change to a large extent. It shows the robustness of proposed method against the content preserving manipulations mentioned above. Results are shown in figs. (4.1, 4.2, 4.3).

4.5 Experiments and Results

The database of 100 images of 400 × 400 size was taken to implement the algorithm. The tampering in the image was affected through Adobe Photoshop CS2. A set of 10 images were taken to compute the similarity values. It was found that the similarity value for tampered images was less than 1 and it goes on reducing as the amount of tampering increases in the image. It is logical to conclude that similarity value for non-tampered image is 1. As result shows, there is significant gap between similarity values of tampered and non-tampered images. This helps us in finding a suitable threshold for similarity value from which we can distinguish tampered image from the original one. Table (4.1) shows the similarity values for 5 images.

4.5.1 Detection and localization of tampering

The experiment was carried out for a database of 100 images. Hash matrices of original and tampered image were computed using a common reference image of size 50 × 50. Absolute difference of hash matrices of original image, H_O and tampered image, H_t was obtained. Difference $|H_o - H_t|$ is suitably normalized and converted into a grey scale matrix. Reconstruction of this matrix gives a grey scale image which is used for localization of tampering. The results are shown in Table (4.2).

The correlation coefficient of an image block with reference image is statistical property of all the pixels contained in the corresponding block and is represented through a single numerical value. Therefore, even if the tampering area is limited to a size much smaller than the sampling block size, the difference matrix

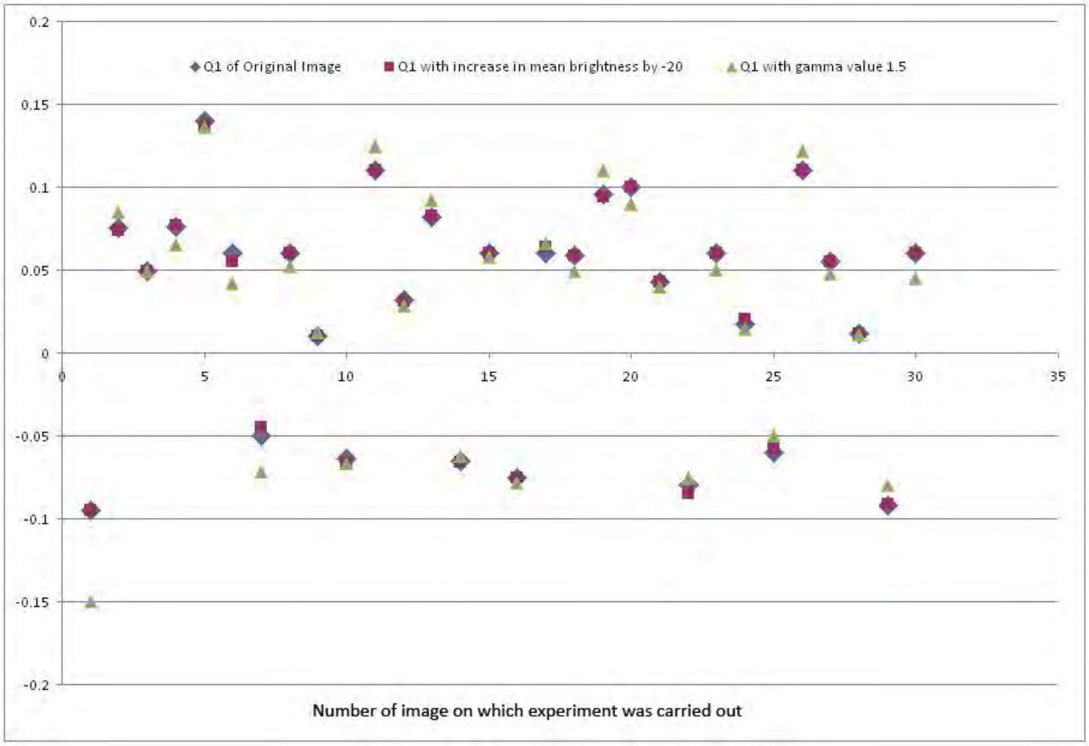


Figure 4.1: Graph showing effect of change in mean brightness level (index Q_2) by -20 and change in contrast variation (index Q_3) with $\gamma = 1.5$ on correlation coefficient (index Q_1) of the image with a preselected reference image.

$|H_o - H_t|$ is spread all over the block size. First, a sampling block size of 50×50 was taken and it was found that $|H_o - H_t|$ was visible over a full block size. To have more accurate localization sampling block size was reduced to 25×25. It was found that tampering localization could be done more precisely over much smaller area. This is a very a significant improvement over existing techniques.

The objectives of using SVD method were fourfold, viz., (a) to test the sensitivity of tampering detection technique, (b) to generate a secure hash, (c) to locate the tampered area accurately and (d) to define a mathematical index to quantify the amount of tampering in the image. It was found experimentally that all the objectives were successfully achieved. It was also demonstrated that Similarity Value gives quantitative measurement of tampering. Robustness of the technique against brightness level manipulations and contrast change was also proved experimentally.

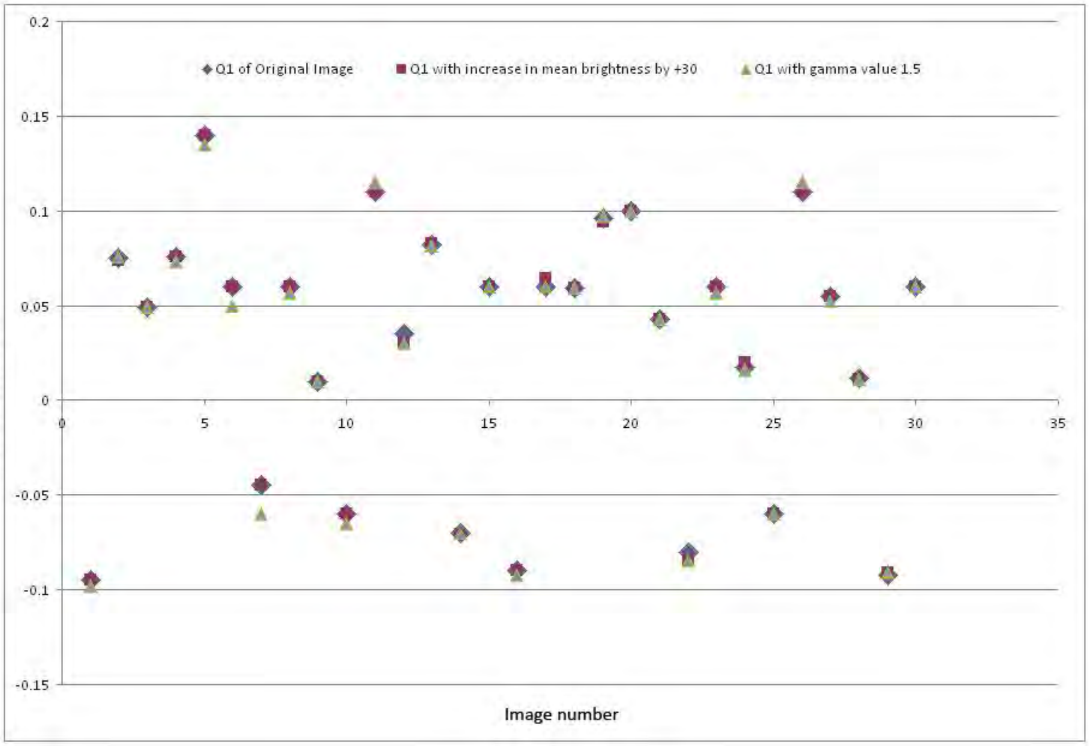


Figure 4.2: Graph showing effect of change in mean brightness level (index Q_2) by 30 and change in contrast variation (index Q_3) with $\gamma = 1.5$ on correlation coefficient (index Q_1) of the image with a preselected reference image.

4.5.2 Collision probability test

As discussed earlier, there should be a strict one-to-one correspondence between image I and its hash representation $h(I)$. In no case, two images should map to same value otherwise authentication of the image will not be possible. It will make no sense to talk about integrity of the image if we are not sure of which image's hash values we are referring to. Therefore probability of hash values of two images $h(I_1)$ and $h(I_2)$ being equal should be extremely low, i.e.,

$$P[h(I_1) \neq h(I_2)] \geq 1 - \theta, for\ some\ 0 \leq \theta \leq 1. \quad (4.6)$$

Here θ is a very small number. It is found experimentally that collision probability of used hash function is extremely low. For this, pair of $124,000$ images were randomly selected. Similarity values for all the pairs were calculated using an algorithm described in this section. Similarity value histogram was plotted which was found resemble gamma distribution.

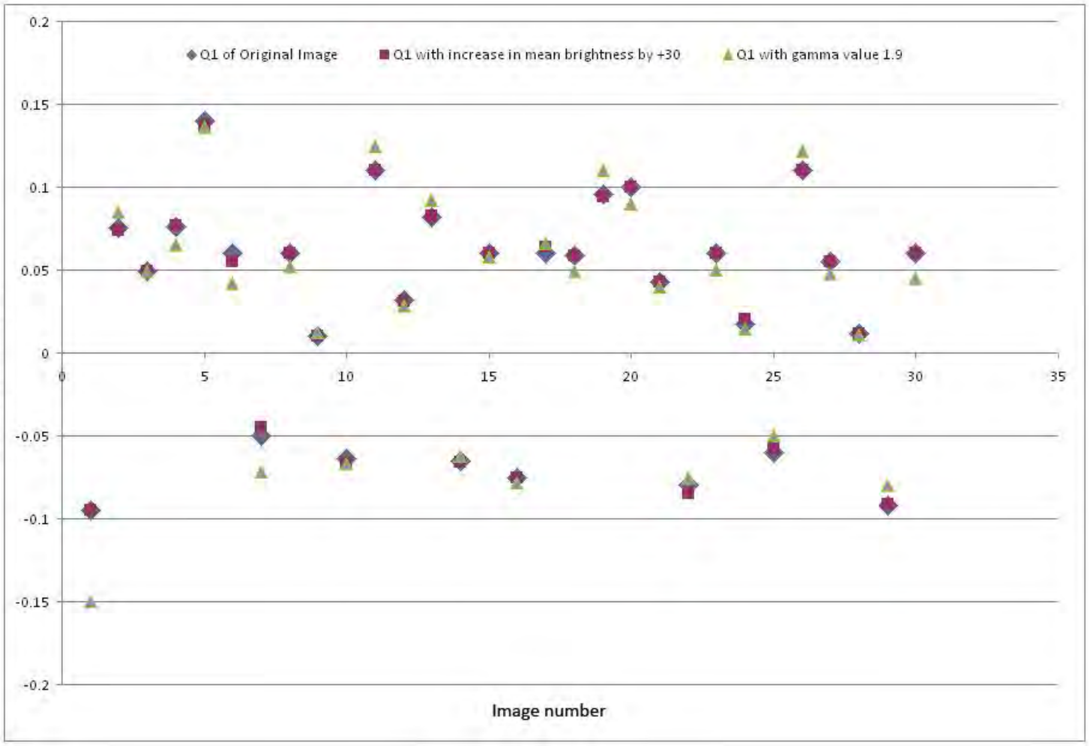


Figure 4.3: Graph showing effect of change in mean brightness level (index Q_2) by 30 and change in contrast variation (index Q_3) with $\gamma = 1.9$ on correlation coefficient (index Q_1) of the image with a preselected reference image.

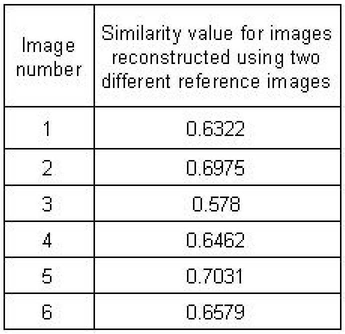


Image number	Similarity value for images reconstructed using two different reference images
1	0.6322
2	0.6975
3	0.578
4	0.6462
5	0.7031
6	0.6579

Figure 4.4: Table shows that by changing the reference image, the reconstructed image also changes resulting into similarity value different from 1 between the two reconstructed images.

Gamma distribution is given by [99]

$$f_x(x) = \frac{a^p e^{-\alpha x} x^{p-1}}{\Gamma_p}, \tag{4.7}$$

where a and p are parameters of the distribution. For curve fitting we need to calculate α and p-parameters for Chi-square test which is carried out as following:

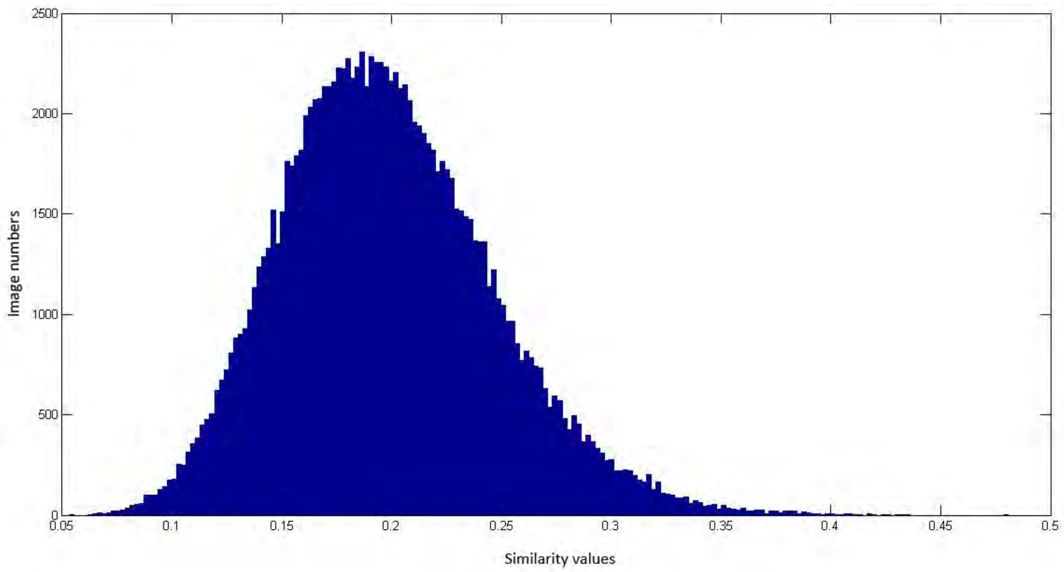


Figure 4.5: Histogram plotted of Similarity values of 124, 000 pairs of images showing a Gaussian distribution.

Expectation $E(x)$ and variance $V(x)$ are given as:

$$E(x) = \int_{-\infty}^{+\infty} xp(x)dx, \tag{4.8}$$

which gives mean

$$\mu = \frac{p}{\alpha} \tag{4.9}$$

and

$$V(x) = \int_{-\infty}^{+\infty} x^2 p(x)dx, \tag{4.10}$$

which gives variance

$$\sigma^2 = \frac{p(p-1)}{\alpha^2}. \tag{4.11}$$

From the given data of the similarity value for a set of 124, 000 pairs, we calculate mean μ and variance σ^2. We divide probability from 0 to 1 in bins of size 0.05. Number of similarity values belonging to each bin is plotted to give a histogram shown in fig. (4.5). From the obtained data μ and σ^2 are calculated using formula:

$$\mu = \frac{1}{N} \Sigma f_i x_i \tag{4.12}$$

and

$$\sigma^2 = \frac{1}{N-1}\Sigma(x_i - \bar{x})^2. \tag{4.13}$$

These values of μ and σ are substituted in eqns. (4.9 and 4.11) to find values of α and p.

