A Novel Approach To Detection and Evaluation of Resampled Tampered Images

Amrit Hanuman *amrit2025@hotmail.com*
Department of Electrical and Computer Engineering
The University of the West Indies
St. Augustine, Trinidad

Azim Abdool *azim.abdool@sta.uwi.edu*
Department of Electrical and Computer Engineering
The University of the West Indies
St. Augustine, Trinidad

Akash Pooransingh *akash.pooransingh@sta.uwi.edu*
Department of Electrical and Computer Engineering
The University of the West Indies
St. Augustine, Trinidad

Aniel Maharajh *admaharajh@outlook.com*
Department of Electrical and Computer Engineering
The University of the West Indies
St. Augustine, Trinidad

Abstract

Most digital forgeries use an interpolation function, affecting the underlying statistical distribution of the image pixel values, that when detected, can be used as evidence of tampering. This paper provides a comparison of interpolation techniques, similar to Lehmann [1], using analyses of the Fourier transform of the image signal, and a quantitative assessment of the interpolation quality after applying selected interpolation functions, alongside an appraisal of computational performance using runtime measurements. A novel algorithm is proposed for detecting locally tampered regions, taking the averaged discrete Fourier transform of the zero-crossing of the second difference of the resampled signal (ADZ). The algorithm was contrasted using precision, recall and specificity metrics against those found in the literature, with comparable results. The interpolation comparison results were similar to that of [1]. The results of the detection algorithm showed that it performed well for determining authentic images, and better than previously proposed algorithms for determining tampered regions.

Keywords: Image Forgery, Resampling, Interpolation, Passive Detection.

1. INTRODUCTION

The availability of affordable high resolution digital cameras and powerful image editing software has increased the ease of digital image tampering [2]. Digital forgeries can be mistaken for authentic images as they seldom leave visual clues of the alteration. People are confronted with digital forgeries recurrently through mainstream media outlets [3]. These forgeries are created to stir controversy, invoke emotion or sell products. In resampled images, interpolating functions are directly responsible for the output image quality when altering them. Image editing tools are continuously being developed and improved to hide any sign of tampering. Consequently, the validity of images used as evidence in the judicial system, news media and scientific journals can be questioned [2]. It is essential to verify the integrity of images to avoid deception.

There are three main classifications of digital forgeries: copy-move, splicing and retouching [4]. In copy-move forgery (Figure 1), part of a source image is copied and pasted at another location in the same image. Image splicing (Figure 2) creates a new image using composites of two or more images, while retouching (Figure 3) deals with changes to image features such as color, textures or blurs. Each forgery type can be detected using different methods.

Splicing can be considered the most malicious of the forgery types where resampling is used to change a foreign image's dimension or angle such that it looks natural on the host image.

FIGURE 1: An Example of Copy-Move Forgery [4].

FIGURE 2: An Example of Splicing Forgery [4].

FIGURE 3: An Example of Retouching Forgery [4].

Resampling is required since the new image's coordinate points will not always line up with the old points. Resampling is defined as “the process of transforming a discrete image which is

defined at one set of coordinate locations to a new set of coordinate points" [5]. There are two steps to resampling (Figure 4): interpolation and sampling.

Interpolation uses known data to estimate values at unknown points. It is the process of fitting a continuous function to the discrete points in the digital image [5]. The discrete data is convolved with a continuous interpolating function to create a continuous signal. This continuous signal is then sampled to obtain the values at the new coordinate points [5].

Image interpolation has many applications ranging from simple viewing and editing online images, to medical image processing [6, 1, 7, 8]. In image editing software, the three commonly used interpolating functions are nearest-neighbor, bilinear and bicubic interpolation. Additionally, some include the lanczos3 function (see Figure 5). These are categorized as non-adaptive algorithms since all pixels are treated equally. Adaptive algorithms rely on the intrinsic image features and contents such as sharp edges or smooth texture [6]. The choice of interpolation function produces different results, and successive interpolation diminishes image quality except for rotations that are multiples of 90° or shifts over integer pixel values [1].

