Review on Tools for Image Detection Forgery

Fatma Salman Hashem*, Ghazali Bin Sulong

Faculty of Computing, University Technology Malaysia, UTM

Abstract

This paper defines the presently used methods and approaches in the domain of digital image forgery detection. A survey of a recent study is explored including an examination of the current techniques and passive approaches in detecting image tampering. This area of research is relatively new and only a few sources exist that directly relate to the detection of image forgeries. Fake images have become widespread in society today. The accessibility to powerful simple to use image editing computer software to end users helps make the job of manipulating image incredibly easy. One can find forged images used to sensationalize news, spread political propaganda and rumors, introduce psychological bias, etc. in all forms of media.

Keywords. Detecting Image, Forgery Detection, Tampering Techniques.

1 Introduction

Maliciously manipulate, and tamper digital images without leaving any obvious clues became very easy with the widely available, easy to use and extremely powerful digital image processing tools such as Photoshop. As a result, there is a rapid increase of the digitally manipulated splicing in mainstream media and on the Internet. This trend indicates serious vulnerabilities and decreases the credibility of the digital images. Therefore, developing techniques to verify the integrity and the authenticity of the digital images became very important, especially considering the images presented as evidence in a court of law, as news items, as a part of a medical record, or as a financial document. In this sense, image tamper detection is one of the primary goals in image forensics.

Fake images have become widespread in society today. The accessibility to powerful simple to use image editing computer software to end users helps make the job of manipulating image incredibly easy. One can find forged images used to sensationalize news, spread political propaganda and rumors, introduce psychological bias, etc. in all forms of media. Claims of image tampering are common in scandals and controversies. As the credibility of images suffers, it is necessary to devise techniques in order to verify their authenticity and trustworthiness of digital images [1].

*Corresponding author: eng_fatmas@yahoo.com.

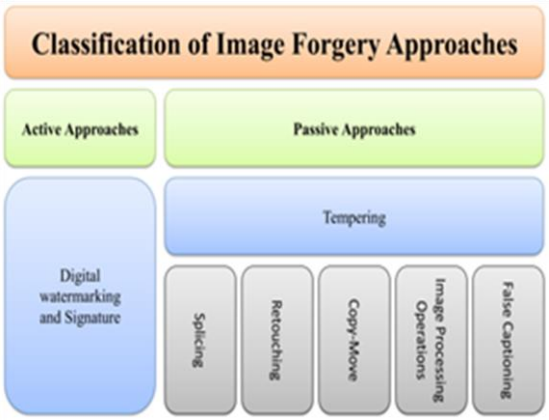


Fig. 1. Classification of image forgery approaches

2 TOOLS FOR IMAGE FORGERY DETECTION

2.1 Edge Detection using first-order operators

A typical image processing method which is an edge detection algorithms have been examined against a number of forged test images [10]. Lukas was the first one to examine them meanwhile edge detection algorithms are considered as an important application in image processing. In many applications, the image's edges are tremendously major because they offer information that concerns the texture, size, shape and the location of objects. Image tampering offers unknown differences frequently related with double edge around the tampered objects which makes this concept interested in forgery detection. This event happened when the blurring of space around the tampered objects is formed, while a "ghost" or a double edge is formed with the real edge.

A definition of an edge says that an edge in an image is those zones or areas where pixels' intensity fluctuated from a low to a high value or the opposite (Luong, 2004).This leads into an analysis of first-order operators and their power at detecting discontinuities. First-order operators detect points in the image that are discontinuous by calculating a function of the image which uses first-order derivatives. There are various convolution masks used in image processing and some have already been used to analyze forged digital images. Previous images were analyzed using the Roberts, Sobel, and Prewitt masks [10]. The Sobel mask is more receptive to edges that are diagonal in nature rather than horizontal or vertical. The Roberts mask is more susceptible to noise than the other masks while Prewitt is better at horizontal and vertical edges [11].