Chi Square Test was carried out to fit the data distribution using formula

$$\chi^2 = \sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i}. \tag{4.14}$$

γ-distribution parameters were obtained as $\alpha = 15.9445$ and $p = 0.0127$. These values were used to calculate χ^2 which shows that the data fits γ-distribution. These parameters were used to calculate collision probability as below:

$$P[S \geq 0.8] = 3.95 \times 10^{-13}. \tag{4.15}$$

Therefore

$$P[dissimilarity] \geq 1 - 3.95 \times 10^{-13}. \tag{4.16}$$

Here dissimilarity has been calculated in a sense where similarity has been chosen (defined) as $S \geq 0.8$. Thus we conclude that collision probability requirement described by eqn. (4.6) is satisfied where $\theta = 3.95 \times 10^{-13}$. This experimental result shows that our hash function has extremely low collision probability. Low collision probability is required quality of a good hash function and it has been shown mathematically through experiment. It ensures that the hash value under consideration corresponds to a particular image. If collision probability is high then checking of integrity of an image becomes meaningless as we are not sure of the correspondence between image and its hash.

4.6 Conclusion

Sensitivity and robustness of SVD method was, thus, proved experimentally which have been shown in figs. (4.1), (4.2) and (4.3) graphically. But the salient feature of this method is higher security of hash due to introduction of second key through importing singular value matrix of reference image. This technique is therefore highly recommended for high security applications such as defense and space. High quality of hash function was proved through calculating collision probability which came to be 3.95×10^{-13} which is indeed very low.

No.	Original Image	Tampered Image	Similarity Value
1			0.17661479
2			0.34758645
3			0.31756991
5			0.0001

Table 4.1: Similarity values for a set of five images are shown in this table along with the corresponding original and tampered images. The regions where tampering is done have also been highlighted in white circles to show that none of these tampering is possibly visually perceptible.

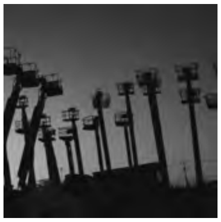
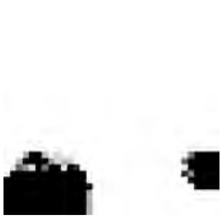
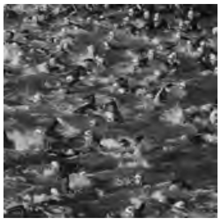
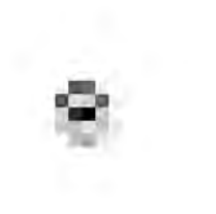
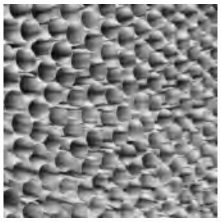
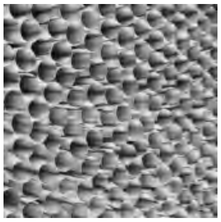
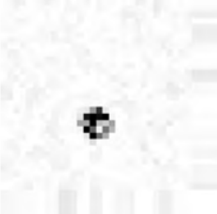
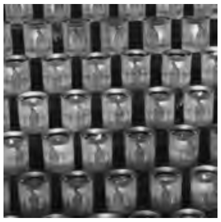
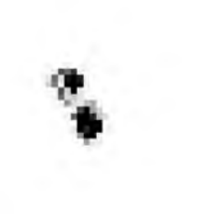
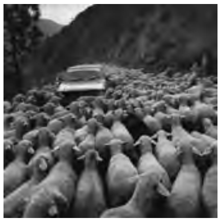
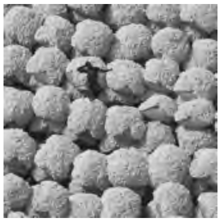
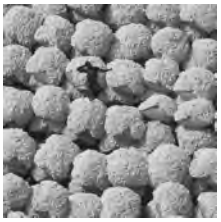
No.	Original Image	Tampered Image	Localization of tampering using 50×50 block	Localization of tampering using 25×25 block
1				
2				
3				
4				
5				
6				

Table 4.2: Accuracy in localization of tampering in the images is achieved by reducing the sampling block size.

Chapter 5

Hash Technique Using Canny Edge Detector for Structural Tampering Detection

The structural information in an image is contained in the edges existing in it [100]. Any alteration, removal or insertion of edges amounts to structural tampering in the image. This can be affected through bringing part of a different image into the original one or by removing the part of original image or sometimes removing a part followed by inserting segment of a different image. Proposed method aims that generating a hash value corresponding to the edge content of the image. Canny Edge Detector will be used to extract the edges in the image. A suitable hash function will be used to generate a hash value depending on the edge information in the block.

5.1 Standard edge detectors

Edge detection process serves to simplify the analysis of image by drastically reducing the amount of data to be processed while at the same time preserving useful structural information about object boundaries. As we are concerned about structural tampering, edge detection proves to be a strong tool in tampering detection. There are a number of edge detectors available used for different purposes [101, 102, 103]. A great deal of such detectors have been studied in the literature [104, 105, 106]. All these detectors have common criteria to evaluate to edge detection performance. A number of edge detectors are used in image processing applications, such as, gradient operator, Roberts cross gradient operator, Prewitt operator and Sobel operator.

However, these operators do not have element of thresholding and therefore they are likely to give spurious edges in the output. Therefore these filters are not particularly suitable for tampering detection which should inherently have robustness property. In view of this advanced edge detection technique are needed. Canny Edge Detector is superior to detectors discussed so far and is based on three basic objectives.

1. **Low Error Rate** - Detector should detect all edges and there should not be any spurious edges which means that output must be as close as possible to true edges.

2. **Edge Point Localization** - Edges located must be as close as possible to the true edges, i.e., the distance between point marked as an edge by detector and the center of true edge should be minimum.

3. **Single Edge Point Response** - One edge output should be generated to each single edge in the image, i.e., number of local maxima around the true edge should be minimum.

The beauty of Canny edge detector lies in the fact that above three criteria were expressed mathematically by Canny and optimal solution was found which satisfied all the objectives. Numerical optimization was used for 1-D step edge and noise was assumed to be added in form of white Gaussian noise. It was concluded that an optimal step edge detector can be approximated quite well with first derivative of Gaussian as:

$$\frac{d}{dx}\ e^{\frac{-x^2}{2\sigma^2}} = \frac{x}{2\sigma^2}\ e^{\frac{-x^2}{2\sigma^2}}. \tag{5.1}$$

1-D approach can be extended to 2-D which will require applying 1-D detector in all possible direction. It effectively means first smoothing the image by a circular 2-D Gaussian function followed by finding the gradient of the result. Gradient magnitude and direction are then used to estimate the strength and direction of the edge at each point. Canny Edge Detector algorithm has been used in varied application areas,such as, FPGA Implementation [107], in electron microscopy biological images using statistical dispersion [108], virtual hexagonal image structure [109], for feature extraction in satellite images [110], on Compute Unified Device Architecture (CUDA) [111] which is a parallel computing platform that has attained significance in modern gaming technology, extraction of face contours [112] etc. The ubiquity of Canny Edge Detector has encouraged researchers to attempt on reducing the computational time by skipping the smoothing step. This uses a fractional integral mask instead of integer gradient mask [113]. It is found that if image is too noisy, the final output of Canny Edge Detector may produce broken edges. This limitation is taken care of by using a discrete particle swarm optimization algorithm [114] which finally detects continuous edges in noisy images. In VLSI technology application where the concept of global image gradient histogram may not be desired, a distributed Canny Edge Detecor algorithm has been proposed [115]. This algorithm adaptively computes the edge detection threshold by judging local distribution of gradients in any block.

Let us assume the input image be $f(x,y)$ and Gaussian function be denoted by

$$G(x,y) = e^{\frac{-x^2+y^2}{2\sigma^2}}. \tag{5.2}$$

Smoothening is done by convolving G and f:

$$f_s(x,y) = G(x,y) * f(x,y). \tag{5.3}$$

The magnitude and direction of the edge is computed using following formula:

$$M(x,y) = \sqrt{{g_x}^2 + {g_y}^2}. \tag{5.4}$$

and

$$\alpha(x,y) = tan^{-1}\left[\frac{g_y}{g_x}\right], \tag{5.5}$$

where $g_x = \partial f_s/\partial x$ and $g_y = \partial f_s/\partial y$.

Any of the filters described earlier can be used to find g_x and g_y. The convolution of G and f is implemented using an $n \times n$ Gaussian mask. As $M(x,y)$ and $\alpha(x,y)$ are computed at each pixel location, their array size same as that of the image.

As edges are computed using gradient function, $M(x,y)$ will contain wide ridges around local maxima which needs to be thinned down. This is done by using non-maxima suppression. This is achieved by specifying a number of discrete orientations of gradient vector (edge normal). For example, a 3×3 matrix will have four orientations for an edge passing through centre point of region namely, horizontal, vertical, $+45^0$ and -45^0.

For a 3×3 region cantered at every point (x,y) we can formulate non maxima suppression scheme as below:

1. Find the closest direction d_k to $\alpha(x,y)$

2. If value of $M(x,y)$ is less than at least one of its two neighbours along d_k then let $g_N(x,y) = 0$ (suppression), otherwise let $g_N(x,y) = M(x,y)$ where $g_N(x,y)$ is the non maxima suppressed image. With non maxima edges being suppressed, $g_N(x,y)$ contains only these edges and equals $M(x,y)$.

The final step is to threshold $g_N(x,y)$ to eliminate false edges points. Single thresholding has limitations. If threshold is set too low, it gives false positives. On the contrary, if it is set too high, it gives false negatives. To overcome this situation double thresholding is adopted. Canny's algorithm attempts to improve the result by hysteresis thresholding. Ratio between high threshold T_H and low threshold T_L should be two or three to one.

Two level thresholding can be considered as creating two additional images given by

$$g_{NH}(x,y) = g_N(x,y) \geq T_H \tag{5.6}$$

$$g_{NL}(x,y) = g_N(x,y) \geq T_L. \tag{5.7}$$

Initially $g_{NH}(x,y)$ and $g_{NL}(x,y)$ are set to 0. After thresholding $g_{NH}(x,y)$ will have lesser nonzero pixels than $g_{NL}(x,y)$ but all nonzero pixels in $g_{NH}(x,y)$ will be contained in $g_{NL}(x,y)$ as $g_{NL}(x,y)$ is created with lesser threshold. All nonzero pixels of $g_{NH}(x,y)$ are eliminated from $g_{NL}(x,y)$ by letting

$$g_{NL}(x,y) = g_{NL}(x,y) - g_{NH}(x,y). \tag{5.8}$$

Nonzero pixels in $g_{NH}(x,y)$ and $g_{NL}(x,y)$ are strong and weak edges respectively.

After thresholding, all strong pixels in $g_{NH}(x,y)$ are marked valid as edges. Depending upon value of T_H, there may be gaps in edges in $g_{NH}(x,y)$. Edges with gaps are converted into longer edges using following method.

1. Select and go to next unvisited edge pixel, P in set $g_{NH}(x,y)$
2. Weak pixels in $g_{NH}(x,y)$ which are connected are marked as valid pixels. For this, 8 connectivity criteria can be used.
3. If all nonzero pixels in $g_{NH}(x,y)$ have been visited go to step 4, otherwise return to step 1.
4. Mark all the pixels as zero in $g_{NH}(x,y)$ that were not marked as valid edge pixels.

Final image output of Canny algorithm is formed by appending to $g_{NH}(x,y)$ all the nonzero pixels from $g_{NL}(x,y)$.

5.2 Steps involved in edge extraction using Canny Edge Detector

Canny Edge Detector is used to obtain an efficient representation of the image by locating the edges in it. This representation is highly appropriate as we are interested in the structural information which is very adequately represented through edges. It helps in reducing the amount of data in the image without compromising the structural properties in the hash representation. It possesses all the required properties of a good edge detector, i.e., detection with low error rate, edge point localization and single edge point response as discussed earlier.

The first step in Canny Edge Detection algorithm is smoothening. In this step, the image is smoothened by using a Gaussian filter. This is done in order to remove the noise in the image, such as camera noise which can be mistaken for edge or edges. This is followed by finding the gradient of the edge which measures the sudden gray scale intensity change. Sobel Operator is used to determine the gradient at each pixel location. The gradients in X and Y direction, G_x and G_y respectively are found using following operators:

$$K_{GX} = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \text{ and } K_{GY} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

The resultant edge strength G is given by $G = \sqrt{{G_x}^2 + {G_y}^2}$. Direction of the edge is given by the angle

$$\Theta = arctan\left[\frac{|G_y|}{|G_x|}\right], \tag{5.9}$$

After finding out the magnitude and gradient of the edges, Non-maximum Suppression is carried out to convert the blurred edges in the image to sharp edges. This is achieved by preserving all local maxima and deleting the rest. The output of non-maximum suppression is the edges represented by pixel strength at each pixel. Many of these edges may be true edges and some may be due to noise. To reject these noisy edges, a thresholding mechanism is applied. Canny Edge Detector uses double thresholding to make sure that only the edge pixels stronger than upper threshold are categorized as "strong edges". Edge pixel value lesser than lower threshold are categorized as weak and hence rejected straight away. Hysteresis method of edge tracking is used to track semi-weak edges whose pixel values lie between upper and lower thresholds. It detects semi-weak edges which are connected with strong edges and are retained. The semi-weak edges which are not connected with strong edges are rejected.

5.3 Average Edge Index

The proposed technique uses the output of Canny Edge Detector to generate a hash value for each block in the image. The output which is in the form of $1s$ and $0s$ is summed up and divided by total number of pixels in the block to assign a hash value to the particular block. This index will be called "Average Edge Index" of the block and is denoted by e. As mentioned above, e is defined as:

$$e_{ij} = \frac{\sum_{i=1,j=1}^{i=m,j=m} X_{ij}}{m^2}, \tag{5.10}$$

where X_{ij} is random variable representing values of $(i,j)^{th}$ pixel of Canny Edge Detector output. Value of e_{ij} represents hash value of $(i,j)^{th}$ block. Such $e_{ij}s$ when calculated for all the blocks in the image and arranged according to their position, constitute a hash matrix. Similarity value is calculated as discussed in earlier sections.

This hash matrix is then used for tampering detection and localization. All the rows of the matrix when put together in form of a single row, they constitute hash vector for that image. The hash vector representation is a very appropriate mathematical entity to find out a quantitative measure of the tampering. Hash vector, thus generated is converted into a binary string and can be sent as a header to the person who needs to check the integrity of received image.