Interpolation affects the underlying statistics of the image by introducing specific correlations of pixel values into the image [2]. The detected correlation can be used as evidence of tampering. Farid [3] defined pixel-based forensic tools as those that detect statistical anomalies introduced at the pixel level. Detection methods are classified into two types: active and passive. Active methods require prior knowledge of the original image whereas passive methods do not [4].

An example of a current active detection method is digital watermarking. A digital code (a watermark) is embedded into the image at the time of recording through specially equipped devices. If the image is tampered with, this code will be altered. The image is authentic if the extracted code matches the inserted code, after transmission. Passive or blind methods are favored over active methods since the original image is often unavailable. These are the methods that detect the statistical anomalies in the image [9].

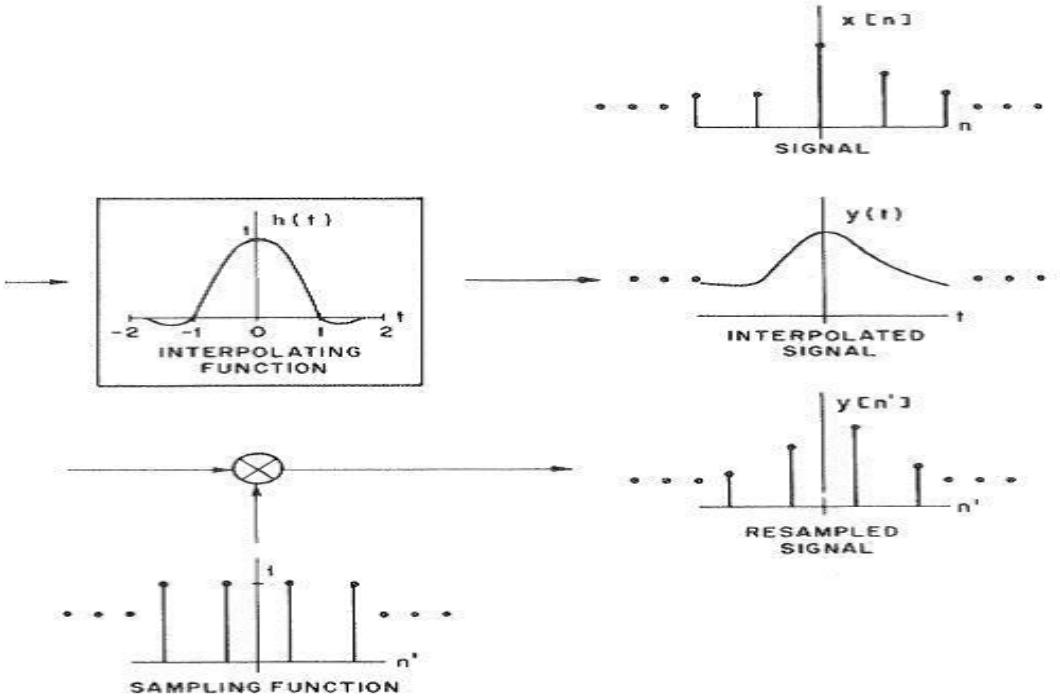

FIGURE 4: The Resampling Process: A discrete signal is fitted with a continuous function (interpolated) to create a continuous signal that is sampled to get the new discrete values. [5].