The following formula computes the convolution of an image [10]:

$$h_{x,y} = \sum_{i=d}^{n} \sum_{j=d}^{n} g_{i,j} f_{x+i,y+j},$$

Where and, g is a convolution mask of size s x s, and f is the image function.

The following are the masks described above and used for the variable g

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Roberts' mask

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Sobel mask

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

Prewitt mask

2.2 Edge Detection using second-order operators

First-order operators are a good fundamental technique to use in image processing and forgery detection, but second-order operators offer a distinct approach in the detection of image forgeries. Second-order operators provide an alternative method at detecting what is considered an edge, which allows for more robustness. This is true because second-order operators provide much better edge localization based on how they calculate the edge. Instead of calculating an edge several pixels wide, and thus posing the problem of determining the center of an edge, second-order operators attempt to guard against this [11]. Second-order operators use Laplacian and Gaussian functions to calculate the convolutions of the image in question. These techniques are robust against various image degradations, i.e. noise, because of the Gaussian function [10]. Marr and Hildreth posed this technique which looks for zero-crossings after convolution with the Laplacian and the Gaussian functions. The Marr edge detector first performs Gaussian smoothing before convolving the image with the Laplacian function [11].

An example of a Marr edge detector of order 5 x 5 is given below [10]:

$$\begin{bmatrix} 0 & 3 & 6 & 3 & 0 \\ 3 & 15 & 0 & 15 & 3 \\ 6 & 0 & -108 & 0 & 6 \\ 3 & 15 & 0 & 15 & 3 \\ 0 & 3 & 6 & 3 & 0 \end{bmatrix}$$

Marr edge detector of order 5 x 5

This mask provides symmetry both horizontally and vertically. This is due to the symmetry of the Gaussian function which enables equal balance across portions of the image being filtered. The power of edge detection permits the possibility of detecting hidden discontinuities, which might be prevalent in image forgeries [10]. The Marr edge detector follows similar symmetry for larger size matrices of higher order. The next subsection presents a different, but equally interesting, image processing approach dealing with

frequency analysis.

2.3 Spectral Analysis

Spectral analysis methods use the power of Discrete Fourier Transforms (DFTs) and (DFTs) capability to detect the intensity and the brightness levels in an image. The formula represented below is utilized to figure the DFT of a sample image [10]

$$F_{x,y} = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_{m,n} e^{-2\pi i (\frac{xm}{M} + \frac{yn}{N})},$$

Where f is the image of size M x N represented as a brightness function of each pixel. Lukas analyzed some preliminary test images using the power of DFTs [10]. This technique allows one to see areas of the image that may be manipulated, by looking for the natural logarithm of the amplitude in high frequencies of the image. Since a digital image can be treated as a two dimensional signal, tampering with an area of an image introduces anomalies in the frequency of this signal. If a local maximum in the high frequency range is present when performing spectral analysis, the image may be a victim of an image forgery. [10].

Farid and Popescu spread out Lukas's spectral analysis method by presenting an encouraging process which detects image forgeries based on the observed effects of resampling an image [12]. Their respective method differs from Lukas' in that it concentrates on pre-processing and filtering the image in an attempt to achieve high detection accuracy. Fully analyzing the forgery process and its effect on the victim image enabled Farid and Popescu to develop a fully customizable method.

Forged images are formed by the combination of two or more host images, which demands the cropping, resizing or rescaling of two or one of the host images. As a result spectral analysis detects the underlying changes in the image's numerical nature. The calculation of the Fourier transform in the manipulated zones reveals that those zones have been re-sampled by looking for a periodic pattern [12] To further explain this technique as well as the expected results, the following figures provide an example.