After the hash matrices for original and tampered images have been calculated, they are compared using a distance function $\mathcal{D}$ to find out whether tampering has taken place or not and if so, what is the amount of tampering. If the distance function measure is zero then, the image is original one or equivalently, no tampering has been done. If value of $\mathcal{D}$ is non-zero but very small then tampering belongs to category of content preserving manipulation. Sufficiently large value of $\mathcal{D}$ corresponds to malicious tampering. Any efficient detection technique should be able to distinguish between content preserving and malicious tampering clearly. If there is sufficient gap between $\mathcal{D}$ values corresponding to content preserving and malicious manipulation, a threshold can be decided to categorize the images accordingly [116].

The proposed hash generation technique based on computation of Average Edge Index method should be sensitive to very minute structural tampering. It should also be able to ignore content preserving manipulation such as contrast enhancement, low pass filtering, brightness improvement etc. It will be shown that the proposed method is actually robust against some of the content preserving manipulations. It is very much expected because only the edges in the image which represents high value of gradient, are being considered. It may be recalled that Canny Edge Detector works on double threshold principle where edges due to noise are rejected. Also the edges which are not connected with strong edges and amount to spurious edges are rejected by the detector. It was also mentioned that first step in Canny Edge Detection is suppression of noise using a Gaussian filter. This filtering removes noise significantly before application of Sobel operator. The above characteristics of Canny Edge Detector makes it superior to other existing techniques with regard to robustness against content preserving manipulation.

5.4 Algorithm for tampering detection using Canny Edge Detector

Method used to compute the hash values and hash matrices is same as used in chapter 2. In this case, average edge index is taken as the hash value. Rest of the method remains same.

5.4.1 Sensitivity

Mathematical analysis was done by finding average edge index of original and tampered images denoted by e_o and e_t for 30 images using eqn. (5.10). Indices e_o and e_t were plotted as shown in fig. (5.1).

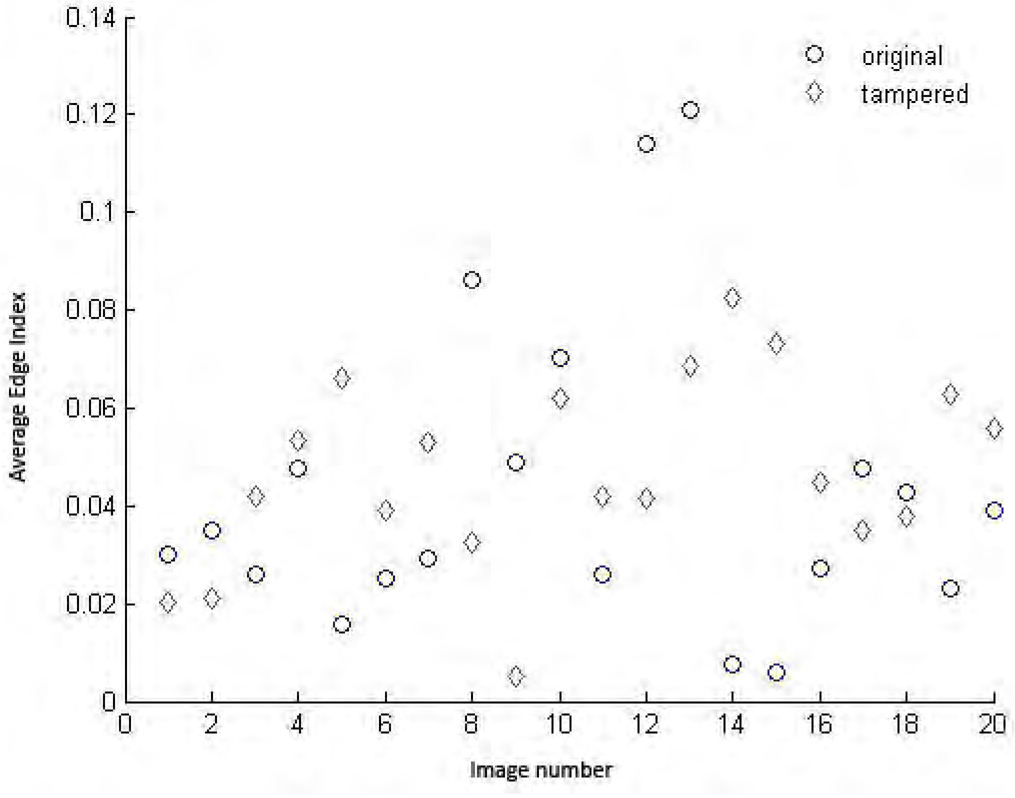


Figure 5.1: Graph showing the sensitivity of Canny Edge Detector.

The graph shows that e_o and e_t are sufficiently separated which enables us to distinguish between tampered and original images. It enables us to choose a threshold value of e somewhere between e_o and e_t sufficiently away from both to distinguish between tampered and original images. This separation is sufficiently noticeable even for minute tampering implying that method is highly sensitive. The graph in fig. (5.1) suggests that a threshold value of e can be selected which will categorize original and tampered images.

5.4.2 Robustness against content preserving manipulations

The proposed method is inherently robust against content preserving manipulations such as low pass filtering, change in average brightness level, contrast change and JPEG compression etc.

5.5 Tampering detection using Canny Edge Detector

Y component of $YCrCb$ representation of an image is used for experiments as only Y contains the structural information. For the time being, square images of pixel size $M \times M$ will be discussed. The proposed method comprises of following steps:

1. The image under consideration is divided into a number of blocks. The block size is so chosen that M is integral multiple of side of the block.

2. Edges in each block are detected using Canny Edge Detector giving $1s$ where the edge exists and $0s$ where it does not.

3. Average Edge Index of the block is calculated by summing all the $1'$s present and dividing it by total number of pixels in the block. This index forms the hash value for the block under consideration.

4. Hash values for different blocks are then arranged at their respective locations to give hash matrix for the image.

5. The above process is done for original as well as the tampered image.

6. The absolute difference of hash matrices corresponding to original and tampered images is computed which gives tampering area as shown in the flowchart illustrated in fig. (2.1).

The size of sampling block is equal to the size of the block in which we wish to divide the image. In the present case, sampling size was taken to be 50×50. As mentioned above, a single hash value (average edge index) is assigned to a block, any tampering of size lesser than 50×50 will be shown over full block size. This is not a problem if we want to detect the existence of tampering only and where accuracy in localization of tampered area is not a concern. But if the tampering area is less than 50×50 and we want to locate it more accurately then the sampling block size should be reduced accordingly.

Mathematical analysis was done by finding average edge index of original and tampered images denoted by e_o and e_t for 30 images using eqn. (5.10). It is observed that e_o and e_t are sufficiently separated such that it enables us to classify the image as tampered or non-tampered. This also shows that the algorithm is sensitive.

5.6 Experiments and Results

To show the robustness of Canny Edge Detector, 30 images of 50×50 sizes were taken. For this average edge index of original images e_o was found out using Canny Edge Detector. Same set of images was then low pass filtered (blurred) and average edge indices e_b were calculated by detecting the edges and using eqn. (5.10). The brightness level of these images was changed and its effect on average edge index was observed. The graph (fig. 5.2) shows that blurring of the images and brightness change only marginally affect average edge index value of the images. The robustness of the proposed method can also be proved for content preserving manipulations such as change in average brightness level and minor contrast changes which will be shown experimentally.

5.6.1 Detection and localization of tampering

Experiment for tampering detection was done on a database of 100 images of 400×400 size. Edges in the image were picked up using Canny Edge Detector and Average Edge Indices were calculated and arranged to get the hash matrices. Tampering area was found out by computing $|H_o - H_t|$. First, tampering detection was done by using sampling block size of 50×50 and same was repeated by changing block size to 20×20. It was found that accuracy of detection improves with reducing the block size which is shown in Table (5.1).

To study the variation in Similarity Value S, a set of 30 images was taken. It was observed that S changes with variation in spatial and structural tampering. As the amount of tampering increases, value of S moves away from 1 towards 0. Result has been shown in Table (5.2).

Robustness of algorithm against content preserving manipulation was tested for 30 images. Two types of such manipulations (a) change in brightness level and (b) blurring using low pass filter were considered for this purpose. Brightness level of the images was changed by $+20$ and -20 and its effect on average edge index was observed. It was found that there was no significant change in the value of e proving the robustness against brightness change. Blurring operation was done using Gaussian filter ($\sigma = 0.3$) and its effect on e was studied. Again, it was found that there was no significant affect on value of average edge index. The results are plotted in fig. (5.2).

5.6.2 Collision probability test

Low collision probability is an important requirement for any good hash generation technique. Collision probability is defined as probability of two different images mapping to same hash values [117]. As discussed in the sub-section (4.5.2), there should be a very strict one to one correspondence between image and its hash. Ideally, no two images should provide same hash value. Mathematically, this requirement has been described in eqn. (4.6). To conduct this test for Canny Edge Detector, $62,000$ pairs of images was taken and similarity value histogram was plotted as shown in fig. (5.3).

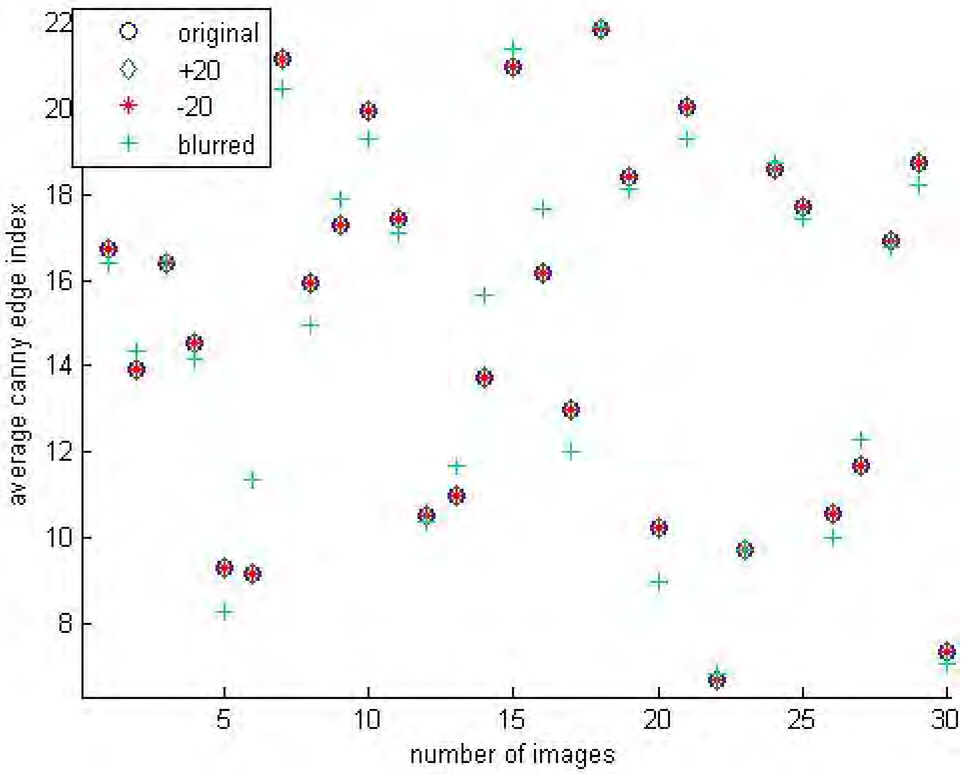


Figure 5.2: Average edge indices for 30 images have been plotted in this graph for the original image, with brightness level changed by +20 and −20 and after low pass filtering (blurred) for showing the effects of content preserving manipulations. It is evident that there is no appreciable change in these average edge indices for any image.

This plot resembles a Gamma distribution. To check the distribution of the data, Chi Square Test was carried out to fit the data using eqn. (4.14). From the Similarity Value data, α and p-parameters for gamma distribution were calculated. Chi square test showed that the data fits Gamma distribution. We get the collision probability as $P[S \geq 0.8] = 2.68 \times 10^{-12}$. Here it has been assumed that hash values for two images are treated as colliding if their similarity value is greater than 0.8. We observe that probability that any two images will be similar is extremely low and therefore method using Canny Edge Detector to generate hash representation of the image satisfies this requirement.

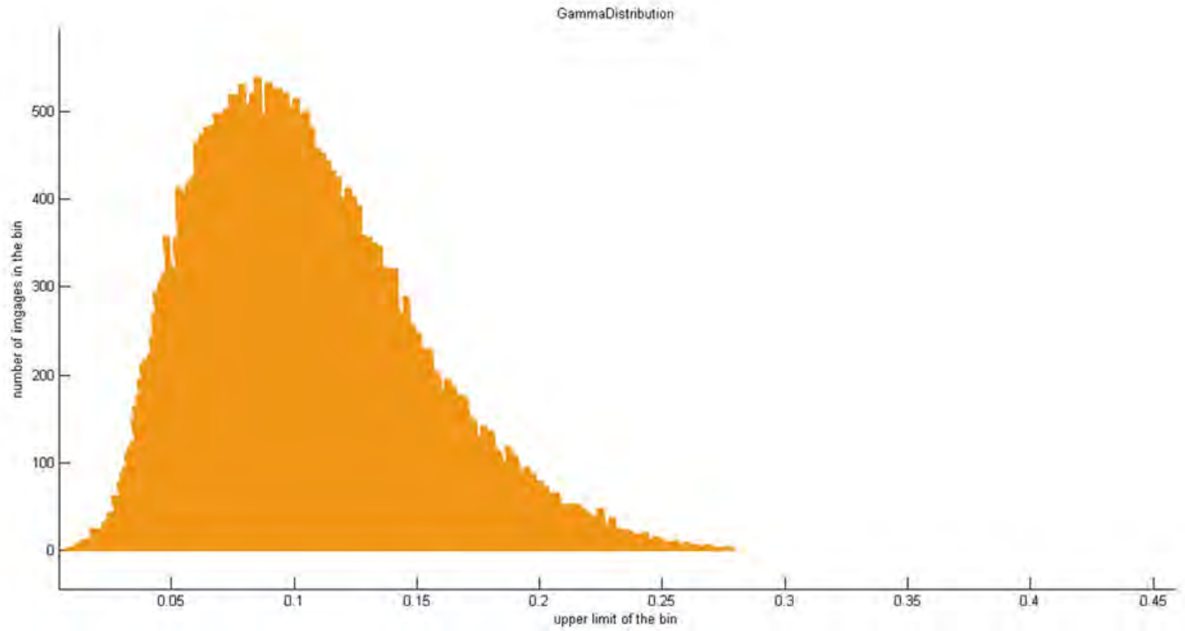


Figure 5.3: Histogram plotted of Similarity values of $62,000$ pairs of images showing a gaussian distribution.

5.7 Conclusion

In earlier sections of this chapter, the theoretical justification for using Canny Edge Detector was discussed from the view of sensitivity, robustness and collision probability. The experiments conducted showed that Canny edge detector, indeed, proves to be a very efficient image feature extractor. Average edge index was defined and used for the first time in this field and it acts as a high quality hash function that provided excellent results with regard to sensitivity and robustness. These results were along expected lines considering various properties of the detector. Collision probability was found to be marginally higher than SVD technique but still good enough for our purpose. In SVD based technique, collision probability was found to be 3.95×10^{-13} whereas in Canny Edge Detector case it is 2.68×10^{-12}. Canny Edge Detector method is particularly suited to detect very minute tampering operations which are not easy to detect and may exist in noisy environment.

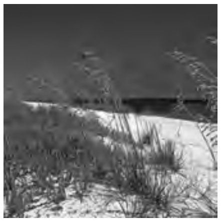
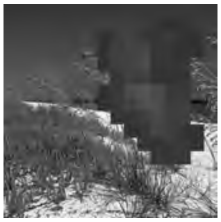
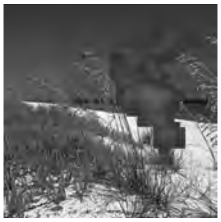
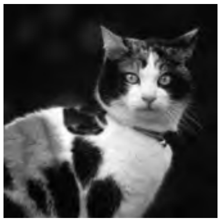
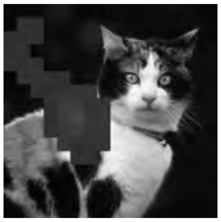
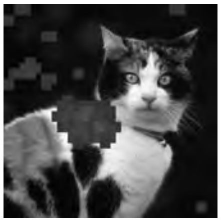
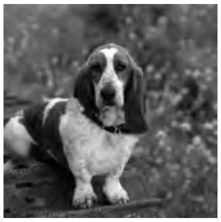
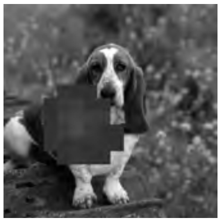
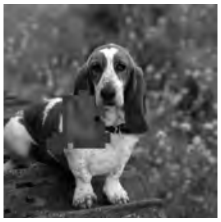
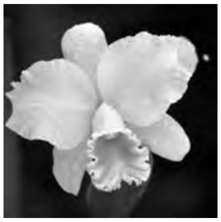
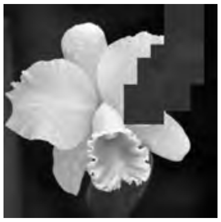
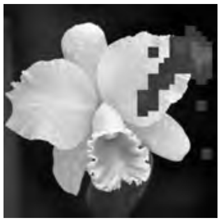
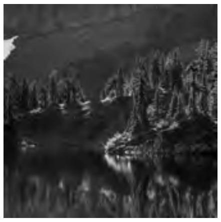
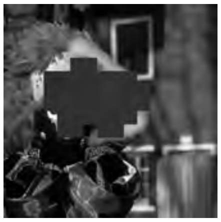
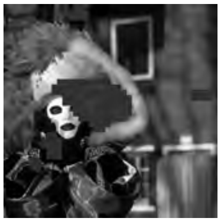
No.	Original Image	Tampered Image	Localization of tampering using 50×50 block	Localization of tampering using 20×20 block
1				
2				
3				
4				
5				

Table 5.1: Accuracy in localization of tampering in the images is achieved by reducing the sampling block size. This has been shown using comparison between 50x50 and 20x20 sampling blocks size.

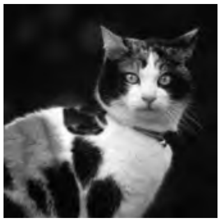
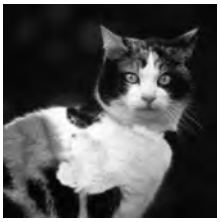
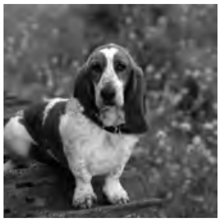
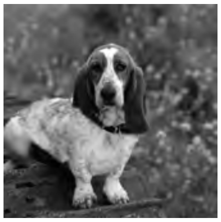
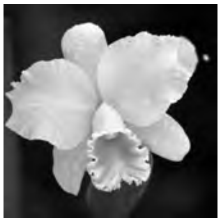
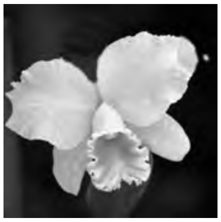
No.	Original Image	Image with small tampering	Image with large tampering	Similarity value for small tampered area	Similarity value for large tampered area
1				0.500975	0.449958
2				0.703281	0.459967
3				0.649599	0.41979
4				0.791836	0.344108
5				0.883379	0.417613

Table 5.2: The similarity value changes according to the area of tampering in the images. It decreases with increase in tampering.

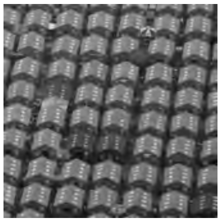
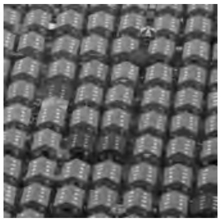
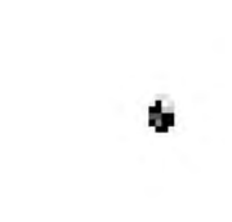
No.	Original Image	Tampered Image	Localization of tampering using 50 × 50 block	Localization of tampering using 25 × 25 block
13				
14				
15				
16				
17				
18				

Table 5.3: Accuracy in localization of tampering in the images is achieved by reducing the sampling block size.

CHAPTER 6

Comprehensive Image Index for Detection of Multiple Tampering Using 3-tupled Hash Function

Most of the existing hashing techniques extract one single feature of the image and develop single index hash function for image tampering detection. However, single feature tampering is very rare and any tampering operation results in multiple tampering. Proposed technique uses a 3-tupled Comprehensive Image Index (CII) which incorporates indices corresponding to structural tampering, brightness level tampering and contrast manipulations. CII enables us to detect existence or otherwise of the three tampering operations mentioned above simultaneously.

6.1 Comprehensive Image Index (CII)

Concept of this work was taken from paper titled "Universal Image Quality Index" (UIQI) by Zhou Wang and Alan C Bovik [118]. The authors defined a mathematical index to express the quality of an image and which is independent of viewing conditions and individual observer. This index has lower mathematical computational requirement and does not change with viewing conditions and observer and therefore it is termed as "universal".

UIQI is defined as below:
If $x = \{x_i | i = 1, 2, ..., N\}$ and $y = \{y_i | i = 1, 2, ..., N\}$ be image under consideration

and reference image respectively, UIQI is defined as

$$Q = \frac{Cov(X,Y)}{\sigma_x \sigma_y} \frac{2\bar{x}\bar{y}}{\bar{x}^2 + \bar{y}^2} \frac{2\sigma_x \sigma_y}{{\sigma_x}^2 + {\sigma_y}^2}, \tag{6.1}$$

where,

$$\bar{x} = \frac{1}{N}\Sigma_{i=1}^{i=N} x_i \tag{6.2}$$

$$\bar{y} = \frac{1}{N}\Sigma_{i=1}^{i=N} y_i \tag{6.3}$$

$$\sigma_x^2 = \frac{1}{N-1}\Sigma_{i=1}^{i=N} (x_i - \bar{x})^2 \tag{6.4}$$

$$\sigma_y^2 = \frac{1}{N-1}\Sigma_{i=1}^{i=N} (y_i - \bar{y})^2 \tag{6.5}$$

$$Cov(X,Y) = \frac{1}{N-1}\Sigma_{i=1}^{i=N} (x_i - \bar{x})(y_i - \bar{y}). \tag{6.6}$$

First component in the equation for Q is correlation coefficient and varies between -1 and $+1$. It has been shown in chapter 3 that correlation coefficient represents the structural content of the image. The second component is calculated using mean pixel value and hence represents the brightness level of the image. The third component is calculated using variance (i.e., variation from mean brightness value) and hence represents the contrast of the image. These three components together give almost the complete information about the image which can be relevant to image tampering detection. It has been discussed later in this chapter that how these three components are complete in describing the image. As an image can be primarily described by structure, brightness level and contrast, the three components of UIQI mentioned above can be used as representative parameters of the image.

An image can primarily be described by the ingredients (a) Structure, (b) Brightness level and (c) Contrast of the image. UIQI as defined above includes all these ingredients. Hence, UIQI can be used as representative which describes the image comprehensibly. We will relate correlation coefficient and structure of the image through a linear equation and call it structural index. If we represent above three ingredients through appropriate indices η_1, η_2 and η_3 respectively then Comprehensive Image Index (CII) can be represented as $\eta = (\eta_1, \eta_2, \eta_3)$. It is evident that η_1 represents Structural Index, η_2 represent Brightness Level Index and similarly, η_3 represents Contrast Index of the image.

6.1.1 Structural Index, η_1

Structural content of the image is represented through edges present in the image to a large extent [119]. Tampering in an image is basically alternation, removal or insertion of edge/edges fully or partially. It is found that edge are very appropriate and complete description of an image. Y component of $YCrCb$ description of the image will be used to extract the edge features of the image as Y only contains the structural information. An edge detector will be used to detect the edges and its output will be used to calculate image's structural index.

In proposed technique, a gradient filter has been used. It measures rate of change of pixel values in X and Y directions which can be represented by I_x and I_y respectively. Thus

$$I_x = \frac{dI}{dx}, \tag{6.7}$$

and

$$I_y = \frac{dI}{dy}. \tag{6.8}$$

The resultant rate of change of pixel values at any point (x,y) is therefore given by

$$I_{xy} = \sqrt{(I_x^2 + I_y^2)}. \tag{6.9}$$

As value of I_{xy} represents features of an image, it will be used to generate hash value. Correlation coefficient ρ between two images A and B is defined as

$$\rho = \frac{Cov(X, Y)}{\sigma_x \sigma_y}, \tag{6.10}$$

where X and Y are random variables expressing pixel values in images A and B respectively.

It should be noted that ρ varies between -1 to $+1$ and should be linearly transformed to get positive values of structural index η_1. It is done by using the following linear equation:

$$\eta_1 = \rho + 1. \tag{6.11}$$

6.1.2 Brightness Level Index, η_2

Many a times, change in the brightness level of image is carried out with bona fide intention of improving its visual quality. However, it may be changed to hide some information in the image or cover up a tampering. Such brightness change is generally global or locally global. An index to describe brightness level for purpose of tampering detection is denoted by η_2. It is defined with respect to same reference image as mentioned in case of η_1 as:

$$\eta_2 = \frac{2\bar{x}\bar{y}}{\bar{x}^2 + \bar{y}^2}, \tag{6.12}$$

where $\bar{x}$ and $\bar{y}$ are mean brightness level of image A and B respectively. It can be noted that η_2 varies between 0 and 1. It is also observed that for small change in the mean brightness level $\bar{x}$, index η_2 changes very marginally. But for higher value of change, there may be noticeable change in η_2.

6.1.3 Contrast Index, η_3

Contrast enhancement is a popular image processing technique which is used for bona fide improvement of contrast of the image. But in certain situations contrast enhancement may be done to hide some tampered portions in the image. Contrast enhancement is mathematically expressed as

$$P = CI^{\gamma}, \tag{6.13}$$

where, $I =$ Pixel value of original image, $P =$ Pixel value of altered image, γ is the index which decides level of enhancement and C is a constant.

Image Contrast Index η_3 gives range of maximum variation from mean brightness value and decides the contrast in the image. As expected it is expressed through standard deviation σ. It is being measured with respect to same reference image as used in case of other two indices and is given by

$$\eta_3 = \frac{2\sigma_x \sigma_y}{{\sigma_x}^2 + {\sigma_y}^2} \tag{6.14}$$

where σ_x and σ_y are standard deviation for images A and B respectively. It is seen that (a) η_3 varies between 0 and 1 and (b) for small change in σ_x there is very minimal change in η_3 for a given reference image with standard deviation σ_y. However for major change in contrast (say for $\gamma = 2$ in eqn. (6.13)), there may be significant change in η_3.

Low value of change in mean brightness level has very minimal effect on η_2. Similarly, low value of change in contrast ($\gamma = 1.2$ say) does not have any significant bearing on η_2. This fact shows that for small change in mean brightness level and/or contrast change, which amount to content preserving manipulations, proposed method is robust. In cases where change in brightness level and/or contrast level of the image is significant, the corresponding changes in η_2 and η_3 will be large which amounts to malicious tampering. This also suggests that these indices should prompt a threshold level for η_2 and η_3 which will be categorized as content preserving manipulations and malicious tampering separately.

6.1.4 Definition of Comprehensive Image Index

Comprehensive Image Index, η, is represented as

$$\eta = (\eta_1, \eta_2, \eta_3), \tag{6.15}$$

where η_1 is structural index, η_2 is mean brightness index and η_3 is contrast index. These three indices define the image comprehensively. The justification for selection of these parameters to describe the features of the image is discussed below.

An image can be described using (a) shape descriptors (b) texture descriptors and (c) colour descriptors. In present case, we are only concerned with structural tampering detection therefore only the gray scale image will be considered. Hence colour descriptor is ignored and only gray scale pixel values will be taken into account. Shape of an image is described by the edges present in it. Exterior as well as interior shape of the image can be effectively picked up through its edges. This is done using a gradient detector and its output generates the hash value representing the shape descriptors.

The textured descriptors are contained in pixel values of the image. This information can be described using contrast and brightness values of the pixels. Thus, hash function containing information about edges, contrast and brightness value gives complete description of the image. Therefore selection of comprehensive image index incorporating above three factors has been chosen for purpose of composite structural tampering detection.

For structural tampering detection, Edge Index η_1 is suitable and η_2 and η_3 are not very significant. However, if a structural tampering is carried out in conjunction with contrast and brightness manipulations to hide it, all the three indices η_1, η_2, and η_3 become relevant.

The three features represented through η_1, η_2, and η_3 give almost complete description of image. However, the three features are not orthogonal. But orthogonality is not really so much a required property for tampering detection. Interdependence of three variables to some extent does not, in any way, limit our capacity to draw conclusion about type and amount of tampering an image.

In view of the above, a more generalized technique is proposed which will generate a 3-tupled hash value each representing three indices η_1, η_2 and η_3 together. It will be represented as (η_1, η_2, η_3). In order to detect the tampered area in an image, the image will be suitably divided into number of blocks. 3-tupled hash value corresponding to each block, i.e., $(h_1, h_2, h_3)_{ij}$ where (i, j) represent the location of the block in the image, will be found out. These hash values when arranged at their respective block locations, will constitute the hash matrix H of the image under consideration. When rows of the matrix are arranged in a single row, they form a hash vector which is a comprehensive representation of the image. It is converted into a binary string and may be provided to the receiver either along with the image or separately which can be used to check the authenticity of the image.

After computation of hash matrices, a distance function $\mathcal{D}$ is defined to compare the distance between the original and tampered images. Value of $\mathcal{D}$ tells us if the tampering in the image has taken place or not. If the distance is zero then there is no tampering in the image. If distance is non-zero but very small then tampering is of content preserving type but if the distance is substantial then tampering belongs to malicious tampering category. In proposed method distance function chosen is Euclidian distance and this is computed by finding out the absolute difference between hash matrices of original and tampered image.