This technique using Fourier transform analysis has been found to work best on uncompressed images, i.e. TIFF. Images saved in the lossy JPEG format exhibit much lower detection accuracy with Quality Factors of 97/100 and lower. When a JPEG image has been saved using a Quality Factor of 90/100 or lower, detection becomes an extremely hit or miss occurrence. The introduction of noise and the periodic block pattern of the JPEG compression algorithm are the suspected reasons for this difficulty. (Farid and Popescu, 2004). As the Quality Factor goes down, the above two observable facts increase, thus causing Fourier transform analysis to become less reliable. Most of JPEG images estimated to around 80/100 of Quality factor for optimal high quality and a lower quality factor for medium to low quality images.

It is worth discussing other spectral analysis techniques dealing with signal and image

processing, namely the Wavelet Transform. Unlike the Cosine and Fourier Domain, Wavelets encompass both frequency and time information of a signal. The Sine wave, which is the basis of Fourier analysis, and the Cosine wave both exhibit a smooth and predictable pattern, while Wavelet analysis breaks up the original signal into a scaled and shifted version focusing on trends and peaks in the signal. This uniqueness allows for an alternative method to examine signals.
While spectral analysis techniques, in general, exhibit distinctive power at breaking down and analyzing images, which are nothing more than two dimensional signals, they do have limitations in detecting image forgeries. These include only having high detection accuracy on uncompressed images while exhibiting poor detection precision on compressed images (i.e. JPEG) with minimal compression [12]. Section 2.5 further discusses the correctness of an example using spectral analysis techniques.

Wavelets are also used to form new compression schemes for digital images. While the JPEG standard, using Discrete Cosine Transforms (DCTs), is the most popular and widely used format on the web and by digital cameras, the Discrete Wavelet Transform (DWT) is currently being researched and forms the basis for the JPEG2000 format. DWT compression in digital images provides a new and unique approach at obtaining images with smaller files sizes and at the same time having better quality. While the International Standards Organization has finalized the JPEG2000 DWT format in late December 1999 (Johnson, 2004), it is not widely supported in web browsers, digital cameras, and image manipulation software]. The JPEG DCT standard is still the most widely used and supported medium for digital images.

3. CORRECTNESS AND PERFORMANCE OF THE PRESENTED DETECTION METHODS

The tampered portion, in this example, has been magnified for better analysis. While a similar pattern arises in this magnified portion, as witnessed in the blocked regions, no firm conclusion signifies that image tampering has occurred. “Off the shelf” convolution masks are not ideal to detect image tampering because they lack the ability to make a solid conclusion in regard to whether an image has been tampered with. They may be good to use in extending other more conclusive methods, but the several test images analyzed by Lukas [10] .

The results of using the Robust Matching Technique are very promising with regard to the few test images analyzed [13], Similar to the Exact Match Technique, the areas determined to be duplicate copies are shaded with a color that corresponds to the different shift vectors. Everything else not matched is colored black
Overall, the Robust Match Technique is worthy of much praise in detecting copy-move image forgeries. While several of the test images exhibited small areas of false positives, it is still an excellent technique to use as a baseline in the detection of copy-move forgeries. A false positive is common on flat backgrounds that contain very similar color and texture patterns, such as the sky. Therefore, human examination is obviously necessary to interpret the results of any algorithm designed to detect image forgeries [13].

The methods presented here focus mainly on the detection of copy-move forgeries saved in any image format as well as copy-create forgeries saved in uncompressed formats, i.e. TIFF. Much work still needs to be performed with respect to copy-create forgeries saved in

the very common and widely used JPEG image format.

Section 2.4 discussed previously proposed forgery detection methods and their correctness at detecting various types of image forgeries. Several other methods in image processing should be further investigated to determine which are better suited at detecting image tampering. These methods include an analysis of the Luminance and HSV (Hue-Saturation-Value) intensity levels of an image. Also, various custom filtering masks should be investigated to capture their flexibility in filtering an image using customizable parameters. Finally, in-depth analysis of the JPEG compression algorithm is a viable research path since it is the foundation of detecting "hidden" information about an image not easily detected by the human eye.