6.2 Robustness of CII against content preserving manipulation (CPM)

When talking about robustness against CPM, it is always desirable to quantify the threshold level up to which a particular technique is robust. The efficiency of technique lies in ignoring content preserving manipulation at the same time detecting even the minutest tampering operations. In the proposed method, 3-tupled hash value generated for the image will include indices for brightness level change and contrast variation along with structural index. This will give an exact quantification of the manipulation with respect to three different parameters. For ensuring robustness against brightness and contrast change parameters η_2 and η_3 are so defined that small changes in brightness and contrast respectively do not alter their values much. Chosen formulation for η_2 and η_3 also ensures that effect of changes in η_2 and η_3 on η_1 is very minimal. Hence detection of structural tampering through CII is robust against CPMs.

From hash matrix H, three hash matrices each corresponding to indices η_1, η_2 and η_3 respectively, can be generated. Set of these three matrices are computed for original image and the suspected image for tampering detection. Absolute difference of respective matrices $|H_o - H_t|$ corresponding to the three types of tampering operations, i.e., structural tampering, change in brightness level and change in contrast level, is then computed to find out the amount and area of tampering.

6.3 Proposed algorithm for multiple tampering detection through CII: 3-tupled hash vector representation

Structural information in an image is described by the edges in it. Therefore, any tampering in the structure of the image will be affecting the position and amount of the edges. Here, an edge detector, as described in eqns. (6.7), (6.8) and (6.9) which is primarily a gradient filter, will be used to extract the edges and develop an index for the same. Normally, a colour image is described through YCrCb representation but only Y component contains the structural information and will be used for our purpose. An $M \times M$ image will be used in the proposed method for image tampering detection. The algorithm comprises of following steps:

1. $M \times M$ image is divided into blocks of size $q \times q$. Here, q is chosen such that M is an integral multiple of q.

2. Edge detector is applied to the blocks to extract edges in it. It is done by using a derivative filter and actual pixel value gradient is computed for all pixels in the image. The output of edge detector is a matrix of $q \times q$ size and it is used to find out correlation coefficient of the block with a reference image of the same size.

3. Value of correlation coefficient ρ varies between -1 to $+1$ and it is suitably scaled using a linear equation to avoid negative values. The value for the block so calculated, gives its hash value.

4. These hash values are arranged at their respective block location to give hash matrix for the image corresponding to structural index η_1.

5. Following the block generation method mentioned above, hash matrices corresponding to brightness level and contrast index are also found out. The reference image used in all the three cases is same.

6. Above process gives a hash value $[h_{ij}(\eta_1), h_{ij}(\eta_2), h_{ij}(\eta_3)]$ representing hash values for all the three indices for the block (i, j).

7. Steps 1 to 8 are carried out for original and tampered image [116].

Algorithm described in sub-section 2.2 along with eqns. (6.10, 6.11, 6.12, 6.14) are used to find 3-tupled CII for each block in the image. 8-neighbourhood sum is used to find corresponding hash value given by the eqn. (6.16). This process is carried out for original and tampered images.

$$h_{ij} = \eta_{i-1,j} + \eta_{i,j} + \eta_{i+1,j} + \eta_{i,j+1} + \eta_{i,j-1} + \eta_{i-1,j+1} + \eta_{i+1,j+1} + \eta_{i-1,j-1} + \eta_{i+1,j-1}, \tag{6.16}$$

where $i, j = 1, 2, 3, ..., t$.

$$\begin{pmatrix} h_{11}(\eta_1), h_{11}(\eta_2), h_{11}(\eta_3) & h_{12}(\eta_1), h_{12}(\eta_2), h_{12}(\eta_3) & ... & h_{1t}(\eta_1), h_{1t}(\eta_2), h_{1t}(\eta_3) \\ h_{21}(\eta_1), h_{21}(\eta_2), h_{21}(\eta_3) & h_{22}(\eta_1), h_{22}(\eta_2), h_{22}(\eta_3) & ... & h_{2t}(\eta_1), h_{2t}(\eta_2), h_{2t}(\eta_3) \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ h_{t1}(\eta_1), h_{t1}(\eta_2), h_{t1}(\eta_3) & h_{t2}(\eta_1), h_{t2}(\eta_2), h_{t2}(\eta_3) & ... & h_{tt}(\eta_1), h_{tt}(\eta_2), h_{tt}(\eta_3) \end{pmatrix},$$

Experiment was carried out on a set of 100 images. Indices η_1, η_2 and η_3 were obtained for original as well as tampered images. The graph in fig. (6.3) shows that η_1 responds to structural tampering. It can be observed that structural index for original and tampered images, i.e., $\eta_{1(o)}$ and $\eta_{1(t)}$ are separated sufficiently

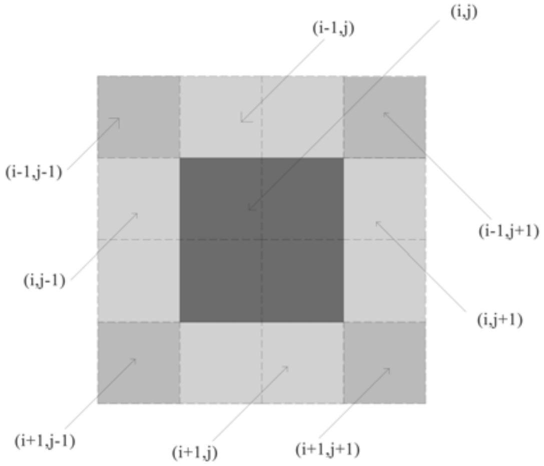


Figure 6.1: The central block is at i^{th} row and j^{th} column of the blocks of the image. The four neighbouring blocks are obtained shifting by half block width to the left (i,j-1), right (i,j+1), up (i-1,j), down (i+1,j) and four corner blocks at (i-1,j-1), (i+1,j+1), (i-1,j+1) and (i+1,j-1).

and can be distinguished easily. It implies that detection method is sensitive in locating the tampering in an image.

The other two indices, i.e., brightness level index and contrast index are carefully defined so that they respond to small variations very marginally. Small change in brightness level of image, changes η_2 value insignificantly. Similarly, small change in contrast ($\gamma = 1.2, 0.9$ in gamma correction equation) does not alter η_3 much and can be easily ignored. Nevertheless, this small change in η_2 and η_3 is listed in the hash value and consequently in hash matrix. For higher value of change in brightness level and contrast, η_2 and η_3 change noticeably while doing the tampering detection. Absolute difference between hash matrices of original and tampered image corresponding to three indices η_1, η_2 and η_3, is found out. Let the difference be given by distance function $\mathcal{D}$. Then,

$$D = |H_t - H_o|. \tag{6.17}$$

In component form

$$D_{(1)} = |H_t(\eta_1) - H_o(\eta_1)|, \tag{6.18}$$

$$D_{(2)} = |H_t(\eta_2) - H_o(\eta_2)|, \tag{6.19}$$

$$D_{(3)} = |H_t(\eta_3) - H_o(\eta_3)|, \tag{6.20}$$

where $\mathcal{D}_{(1)}$ is absolute difference between hash matrices of tampered and original image corresponding to structural index η_1. Similarly, $\mathcal{D}_{(2)}$ and $\mathcal{D}_{(3)}$ are absolute differences corresponding to η_2 and η_3.

Three components hash calculations serve following purposes:

1. It helps in deciding if the tampering is structural in nature or occurred due to brightness level change and/or contrast change.

2. Low values of $\mathcal{D}_{(2)}$ and $\mathcal{D}_{(3)}$ amount to content preserving manipulations and we can ignore them by setting a threshold $\mathcal{D}_T$.

3. If value of $\mathcal{D}_{(2)}$ and $\mathcal{D}_{(3)}$ is large then we can conclude that tampering done through brightness level change or contrast variation, has been done with malicious intention. It is interesting to note that if $\mathcal{D}_{(1)}$ and $\mathcal{D}_{(3)}$ both are large then contrast change could have been carried out to cover structural tampering. Similarly conclusions can be drawn with other combinations of $\mathcal{D}_{(1)}$, $\mathcal{D}_{(2)}$, and $\mathcal{D}_{(3)}$. If $\mathcal{D}_{(1)}$ and $\mathcal{D}_{(2)}$ both are large, it could mean that structural tampering has been done followed by brightness change to cover it. If only $\mathcal{D}_{(2)}$, and $\mathcal{D}_{(3)}$ are large with no change in $\mathcal{D}_{(1)}$, it may mean that purpose of tampering is to improve the visual quality of the image without any malicious intention.

6.4 Independence of indices η_1, η_2 and η_3

Image processing operations associated with η_1, η_2 and η_3 are not independent. For example, structural tampering affects the pixel values which change mean brightness level resulting into change in η_2. On the other hand, contrast enhancement changes the pixel values in accordance with $P = CI^{\gamma}$, resulting into change in η_1 and η_3. Similarly, drastic change in mean brightness level may suppress relatively weak edges and thus affecting η_1. It will suffice to say that η_1, η_2 and η_3 are not independent and affect each other in general. However, in our analysis, final combined effect of various tampering operations is taken into account to calculate η_1, η_2 and η_3.

To demonstrate the performance of proposed algorithm in conjunction with an edge detector, a set of 100 images of 40×40 is taken. For calculating η_1, the edges in the image are extracted using gradient edge detector. The correlation coefficient ρ of output matrix with reference image is found out which in turn gives η_1. Indices η_2 and η_3 are calculated before application of edge detector using formulae shown in eqns. (6.12) and (6.14) respectively. Above process is repeated for same set of images but after tampering them. Index η_1 is plotted for different amount of structural tampering which is shown in fig. (6.3). Similarly, η_2 is plotted for different values of brightness level and η_3 is plotted for different values of gamma correction. It is observed that for low level of these tampering operations, η_2 and η_3 change very marginally. Also minor changes in brightness level and contrast does not have any noticeable impact on η_1. However, as the amount of tampering increases, the respective indices move away significantly from those of the original image. This observation demonstrates the robustness of proposed algorithm as well as the sensitivity of detection for all three types of tampering operations.

Extraction of edges using gradient filter ignores noise as it is basically low frequency signal. The output of gradient filter is zero to these low frequency components. This quality of gradient filter makes it robust against low frequency brightness and contrast manipulations.

6.5 Similarity Value Vector

Having done the tampering detection, it is desirable to develop an index to quantify amount of tampering in an image. For this, Similarity Value Vector will be defined as follows. Rows of $t \times t$ hash matrix are arranged one by one in a single row to generate hash vector $\mathbf{H}$ of the image. For two images A and B with their respective hash vectors $\mathbf{H}^a$ and $\mathbf{H}^b$, a ratio R is defined as

$$R_i = \frac{exp\left[min\left(\mathbf{H}_i^a, \mathbf{H}_i^b\right)\right]}{exp\left[max\left(\mathbf{H}_i^a, \mathbf{H}_i^b\right)\right]}, \tag{6.21}$$

where $i = 1, 2, 3, ..., t^2$. R_i assumes a value 1 if ith component of two hash vectors $\mathbf{H}_i^a$ and $\mathbf{H}_i^b$ have same value implying that there is no tampering in that block. Similarity Value for images A and B is defined as

$$S\left(\mathbf{H}^a, \mathbf{H}^b\right) = \frac{\prod_{R_i \in R_S} R_i}{\prod_{R_i \in R_L} R_i}. \tag{6.22}$$

Numerator in eqn. (6.22) is product of m minimum most R_i ratios, denominator being product of m maximum most R_i ratios. For accurate calculation of S, number m should be more than or equal to the number of tampered blocks. An iterative method can be used to arrive at an appropriate value of m. It is observed that S varies between 1 and 0. If two images are same, their S value is 1 and it moves away towards 0 when amount of tampering is increased. As there are three types of tampering operations represented through η_1, η_2 and η_3, there will be 3-tupled Similarity Value Vector (S_1, S_2, S_3) for a pair of two images.

By virtue of having a 3-tupled Similarity Value Vector, proposed technique becomes very comprehensive and is superior to many other existing techniques which normally work around single parameter detection. The technique not only ignores content preserving manipulations easily but also responds adequately to any manipulation crossing into malicious tampering zone by detecting and quantifying it properly through Similarity Value Vector.

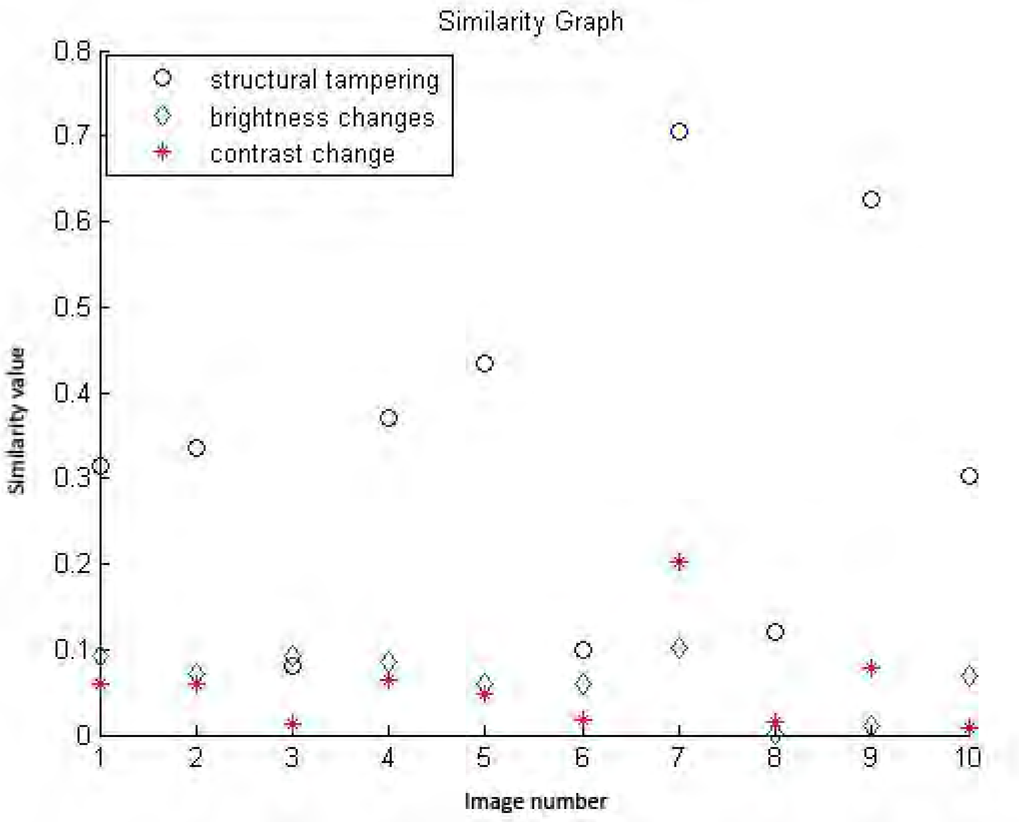


Figure 6.2: Similarity Value Vector for set of 10 original and tampered images.

6.6 Experiments and Results

A set of 100 images was taken and proposed algorithm was used to calculate indices η_1, η_2 and η_3. Hash matrices $H(\eta_1)$, $H(\eta_2)$ and $H(\eta_3)$ were calculated for original and tampered images. Absolute distances $|H_t(\eta_1) - H_o(\eta_1)|$, $|H_t(\eta_2) - H_o(\eta_2)|$ and $|H_t(\eta_3) - H_o(\eta_3)|$ were computed to detect all three types of tampering separately as shown in Table (6.2). Robustness and sensitivity of the technique was tested for a set of 20 images as shown in fig. (6.3).