3.1 Detection of tampering based on analysis of Luminance levels

The recognized brightness levels in an image are measured by the luminance [14]. A kind of conflict may show in the copied and pasted areas because these two images are captured with different cameras and obviously with different lighting. The examined areas in the forged image have different luminance levels although these areas have almost an identical space away from the lens. The creation of the forged images and used resources fully rely on the person's skills, which is the base of this study. "Auto-brightness" features available in the latest image processing software versions have made it even easier for beginner users to create forgery. Fig. 7(b,c) shows the original test image in Fig. 7(a) with luminance levels at both extremes on the scale. The 7(b) image has a low level of luminance while the 7(c) image has a much higher level.

Fig.7. Example of changes in luminance levels

3.2 Detection of tampering based on Hue-Saturation-Value (HSV) levels

As in the previous section dealing with the luminance of an image, an analysis method based on the Hue-Saturation-Value (HSV) levels of an image follows. The Hue of a color is best described as the "tint" [14]. Saturation or "shade" is defined as the level of how pure or intense a color is [14]. Value is the level of brightness (luminance) of a color or how light or dark it is [14]. Intuitively, if an area or areas throughout an image are copied and pasted from different sources, the color and brightness, as captured from each respective camera, may be slightly different.

Thorough analysis of HSV levels helps to determine this. Figure 2.18 shows an example of changes in HSV levels of Fig. 7(a).

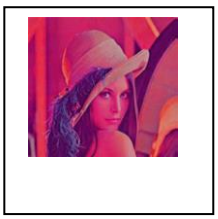
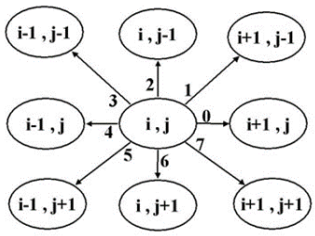


Fig.8. Example of a change in HSV levels to Fig. 7(a)

3.3 Detection of tampering based on alternative filtering masks

As discussed in Section 2.1., Lukas has looked at several edge detectors based on the Sobel and Prewitt masks. This technique of image filtering is officially categorized as pixel-group or spatial domain filtering it's used to detect edges as presented in Section 2.1, other interesting information can also be gathered from an image, such as the low or high pass filtered version. These filtering approaches give an alternative way to view an image and therefore may uncover small anomalies introduced from image tampering. The power to create customized masks may prove to be of some interest in detecting image forgeries, or providing further validation that one has occurred.

Spatial domain filtering deals with calculating a pixel value based upon its surrounding pixels. This type of "pixel group" processing provides a way to show trends in an image, such as brightness levels across particular areas[61]. In the 3 x 3 case, every pixel is evaluated with its eight neighboring ones. Below is an abstract representation of each pixel and its eight neighbors:



Where $x_{i,j}$ is the pixel at location i,j in image X and the rest of the letters represent $x_{i,j}$'s eight neighbors. The integer values of each pixel are extracted and manipulated with a convolution kernel. Formally, the values obtained from pixel $x_{i,j}$ and its eight neighbors are multiplied by their respective convolution kernel coefficients and then the summation over all nine is taken. Finally, this value is then divided by the total number of elements summed. This returned number is now the new value for the pixel $x_{i,j}$. This same technique is applied to every pixel in the image, with all pixels eventually assuming the representation $x_{i,j}$. Care is taken at the image boundaries to only use those pixels that would fall within the image. Below is a depiction of the convolution kernel, which maintains consistency throughout the entire filtering process:

$$\begin{matrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{matrix}$$

The following is a representation of the summation of pixels xi-1,j-1 through xi+1,j+1 with the respective convolution kernel:

$$\text{Output pixel } x_{i,j} = [(x_{i-1,j-1}(k_{11}) + x_{i-1,j}(k_{12}) + x_{i-1,j+1}(k_{13}) + x_{i,j-1}(k_{21}) +$$

$$x_{i,j}(k_{22}) + x_{i,j+1}(k_{23}) + x_{i+1,j-1}(k_{31}) + x_{i+1,j}(k_{32}) + x_{i+1,j+1}(k_{33})) / 9]$$