Robustness was tested for set of 30 images. For detection of brightness level change, experiment was conducted by altering pixel values by $+20, -20, +90$ and its effect on η_2 was observed. It was found that for small brightness changes, there is almost no change in η_2 but when brightness level change becomes $+90$, η_2 changes significantly as shown in fig. (6.4). Effect of change in contrast index γ on η_3 was studied by changing γ values to $0.2, 1.5$ and 5.0. Again for small change in

No.	Original Image	Tampered Image	S_1 value	S_2 value	S_3 value
1			0.31445	0.09140	0.06012
2			0.33450	0.07108	0.05912
3			0.43453	0.06049	0.04828

Table 6.1: Similarity Value Vectors (S_1, S_2, S_3) have been shown for three images. (S_1, S_2, S_3) speaks about similarity value for structural, brightness level and contrast tampering in the image simultaneously.

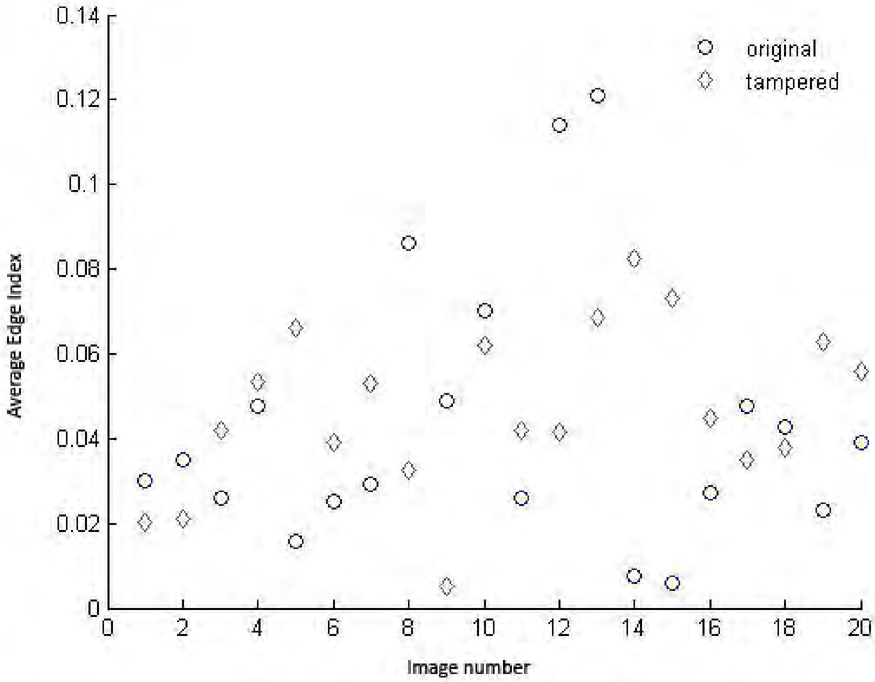


Figure 6.3: Average Structural indices for 20 images have been plotted in this graph for the original image and tampered image showing the effect of structural tampering on η_1.

γ, there was no significant change in η_3. However, change in η_3 became noticeable when γ value was changed drastically. These are shown in fig. (6.5).

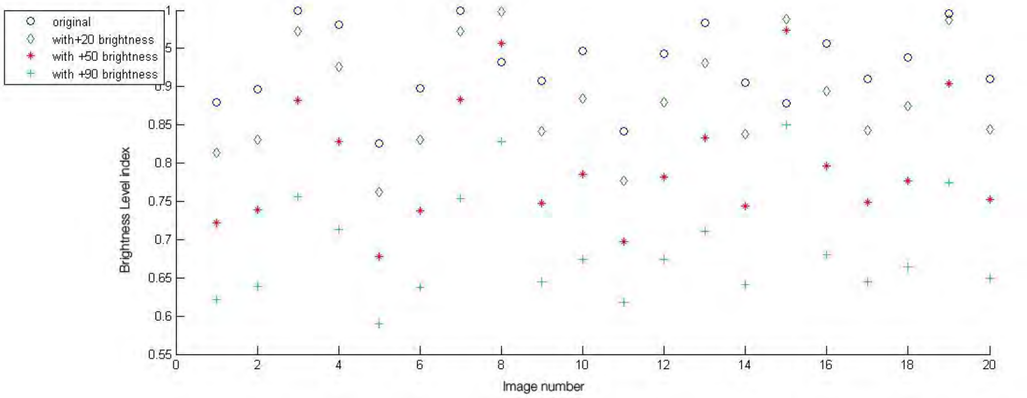


Figure 6.4: Brightness Level indices for 20 images have been plotted in this graph for the original image, with change in brightness level by +20, +50, and +90 for showing effect of change in brightness level on η_2.

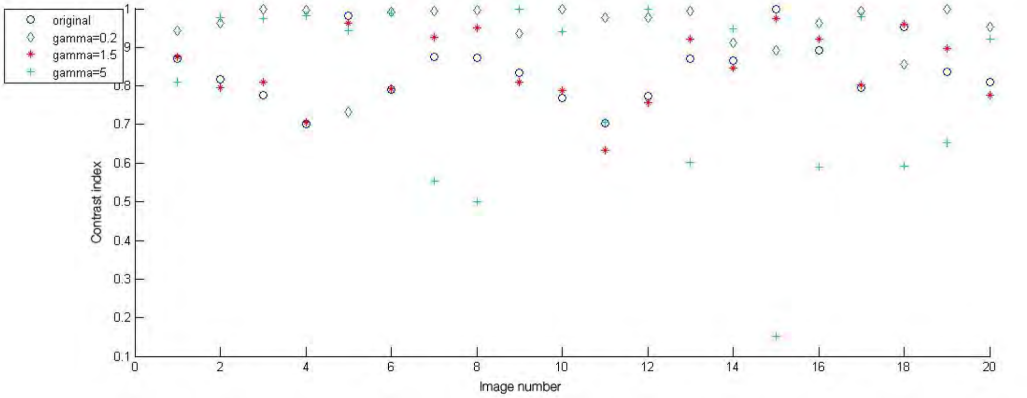


Figure 6.5: Contrast indices for 20 images have been plotted in this graph for the original image, with γ changed to 1.5, 0.2 and 5.0 for showing effect of γ change on η_3.

6.7 Conclusion

The aim of proposed content based hashing technique was to offer a comprehensive method to detect multiple image tampering. This was done through generation of 3-tupled hash functions which incorporates indices for structural, brightness level and contrast tampering. For the first time, tampering detection was carried out around three different parameters and it worked satisfactorily. Issue of robustness and sensitivity was dealt with in quantitative terms. Ingress of content preserving manipulation into malicious tampering zone in case of change in brightness/contrast level drastically, was handled through a 3-tupled hash vectors. This concept makes the proposed technique comprehensive which can detect multiple tampering operations simultaneously.

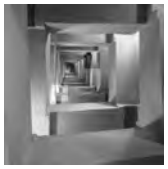
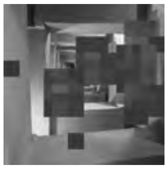
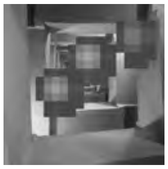
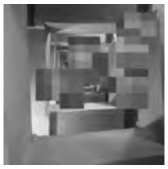
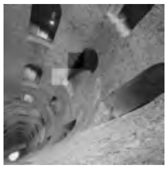
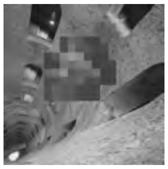
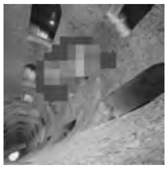
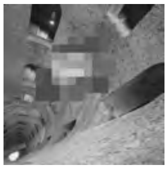
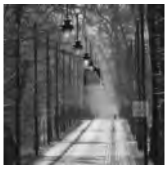
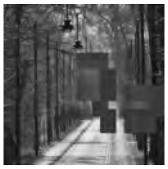
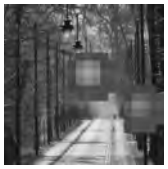
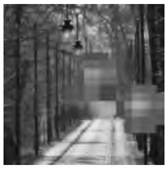
No.	Original Image	Tampered Image	Localization of tampering using η_1	Localization of tampering using η_2	Localization of tampering using η_3
1					
2					
3					

Table 6.2: Simultaneous detection of structural, brightness level and contrast tampering has been shown for three images through generation of 3-tupled hash matrix.

CHAPTER 7

Our Contributions

In the following part we would like to highlight the salient contributions that we made through this study.

7.1 Accuracy of tampering detection and localization

Most of the earlier methods aimed that detection of tampering based on feature based hash generation. Precise localization of tampering is distinct advantage of algorithm used by us.

As mentioned in chapter 3, the sampling block which divides the image horizontally and vertically into a matrix form, is the fundamental area unit over which tampering localization is achieved. The absolute difference of the hash matrices corresponding to original and tampered images shows the area over one or more number of blocks. As a single hash value is assigned to each block a tampered area which is even lesser than a block size, will be shown over one full block. In such a case, if we reduce the sampling block size, we can narrow down the detected tampering region which is closer to actual tampered area. An iterative algorithm can be developed which automatically adjusts the sampling block size with variation of area of tampered region. This will depend on how accurately the size of tampering region is required to be found out.

Reduction in sampling block size will result in increased size of hash matrix and consequently length of hash vector. In certain situations, the length of hash vector can be a limiting factor due to system requirements. In such cases, a trade-off between hash vector size and accuracy of tampering localization needs to be sought for.

7.2 Quantification of Robustness against CPM

One of the requirements of an efficient image tampering detection technique is robustness against content preserving manipulations such as low pass filtering (blur noise), minor contrast adjustment, brightness change and compression. The detection technique should be able to detect very minute structural tampering and at the same time, should be able to ignore content preserving manipulations mentioned above. Researchers have come out with algorithms which achieve both of these requirements. However, the critical level, up to which these manipulations will be ignored, has not been quantified in earlier works.

Generally, a typical tampering operation is limited to small part of the image while the content preserving manipulations are normally global in nature. It might just happen that quantity of content preserving manipulation summed up over full image, exceeds amount of malicious structural tampering which is limited to much smaller area. In that case, tampering index corresponding to content preserving manipulation will fall in category of malicious tampering. Therefore robustness should always be defined in terms of critical level of tampering above which CPM will be detected as malicious tampering.

In our work, we have identified the critical level of content preserving manipulations of each type. For example, in case of contrast change carried out through application of gamma correction, critical value of gamma has been found out up to which change in tampering index is very small and can be ignored. However, higher value of gamma, the resulting contrast change will enter the category of malicious tampering.

7.3 Similarity Value as measure of tampering

As mentioned above, a large number of techniques have been evolved to identify the tampered area in an image based on image feature extraction based hashing method. However, no serious effort has been made to give a mathematical index for amount of tampering. In this work, a quantitative index for amount of tampering called Similarity value as in eqn. (6.22) has been defined. It is very unique feature of our research work. It is a very suitable mathematical index because of two reasons.

1. It assumes values between 0 and 1 which is ideal to understand and use mathematically.
2. It varies with amount of tampering, assuming value 1 for non-tampered pair of images and 0 for completely dissimilar pair of images. In general it lies between 0 and 1 depending on amount of tampering.

7.4 Singular Value Decomposition for highly secure hash function

Properties of singular value matrix and orthogonal matrices in SVD were used to generate 2-level key based hash generation. It provided highly secure hash function specially suited to high security areas. Concept of 2-level key based hash is a novel method for tmapering detection.

7.5 Use of Canny Edge Detector

So far, several techniques have been used for extracting the features of the image for hash generation. The basic requirement of an efficient hash generation technique are:

1. The technique should be extremely sensitive in detecting structural tampering so that even very minute tampering operations are picked up during

feature extraction and are suitably converted into hash value.

2. The technique should be able to ignore content preserving manipulations.

Canny Edge Detector meets both of these requirements very well. Firstly, it has efficient edge detection capability which provides very accurate image feature extraction. Double Edge Threshold method for picking or leaving an edge followed by Hysteresis Tracking for accepting/rejecting connected and disconnected edges respectively, provides very accurate representation of the image. This also ensures that content preserving manipulations are ignored as they normally lie below lower threshold level of Canny Edge Detector.

A very efficient hash function has been defined called "Average Edge Index" which is used to generate hash values for the blocks of image. This formulation has been defined and used for the first time in the field of image forensics. It is, therefore, a major contribution.

7.6 Comprehensive Image Index

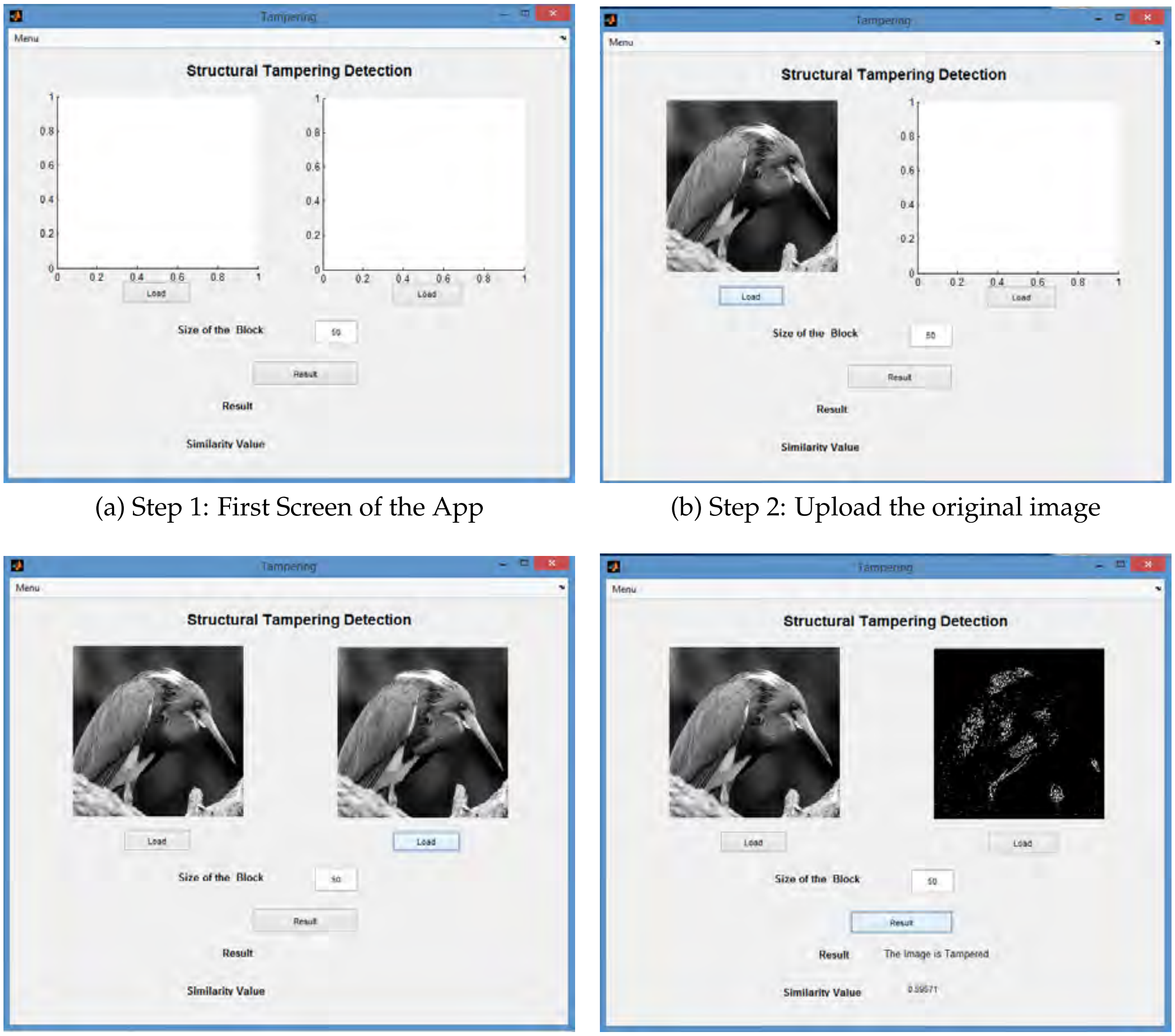
Earlier researchers have used single feature of an image to generate hash value which is used to generate a hash matrix. This hash matrix is used for detection of tampering with respect to that particular feature. As discussed earlier, structural tampering is generally followed by brightness level tampering, contrast level tampering or both in order to hide structuraal tampering. To give a comprehensive measurement of this kind of multiple tampering, Comprehensive Image Index (η_1, η_2, η_3) was defined and used successfully for various combination of tampering operations.