Intuitively, the result of the above operation emphasizes the trends in an image, particularly abrupt pixel variability as witnessed in edges and, more importantly, tampered areas. This is because a pixel's eight neighbors is averaged and used to determine its new value. Conversely, with processing the whole image together, effectively a block size equal to the size of the complete image, the power to see any trends or suspicious areas may be lost. This is due to the weighted average approach used by spatial domain processing [15]. Block Based Processing with a relative block size to a single pixel could lend clues or provide further justification that a particular area in question is victim to image manipulation.

3.4 Detection of tampering based on the JPEG compression scheme

"Block Based Processing" classification is ensured when an image is divided into sub-parts or identical squares to execute processing. This technique is similar to that described in section 4.3, but the difference is that each block is considered a separate sub-image. This method is analogous to a recursive type process, with the sub-processing resembling a "divide and conquer" approach. Block Based Processing is useful because the calculations performed are influenced by only the information present in that particular block.

Block Based Processing is important in image processing, specifically image compression. The International Standards Organization (ISO) and International Electro-Technical Commission (IEC) of Joint Photographic Expert Group (JPEG) set the compression standard forward the usage of images Discrete Cosine Transform (DCT) scheme [16]. The DCT area is utilized to transform a signal into coefficient values with the capability to execute truncating and rounding operations, therefore permitting compression of this signal to take place. The JPEG compression process starts by calculating the DCT of each unique 8 x 8 blocks, B kl, in the image based on the following formula [13]:

THE JPEG COMPRESSION ALGORITHM CREATES A TYPE OF :

$$D_{ij} = \sum_{k,l=0}^{7} a_{kl}(i,j)B_{kl},$$

Where :

$$a_{kl}(i,j) = \frac{1}{4}w(k)w(l)\cos\frac{\pi}{16}k(2i+1)\cos\frac{\pi}{16}k(2j+1) \text{ and } w(k) = \frac{1}{\sqrt{2}} \text{ for } k = 0$$

$$D_{ij} = round\left(\frac{D_{ij}}{Q_{ij}}\right), i, j \in \{0,1,2,3,4,5,6,7\}$$

And $w(k) = 1$ otherwise.

Matrix D, that has 64 DCT coefficients, is quantized using a quantization matrix Q [5813

The quantized coefficients, Di,j , are organized in a zigzag order and encoded.

Huffman Algorithm introduced into what creates the JPEG file [13]. Decomposition works alike but in reverse order. An integer value is gained which lets an image to be compressed by rounding the ratio above. A threshold is established to define what integer values should efficiently be discarded. The parts to be discarded are cautiously calculated based on a "Quality Factor", which is a reference number between 0 and 100 [17] The image has a better quality when its less compressed which indicated that the quality factor is higher. A trade-off between file size and image quality is constantly essential in this type of lousy compression.

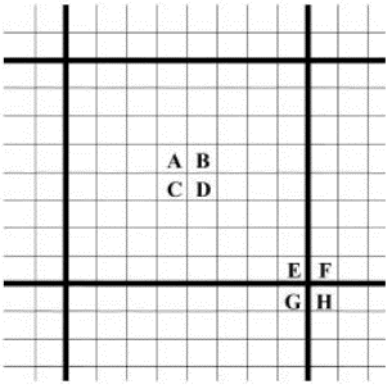
A JPEG image can either be color or grayscale. The given operations encode pixel values that are generally in the 0 to 255 range (8-bit). In the situation of grayscale images, a sole 8-bit number shows the level of gray in each pixel. Color images utilize identical boundaries but contain three 8-bit numbers, one for the Blue, Green, and Red channels.