Concept of 3-tupled hash function for representation of image was used for the first time and therefore it is a very significant contribution.

7.7 An Image Tampering App

Based on our findings and results, we developed a plug-and-play type App for identification and localization of tampering in a suspect image against a given image. The situation, we may visualize, the Forensic Science Lab be given to identify whether a suspect image is tampered with or not. This App is so handy and easy to use which requires three steps:

1. Upload the original image
2. Upload the suspect image
3. Decide the block size (such as 50, 25, 20 etc.)
4. Click on 'Result' button to find the result and similarity value in no time.



(a) Step 1: First Screen of the App

Size of the Block 50

Result

Similarity Value

(b) Step 2: Upload the original image

Size of the Block 50

Result

Similarity Value

(c) Step 3: Upload the suspect image

Size of the Block 50

Result

Similarity Value

(d) Step 4: Get result

Size of the Block 50

Result The Image is Tampered

Similarity Value 0.99571

Figure 7.1: The developed App which may make the FSL professional's life easy.

CHAPTER 8

Conclusion and Future Scope of Our Work

In the ever-expanding digital image environment, tampering of digital image poses a serious threat especially in legal, commercial and scientific research area. A humble effort has been made by us to provide efficient tampering detection algorithms using image feature based hash generation technique. Few methods were attempted and tested against the required properties of hash functions through experiments on large image data base. Relative merits of all the technique have been discussed in respective chapters.

Firstly, correlation coefficient was used to extract image features and was used to calculate the hash values. Certain mathematical tools such as hash matrix, hash vector and similarity value were used to carry out tampering detection, localization along with quantitative measurement of amount of tampering. Robustness and sensitivity was also proved experimentally. Second method based on Singular Value Decomposition of image matrix was used to achieve very high degree of security of hash function. Collision probability test was carried out over a very large data set and it was found to be extremely low and thus proving the high quality of chosen hash function.

Next step was to look for tampering detection algorithm which was highly sensitive but also equally robust against content preserving manipulations. Canny Edge Detector was used as feature extractor and hash values were generated by defining a new function called Average Edge Index which is based on edge content of the image. High degree of sensitivity and robustness was achieved though with a slightly high collision probability (but still very low) as compared to SVD method.

In the last method, we defined a multiple parameters hash value using a new index called Comprehensive Image Index (CII). This method can detect multiple tampering operations in the image simultaneously. The multiple parameters were structure, brightness and contrast. CII is able to give idea about the motive of the attacker and very useful for forensic laboratories. This technique satisfied various requirements of good hash function.

This work is being continued using different techniques for different objectives. Haar wavelet transform based image hashing is being worked out which will be helpful in detecting tampering in medical images and machine design. Further work may be pursued to use compressive sensing technique for feature based hash generation. Area of work can be expanded to blind techniques which together with our work can prove to be very useful for forensic science laboratories.

CHAPTER 9

Our Publications

1. Mall, Vinod; Bhatt, Kedar; Mitra, Suman K. and Roy, Anil K., “Exposing structural tampering in digital images," Signal Processing, Computing and Control (ISPCC), 2012 IEEE International Conference on, pp. 1-6, Mar. 2012.

2. Mall, Vinod; Roy, Anil K.; Mitra, Suman K. and Bhatt, Kedar, “Non-blind method of detection and localization of structural tampering using robust Hash-like function and similarity metric for digital images," TENCON 2012 - 2012 IEEE Region 10 Conference, pp. 1-6, Nov. 2012.

3. Mall, Vinod; Roy, Anil K.; Mitra, Suman K. and Shukla, Shivanshu, “Comprehensive Image Index and Detection of Tampering in a Digital Image," Proceedings of International Conference on Informatics, Electronics & Vision (ICIEV 2013), May 2013.

4. Mall, Vinod; Roy, Anil K.; Mitra, Suman K. and Shukla, Shivanshu, “Detection of Structural Tampering in a Digital Image Using Canny Edge Detector," Proceedings of International Conference on Informatics, Electronics & Vision (ICIEV 2013), May 2013.

5. Mall, Vinod; Roy, Anil K.; and Mitra, Suman K., “Digital image tampering detection and localization using singular value decomposition technique," Proceedings of Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG 2013), Dec 2013.

References

[1] H. R. Center, "The first photograph," *The University of Texas at Austin*. [Online]. Available: http://www.hrc.utexas.edu/exhibitions/permanent/firstphotograph/

[2] "Digital forensics: Photo tampering throughout history," *Scientific American*, June 2008, last accessed on July 15, 2013. [Online]. Available: http://www.scientificamerican.com/slideshow.cfm?id=photo-tampering-throughout-history

[3] M. Fach, "Stats on facebook 2012 [infographic]," *Search Engine Journal*, Posted on February 17, 2012 and last accessed on July 15, 2013. [Online]. Available: http://www.searchenginejournal.com/stats-on-facebook-2012-infographic/40301/

[4] "Our story: A quick walk through our history as a company," *Official site of Instagram*, last accessed on July 15, 2013. [Online]. Available: http://instagram.com/press/#

[5] J. Rich, "ipad and iphone apps for editing and enhancing photos," *Que Online Publishing*, Posted on April 1, 2013 and last accessed on July 15, 2013. [Online]. Available: http://www.quepublishing.com/articles/article.aspx?p=2036546

[6] T. Moynihan, "10 photo editing programs (that aren't photoshop)," *Digital Photography Review*, Posted on May 17, 2013 and last accessed on July 15, 2013. [Online]. Available: http://www.dpreview.com/articles/6648389507/10-photo-editing-programs-that-arent-photoshop

[7] "Photo tampering throughout history," *FourandSix*, last accessed on July 15, 2013. [Online]. Available: http://www.fourandsix.com/photo-tampering-history/tag/politics?currentPage=9

[8] D. Hochanadel, "Restoring photojournalismâĂŹs credibility in the age of photoshop," *an honors Thesis submitted at The University of Toledo*, April 2009, last accessed on July 15, 2013. [Online]. Available: http://davehochanadel.com/accordion/writing/pdfs/davehochanadel-honorsthesis.pdf

[9] D. Lang, "Blade editor: Detrich submitted 79 altered photos this year," *Photo District News Magazine*, Posted on April 15, 2007 and last accessed on July 15, 2013. [Online]. Available: http://www.filmjournal.com/pdn/esearch/article_display.jsp?vnu_content_id=1003571795

[10] Reporter, "Britney spears bravely agrees to release un-airbrushed images of herself next to the digitally-altered versions," *Daily Mail Online*, Posted on 13 April, 2010 and last accessed on July 15, 2013. [Online]. Available: http://www.dailymail.co.uk/tvshowbiz/article-1265676/Britney-Spears-releases-airbrushed-images-digitally-altered-versions.html

[11] "Photo tampering throughout history," *FourandSix*, last accessed on July 15, 2013. [Online]. Available: http://www.fourandsix.com/photo-tampering-history/tag/marketing?currentPage=2

[12] C. Bautista, "Digital harassment linked with physical abuse among dating teens," *Mashable*, Posted on February 22, 2013 and last accessed on July 15, 2013. [Online]. Available: http://mashable.com/2013/02/21/teen-dating/

[13] H. Pearson, "Image manipulation csi: cell biology," *Nature*, vol. 434, pp. 952–953, April 2005.

[14] D. Cyranoski, "Verdict: Hwang's human stem cells were all fakes," *Nature*, vol. 439, pp. 122–123, Jan. 2006.

[15] I. Fuyuno and D. Cyranoski, "Doubts over biochemist's data expose holes in japanese fraud laws," *Nature*, vol. 439, p. 514, Feb. 2006.

[16] *Daily Makeover*, last accessed on July 15, 2013. [Online]. Available: http://www.dailymakeover.com/virtual-makeover/#/home

[17] L. Rao, "Daily makeover tries to re-create the beauty counter online," *Tech Crunch*, Posted on Aug 23, 2009 and last accessed on July 15, 2013. [Online]. Available: http://techcrunch.com/2009/08/23/daily-makeover-tries-to-re-create-the-beauty-counter-online/

[18] "Who got the best makeover on america's next top model?" *PopSugar Fashion*, Posted on Mar 16, 2007 and last accessed on July 15, 2013. [Online]. Available: http://www.fabsugar.com/Who-Got-Best-Makeover-America-Next-Top-Model-176553

[19] L. Indvik, "Nars launches interactive makeover site for beauty enthusiasts," *Mashable*, Posted on May 06, 2011 and last accessed on July 15, 2013. [Online]. Available: http://mashable.com/2011/05/06/nars-beauty-makeover-site/

[20] F. Y. Shih, *Digital Watermarking and Steganography: Fundamentals and Techniques*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2007.

[21] I. Kim, S.-S. Han, and J. H. Shin, "An improved tamper-detection method for digital images," in *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, vol. 1, 2001, pp. 227–231.

[22] T. V. Lanh, K.-S. Chong, S. Emmanuel, and M. Kankanhalli, "A survey on digital camera image forensic methods," in *Multimedia and Expo, 2007 IEEE International Conference on*, July 2007, pp. 16–19.

[23] M. A. Akhaee and F. Marvasti, "A survey on digital data hiding schemes: Principals, algorithms, and applications," *ISeCure, The ISC International Journal of Information Security*, vol. 5, no. 1, pp. 5–36, 2013.

[24] M. Schneider and S.-F. Chang, "A robust content based digital signature for image authentication," in *Image Processing, 1996. Proceedings., International Conference on*, vol. 3, Sep. 1996, pp. 227–230.

[25] J.-B. Martens and L. Meesters, "Image dissimilarity," *Signal Processing*, vol. 70, no. 3, pp. 155–176, Nov. 1998. [Online]. Available: http://www.sciencedirect.com/science/article/B6V18-3VCDJVV-2/1/5d14978d45eb7270976443b4060d61ac

[26] S. Bhattacharjee and M. Kutter, "Compression tolerant image authentication," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 1, Oct. 1998, pp. 435–439.

[27] D.-C. Lou and J.-L. Liu, "Fault resilient and compression tolerant digital signature for image authentication," *Consumer Electronics, IEEE Transactions on*, vol. 46, no. 1, pp. 31–39, Feb. 2000.

[28] A. J. Fridrich, B. D. Soukal, and A. J. LukÃąÅą, "Detection of copy-move forgery in digital images," in *in Proceedings of Digital Forensic Research Workshop*, 2003.

[29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE TRANSACTIONS ON IMAGE PROCESSING*, vol. 13, no. 4, pp. 600–612, 2004.

[30] Q. Li and S. Roy, "On the security of non-forgeable robust hash functions," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, Oct 2008, pp. 3124–3127.

[31] M.-H. Lin, Y.-C. Hu, and C.-C. Chang, "An image self-verification scheme based on the rehash technique," in *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on*, vol. 2, April 2003, pp. 1883–1886.

[32] R. Granty, T. Aditya, and S. Madhu, "Survey on passive methods of image tampering detection," in *Communication and Computational Intelligence (INCOCCI), 2010 International Conference on*, Dec 2010, pp. 431–436.

[33] M. Stamm and K. Liu, "Forensic detection of image manipulation using statistical intrinsic fingerprints," *Information Forensics and Security, IEEE Transactions on*, vol. 5, no. 3, pp. 492–506, Sept. 2010.

[34] W. Luo, J. Huang, and G. Qiu, "Jpeg error analysis and its applications to digital image forensics," *Information Forensics and Security, IEEE Transactions on*, vol. 5, no. 3, pp. 480–491, Sept. 2010.

[35] K. Cai, X. Lu, J. Song, and X. Wang, "Blind image tampering identification based on histogram features," in *Multimedia Information Networking and Security (MINES), 2011 Third International Conference on*, Nov. 2011, pp. 300–303.

[36] C.-Y. Lin and S.-F. Chang, "A robust image authentication method distinguishing jpeg compression from malicious manipulation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 2, pp. 153–168, Feb 2001.

[37] V. Thing, Y. Chen, and C. Cheh, "An improved double compression detection method for jpeg image forensics," in *Multimedia (ISM), 2012 IEEE International Symposium on*, Dec 2012, pp. 290–297.

[38] A. C. Popescu and H. Farid, "Statistical tools for digital forensics," in *In 6th International Workshop on Information Hiding*. Springer-Verlag, Berlin-Heidelberg, 2004, pp. 128–147.

[39] A. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *Signal Processing, IEEE Transactions on*, vol. 53, no. 2, pp. 758–767, Feb. 2005.

[40] X. Zhang, Z. Qian, Y. Ren, and G. Feng, "Watermarking with flexible self-recovery quality based on compressive sensing and compositive reconstruction," *Information Forensics and Security, IEEE Transactions on*, vol. 6, no. 4, pp. 1223–1232, Dec 2011.

[41] P. Korus and A. Dziech, “Efficient method for content reconstruction with self-embedding,” *Image Processing, IEEE Transactions on*, vol. 22, no. 3, pp. 1134–1147, March 2013.

[42] X. Zhou, X. Duan, and D. Wang, “A semifragile watermark scheme for image authentication,” in *Multimedia Modelling Conference, 2004. Proceedings. 10th International*, Jan 2004, pp. 374–377.

[43] A. Swaminathan, M. Wu, and K. Liu, “Digital image forensics via intrinsic fingerprints,” *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 1, pp. 101–117, March 2008.

[44] Z. Guojuan and L. Dianji, “An overview of digital watermarking in image forensics,” in *Computational Sciences and Optimization (CSO), 2011 Fourth International Joint Conference on*, April 2011, pp. 332–335.

[45] J. Fridrich and M. Goljan, “Robust hash functions for digital watermarking,” *Information Technology: Coding and Computing, International Conference on*, p. 178, 2000.

[46] R. Granty, T. Aditya, and S. Madhu, “Survey on passive methods of image tampering detection,” in *Communication and Computational Intelligence (INCOCCI), 2010 International Conference on*, Dec. 2010, pp. 431–436.

[47] A. Smoaca, M. Petrovici, D. Coltuc, and V. Lazarescu, “A robust hashing of id photos,” in *Signals, Circuits and Systems (ISSCS), 2011 10th International Symposium on*, June 2011, pp. 1–4.

[48] F. Khelifi and J. Jiang, “Perceptual image hashing based on virtual watermark detection,” *Image Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 981–994, April 2010.

[49] Q. Sun, Q. Tian, and S.-F. Chang, “A robust and secure media signature scheme for jpeg images,” in *Multimedia Signal Processing, 2002 IEEE Workshop on*, Dec 2002, pp. 296–299.

[50] C. Rey and J.-L. Dugelay, “A survey of watermarking algorithms for image authentication,” *EURASIP J. Appl. Signal Process.*, vol. 2002, no. 1, pp. 613–621, Jan 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=1283100.1283165

[51] M. Iwata, T. Hori, A. Shiozaki, and A. Ogihara, “Digital watermarking method for tamper detection and recovery of jpeg images,” in *Information Theory and its Applications (ISITA), 2010 International Symposium on*, Oct 2010, pp. 309–314.

[52] J. O'Ruanaidh and T. Pun, “Rotation, scale and translation invariant digital image watermarking,” in *Image Processing, 1997. Proceedings., International Conference on*, vol. 1, Oct 1997, pp. 536–539.

[53] P. W. Wong and N. Memon, “Secret and public key image watermarking schemes for image authentication and ownership verification,” *Image Processing, IEEE Transactions on*, vol. 10, no. 10, pp. 1593–1601, Oct. 2001.

[54] N. Dharwadkar, B. Amberker, and A. Gorai, “Non-blind watermarking scheme for color images in rgb space using dwt-svd,” in *Communications and Signal Processing (ICCSP), 2011 International Conference on*, Feb. 2011, pp. 489–4933.

[55] P.-L. Lin, P.-W. Huang, and A.-W. Peng, “A fragile watermarking scheme for image authentication with localization and recovery,” in *Multimedia Software Engineering, 2004. Proceedings. IEEE Sixth International Symposium on*, Dec 2004, pp. 146–153.

[56] P. Wong and N. Memon, “Secret and public key image watermarking schemes for image authentication and ownership verification,” *Image Processing, IEEE Transactions on*, vol. 10, no. 10, pp. 1593–1601, Oct 2001.

[57] Y. Wu, F. Bao, and C. Xu, “The security flaws in some authentication watermarking schemes,” in *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, vol. 2, July 2003, pp. II–493–6.

[58] S. Dadkhah, A. Manaf, and S. Sadeghi, “Efficient two level image tamper detection using three lsb watermarking,” in *Computational Intelligence and Communication Networks (CICN), 2012 Fourth International Conference on*, Nov 2012, pp. 719–723.

[59] X. Kang and S. Wei, “Identifying tampered regions using singular value decomposition in digital image forensics,” in *Computer Science and Software Engineering, 2008 International Conference on*, vol. 3, Dec. 2008, pp. 926–930.

[60] S. Roy and Q. Sun, “Robust hash for detecting and localizing image tampering,” in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 6, Oct. 2007, pp. VI –117 –VI –120.

[61] S. Dehnie, T. Sencar, and N. Memon, “Digital image forensics for identifying computer generated and digital camera images,” in *Image Processing, 2006 IEEE International Conference on*, Oct. 2006, pp. 2313–2316.

[62] J. Lukas and J. Fridrich, “Estimation of primary quantization matrix in double compressed jpeg images,” in *Proc. of DFRWS*, 2003.

[63] M. Stamm, S. Tjoa, W. Lin, and K. Liu, “Undetectable image tampering through jpeg compression anti-forensics,” in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, Sept. 2010, pp. 2109–2112.

[64] J. Qiu and P. Wang, “An image encryption and authentication scheme,” in *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on*, Dec 2011, pp. 784–787.

[65] A. Rajput, N. Mishra, and S. Sharma, “Towards the growth of image encryption and authentication schemes,” in *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, Aug 2013, pp. 454–459.

[66] C.-S. Lu and H.-Y. Liao, “Structural digital signature for image authentication: an incidental distortion resistant scheme,” *Multimedia, IEEE Transactions on*, vol. 5, no. 2, pp. 161–173, June 2003.

[67] A. Jeng, L.-C. Chang, and H.-J. Li, “Exploring better parameter set for singular value decomposition (svd) hashing function used in image authentication,” in *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, vol. 5, July 2010, pp. 2600–2604.

[68] T. Chihaoui, S. Bourouis, and K. Hamrouni, “Copy-move image forgery detection based on sift descriptors and svd-matching,” in *Advanced Technologies for Signal and Image Processing (ATSIP), 2014 1st International Conference on*, March 2014, pp. 125–129.

[69] R. Venkatesan, S.-M. Koon, M. Jakubowski, and P. Moulin, “Robust image hashing,” in *Image Processing, 2000. Proceedings. 2000 International Conference on*, vol. 3, 2000, pp. 664–666.

[70] C.-S. Lu and H.-Y. Liao, “Structural digital signature for image authentication: an incidental distortion resistant scheme,” *Multimedia, IEEE Transactions on*, vol. 5, no. 2, pp. 161–173, June 2003.

[71] V. Monga and B. Evans, “Perceptual image hashing via feature points: Performance evaluation and tradeoffs,” *Image Processing, IEEE Transactions on*, vol. 15, no. 11, pp. 3452–3465, Nov. 2006.

[72] V. Monga and M. Mihcak, “Robust and secure image hashing via non-negative matrix factorizations,” *Information Forensics and Security, IEEE Transactions on*, vol. 2, no. 3, pp. 376–390, Sept. 2007.

[73] S. Lin and Q. Xie, “A secure and efficient mutual authentication protocol using hash function,” in *Communications and Mobile Computing, 2009. CMC '09. WRI International Conference on*, vol. 3, Jan 2009, pp. 545–548.

[74] M. Tagliasacchi, G. Valenzise, and S. Tubaro, “Hash-based identification of sparse image tampering,” *Image Processing, IEEE Transactions on*, vol. 18, no. 11, pp. 2491–2504, Nov 2009.

[75] F. Lefebvre, J. Czyz, and B. Macq, "A robust soft hash algorithm for digital image signature," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 2, Sept. 2003, pp. II – 495–8 vol.3.

[76] J. Fridrich and M. Goljan, "Robust hash functions for digital watermarking," in *Information Technology: Coding and Computing, 2000. Proceedings. International Conference on*, 2000, pp. 178–183.

[77] H. Kobayashi and H. Kiya, "Robust image authentication using hash function," in *TENCON 2004. 2004 IEEE Region 10 Conference*, vol. A, Nov 2004, pp. 435–438 Vol. 1.

[78] L. Yu and S. Sun, "Image robust hashing based on dct sign," in *Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP '06. International Conference on*, Dec 2006, pp. 131–134.

[79] S. Bayram, I. Avcibas, B. Sankur, and N. Memon, "Image manipulation detection," *Journal of Electronic Imaging*, vol. 15, no. 4, p. 041102, 2006.

[80] F. Ahmed, M. Siyal, and V. U. Abbas, "A secure and robust hash-based scheme for image authentication," *Signal Processing*, vol. 90, no. 5, pp. 1456–1470, 2010.

[81] Z. Tang, S. Wang, X. Zhang, and W. Wei, "Perceptual similarity metric resilient to rotation for application in robust image hashing," in *Multimedia and Ubiquitous Engineering, 2009. MUE '09. Third International Conference on*, June 2009, pp. 183–188.

[82] A. Swaminathan, Y. Mao, and M. Wu, "Robust and secure image hashing," *Information Forensics and Security, IEEE Transactions on*, vol. 1, no. 2, pp. 215–230, June 2006.

[83] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and P. Moulin, "Robust image hashing," in *Image Processing, 2000. Proceedings. 2000 International Conference on*, vol. 3, 2000, pp. 664–666 vol.3.

[84] L. Xie, G. Arce, and R. Graveman, "Approximate image message authentication codes," *Multimedia, IEEE Transactions on*, vol. 3, no. 2, pp. 242–252, Jun 2001.

[85] S. Bhattacharjee and M. Kutter, "Compression tolerant image authentication," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 1, Oct 1998, pp. 435–439 vol.1.

[86] F. Lefebvre, J. Czyz, and B. Macq, "A robust soft hash algorithm for digital image signature," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 2, Sept 2003, pp. II–495–8 vol.3.

[87] F. Khelifi and J. Jiang, "Analysis of the security of perceptual image hashing based on non-negative matrix factorization," *Signal Processing Letters, IEEE*, vol. 17, no. 1, pp. 43–46, Jan 2010.

[88] Y. Li, "Robust image hash function based on polar harmonic transforms and feature selection," in *Computational Intelligence and Security (CIS), 2012 Eighth International Conference on*, Nov 2012, pp. 420–424.

[89] Y. Zhao and W. Wei, "Extraction of shape feature for image authentication," in *Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on*, vol. 1, June 2011, pp. 404–408.

[90] S. Jothimani and P. Betty, "Image authentication using global and local features," in *Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on*, March 2014, pp. 1–5.

[91] L.-W. Kang, C.-S. Lu, and C.-Y. Hsu, "Compressive sensing-based image hashing," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, Nov 2009, pp. 1285–1288.

[92] R. Radhakrishnan, W. Jiang, and C. Bauer, "On improving the collision property of robust hashing based on projections," in *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, June 2009, pp. 862–865.

[93] H. J. Kim, S. Y. Kim, I.-K. Yeo, and A. Md, “Mathematical performance evaluation tool for image hash generation functions,” in *Visual Information Engineering, 2008. VIE 2008. 5th International Conference on*, July 2008, pp. 221–226.

[94] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*, 2nd ed. Springer, 2009. [Online]. Available: http://www-stat.stanford.edu/~tibs/ElemStatLearn/

[95] Z. Tang, S. Wang, X. Zhang, and W. Wei, “Structural feature-based image hashing and similarity metric for tampering detection,” *Fundam. Inf.*, vol. 106, no. 1, pp. 75–91, Jan. 2011.

[96] Z. Liu, Q. Li, H. Zhang, and X. Peng, “An image structure information based robust hash for tamper detection and localization,” in *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on*, Oct. 2010, pp. 430–433.

[97] R. Hernandez, M. Miyatake, and B. Kurkoski, “Robust image hashing using image normalization and svd decomposition,” in *Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on*, Aug 2011, pp. 1–4.

[98] D. C. Lay, *Linear Algebra and Its Applications*. Pearson Education, 2002.

[99] A. Goon, B. Gupta, and M. Dasgupta, *Fundamentals of Statistics*. World Press, 2005.

[100] J. Wang, G. Liu, B. Xu, H. Li, Y. Dai, and Z. Wang, “Image forgery forensics based on manual blurred edge detection,” in *Multimedia Information Networking and Security (MINES), 2010 International Conference on*, Nov. 2010, pp. 907–911.

[101] L. DAVIS, “A survey of edge detection techniques,” *CGIP*, vol. 4, no. 3, pp. 248–270, 1975.

[102] E. Argyle and A. Rosenfeld, “Techniques for edge detection,” *Proceedings of the IEEE*, vol. 59, no. 2, pp. 285–287, Feb 1971.

[103] K. Vasavi, M. Latha, and N. Kumar, “A deterministic edge detection using statistical approach,” in *Conference on Computational Intelligence and Multimedia Applications, 2007. International Conference on*, vol. 3, Dec 2007, pp. 282–286.

[104] D. Marr and E. Hildreth, “Theory of edge detection,” *Proceedings of Royal Society of London B*, vol. 207, no. 1167, pp. 187–217, Feb 1980.

[105] V. Torre and T. Poggio, “On edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 147–163, 1984.

[106] S. Bhardwaj and A. Mittal, “A survey on various edge detector techniques,” *Procedia Technology*, vol. 4, no. 0, pp. 220 – 226, 2012, 2nd International Conference on Computer, Communication, Control and Information Technology(C3IT-2012) on February 25 - 26, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S221201731200312X

[107] Q. Xu, S. Varadarajan, C. Chakrabarti, and L. Karam, “A distributed canny edge detector: Algorithm and fpga implementation,” *Image Processing, IEEE Transactions on*, vol. 23, no. 7, pp. 2944–2960, July 2014.

[108] V. Bhadouria and D. Ghoshal, “Edge detection in electron microscopy biological images using statistical dispersion,” in *Machine Vision and Image Processing (MVIP), 2012 International Conference on*, Dec 2012, pp. 96–100.

[109] X. He, J. Li, D. Wei, W. Jia, and Q. Wu, “Canny edge detection on a virtual hexagonal image structure,” in *Pervasive Computing (JCPC), 2009 Joint Conferences on*, Dec 2009, pp. 167–172.

[110] S. Pirzada and A. Siddiqui, “Analysis of edge detection algorithms for feature extraction in satellite images,” in *Space Science and Communication (IconSpace), 2013 IEEE International Conference on*, July 2013, pp. 238–242.

[111] Y. Luo and R. Duraiswami, "Canny edge detection on nvidia cuda," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, June 2008, pp. 1–8.

[112] C.-Y. Hsu, H.-F. Wang, H.-C. Wang, K.-K. Tseng, and Y.-J. Tang, "Automatic extraction of face contours," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*, July 2010, pp. 1–8.

[113] M. Mekideche and Y. Ferdi, "A new edge detector based on fractional integration," in *Multimedia Computing and Systems (ICMCS), 2014 International Conference on*, April 2014, pp. 223–228.

[114] M. Setayesh, M. Zhang, and M. Johnston, "Edge detection using constrained discrete particle swarm optimisation in noisy images," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, June 2011, pp. 246–253.

[115] S. Varadarajan, C. Chakrabarti, L. Karam, and J. M. Bauza, "A distributed psycho-visually motivated canny edge detector," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, March 2010, pp. 822–825.

[116] V. Mall, A. K. Roy, S. K. Mitra, and K. Bhatt, "Non-blind method of detection and localization of structural tampering using robust hash-like function and similarity metric for digital images," in *TENCON 2012 - 2012 IEEE Region 10 Conference*, Nov. 2012, pp. 1–6.

[117] X. Z. Zhenjun Tang, Shuozhong Wang and W. Wei, "Structural feature-based image hashing and similarity metric for tampering detection," *Fundamenta Informaticae*, vol. 106, pp. 75–91, 2011.

[118] Z. Wang and A. Bovik, "A universal image quality index," *Signal Processing Letters, IEEE*, vol. 9, no. 3, pp. 81–84, March 2002.

[119] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.