This allows for the creation of a 24-bit color image. [16] The analysis in this section applies for all types of JPEG images and several forensics methods apply without taking the color type into consideration.
A distinguish event happened when the JPEG scheme is used to seriously compressed an image which makes it obvious for the subsequent information loss and the 8 x 8 blocks resulting from the DCT function to be noticed .

The blocks are simply different in this image and show the impacts of DCT compression. "Fingerprint" that may show alteration and uses an expected scheme and analyzed image to present a potential progress in detecting image tampering.

If two images are used to create a forgery, it is likely that both have different levels of

compression, specifically the "Quality Factor" discussed previously may be different in both cases. Also, it is likely that resizing, rotating, or cropping was performed on the tampered portion to ensure it blends in with the rest of the image. Therefore, the compression algorithm may leave behind some possible clues. Figure 2.20 depicts an example of the above conjecture. Here, the higher compressed (QF = 5) image from Figure 2.19 (Image A) and the better quality (QF > 70) original from Figure 2.17(a) (Image B) are merged together to form forged Image C. This manipulation was accomplished by simply performing a copy and paste operation. Image A was positioned accurately over Image B, as displayed in the circled area, and then returned to the normal magnification. The result at normal magnification is almost indistinguishable to the human eye. The different levels of compression present should be noted, specifically that seen in the woman's eyes. Her left eye was part of Image B, while the copied portion, Image A, contains her right eye at much lower quality. When looking at the resulting Image C, one would not think anything is suspicious unless prior knowledge of tampering was known. This simple simulation shows the power of attempting to do an analysis of the compression levels used in a JPEG image.
A technique has been previously used in determining if a BMP image in raw format, i.e. one without any compression, has been previously JPEG compressed (Fan and de Queiroz, 2003). By breaking up an image into disjoint 8 x 8 blocks, analysis can be performed to determine if a "fingerprint" exists that will signify that the image has, in fact, been previously JPEG compressed. An intuitive approach is to calculate sample differences from within a block and again at the blocks boundaries [17]. Figure 2.21 shows an abstract representation of an 8 x 8 block with the pixel values marked used in calculations.



Solving the following equations calculates the differences (Fan and de Queiroz, 2003):

$$Z'(i,j) = |A - B - C + D| \qquad Z''(i,j) = |E - F - G + H|$$

Finally, the histograms of Z ' and Z " are computed. The resulting information is analyzed to look for a discrepancy in pixel patterns. If there appears to be differing histogram results over multiple blocks, it is determined that the image has been previously compressed. Respective histograms that are extremely similar over multiple blocks warrant an image that has not been previously compressed [17].

Further analysis of JPEG images exist which build upon the previous paragraph.

This includes the estimation of the primary quantization table from an image that has been JPEG compressed twice [13]. By again analyzing each 8 x 8 block of an image, statistical determination can be made whether an image has been double compressed. The key here is to understand what occurs when an image has been compressed twice, and then take advantage this phenomenon. When an image is compressed for the first time, corresponding pixels are the result of rounded integers. When the second compression occurs, these rounded values are used again to compute with the second quantization table, Q2. By analyzing the histograms of these quantized coefficients, an attempt is made to find a pattern which leads back to the original quantization table, Q1. This technique is useful at blindly detecting images that have been watermarked [13]. Most watermarking programs take a "cover image," insert hidden information, and then save the image again, hence yielding a double compression. Estimating the primary quantization table assists in determining the watermark used.

The methods discussed in this subsection deal with performing analysis of an image with respect to JPEG compression. Much information can be determined from this type of analysis and could be promising at its ability to detect image manipulations. It is possible for an image tampering expert to perfect a technique to create near flawless forgeries, concentrating on covering their tracks of "hidden" attributes of an image, such as JPEG compression blocks. But this area is still worthwhile and should be investigated further.

4 SUMMARY

This paper discussed the current state of research in terms of digital image forensics. While digital watermarking has been the method of choice to safe-guard one's images from manipulation and to secure a copyrightable image, it has been difficult to determine if an image of unknown origin is authentic. Several techniques exist that touch the surface of the subject. These hold some sound results, as previously discussed, but further analysis is needed to determine the best and most efficient method to detect image forgeries. The Exact and Robust Matching algorithm to detect copy-move image forgeries shows potential as a tool already exists to detect this type of tampering [13]. But the areas of copy-create forgeries is in need of more research. First and Second Order convolution filters as well as preliminary spectral analysis approaches analyzed by Lukas returned discouraging results [10]. The recent results of Farid and Popescu take spectral analysis approaches further by devising a useful tool for detecting image forgeries. As with all of the techniques presented, close human interpretation is needed and there appears to be no "silver bullet" in terms of a detection scheme. Various methods available in the image processing toolkit will need to be applied to this area with results closely scrutinized. An interesting approach that requires more investigation is one that looks at the JPEG compression scheme of an image. Even though a forgery may appear to be flawless to the human eye, small underlying details of the JPEG "fingerprint" could be its Achilles' heal.

REFERENCES

1. Lu, W., Sun, W., Huang, J.-W. and Lu, H.-T. (2008). Digital image forensics using statistical features and neural network classifier. In International Conference on Machine Learning and Cybernetics., vol. 5. IEEE, 2831–2834.
2. Farid, H. (2009). Image forgery detection. Signal Processing Magazine, IEEE. 26(2),16–25.
3. Mahdian, B. and Saic, S. (2008b). Blind methods for detecting image fakery. In 42nd

Annual IEEE International Carnahan Conference on Security Technology (ICCST). IEEE, 280–286.

4. Mahdian, B. and Saic, S. (2010). A bibliography on blind methods for identifying image forgery. Signal Processing: Image Communication. 25(6), 389–399.
5. Sencar, H. and Memon, N. (2008). Overview of state-of-the-art in digital image forensics. Algorithms, Architectures and Information Systems Security. 3, 325–348.
6. Kirchner, M. (2010). Linear row and column predictors for the analysis of resized images. In Proceedings of the 12th ACM workshop on Multimedia and security. ACM,, 13-18
7. Mahdian, B. and Saic, S. (2008a). Blind authentication using periodic properties of interpolation. IEEE Transactions on Information Forensics and Security. 3(3), 529–538.
8. Popescu, A. C. and Farid, H. (2005). Exposing digital forgeries by detecting traces of resampling. IEEE Transactions on Signal Processing. 53(2), 758–767.
9. Farid, H. and Lyu, S. (2003). Higher-order wavelet statistics and their application to digital forensics. In Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'03)., vol. 8. IEEE, 94–94.
10. Lukas, J. "Digital Image Authentication Using Image Filtering Techniques." Proceedings of 15th Conference of Scientific computing
11. Luong C. M. "Introduction to Computer Vision and Image Processing," Department of Pattern Recognition and Knowledge Engineering, Institute of Information Technology, Hanoi, Vietnam, May 4, 2004.
    http://www.netnam.vn/unescocourse/computervision/computer.htm.
12. Popescu, A. C. and Farid, H. (2004). Exposing digital forgeries by detecting duplicated image regions. Technical report.
13. Fridrich, A. J., Soukal, B. D. and Luk, A. J. (2003). Detection of copy-move forgery in digital images. In Proceedings of Digital Forensic Research Workshop
14. Sachs, J. Digital Image Basics. Digital Light & Color. 1999
15. Baxes, G. A., Digital Image Processing: Principles and Applications. New York: John Wiley & Sons, Inc, 1994.
16. Saha, S. "Image Compression – from DCT to Wavelets: A Review," May 28, 2004 http://www.acm.org/crossroads/xrds6-3/sahaimgcoding.html.
17. Fan, Z. and de Queiroz, R. L. (2003). Identification of bitmap compression history: JPEG detection and quantizer estimation. IEEE Transactions on Image Processing.12(2), 230–235