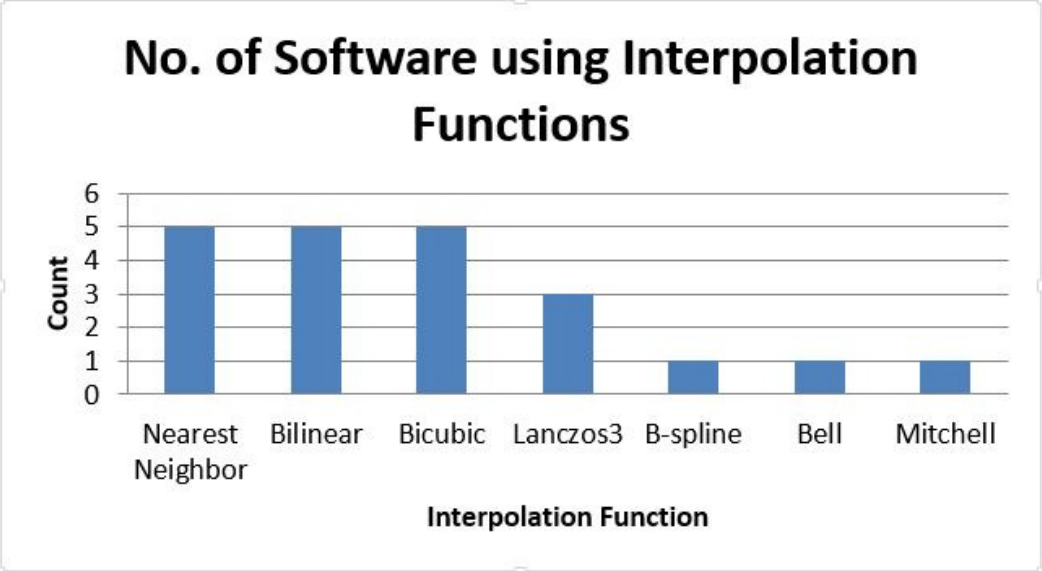

FIGURE 5: Interpolation Functions Found in 5 Popular Image Editing Software.

This paper presents a brief review of common interpolation functions. A novel algorithm is proposed for automatic detection of tampered images. This paper does not include estimation of resampling factors, but presents an evaluation method to determine the detection efficiency. The organization of this paper is as follows. Section 2 reviews previous works done on comparison and detection of interpolation. The proposed detection algorithm is described in Section 3. The details of the implementation are provided in Section 4, with results in Section 5. Section 6 discusses the results and future works, followed by the concluding remarks in Section 7.

2. RELATED WORK

2.1 Comparison of Interpolation Techniques

A comparison of different interpolation methods, examining the frequency response of the interpolating functions, was done in [5]. It was noted by Lehmann, Gonner and Spitzer [1] that the appearance of the image after resampling should also be taken into account. They presented a comprehensive survey of interpolation methods, investigating the Fourier analysis, qualitative and quantitative error, computational complexity and run time measurements. Their comparison was extended to other interpolation methods in [10]. Following is a brief review of the different interpolation techniques.

2.1.1 Ideal Interpolation

A continuous band-limited signal can be sufficiently reconstructed from samples if it is sampled at the Nyquist frequency or higher. [5]. However, sampling produces infinite replicas of the continuous spectrum in the frequency domain. To recover the band-limited signal, only one of these replicas is needed. The interpolating function used should be an ideal low-pass filter. Multiplying the continuous spectrum with the rectangle function in the frequency domain is equivalent to convolving the discrete data with a sinc function in the spatial domain [1]. The function in the spatial domain is given by Equation (1).

$$h_{ideal}(x) = \frac{\sin(\pi x)}{\pi x} = sinc(x) \tag{1}$$

2.1.2 Nearest Neighbor Interpolation

Nearest neighbor is the simplest interpolation technique, requiring the least computational processing time. Only one pixel is considered for each new coordinate point. The value of the

pixel at the new coordinate point of the output image is equal to that of the pixel nearest its corresponding point in the source image. The interpolation is done by convolving the sampled image with a rectangle function in the spatial domain described by Equation (2) [1]. In the frequency domain, this is a sinc function, a poor low pass filter. The resultant image is pixelated or blocky [1].

$$h_{nearest(x)} = \begin{cases} 1 & 0 \leq |x| < 0.5 \\ 0 & elsewhere \end{cases} \tag{2}$$

2.1.3 Bilinear Interpolation

To estimate the value at the new point, bilinear interpolation considers the closest 2x2 neighborhood of known pixels. This value is a weighted average, where the points are weighted based on the distances to the diagonally opposite point [6, 1]. It is a combination of linear interpolation along the horizontal and vertical axes. Bilinear interpolation is a convolution of the discrete data with a triangle function expressed in Equation (3). There is attenuation of high frequency components and the aliasing of data beyond the cutoff point into the low frequencies. Bilinear interpolation produces a smoother output image [5].

$$h_{bilinear}(x) = \begin{cases} 1 - |x| & 0 \leq |x| \leq 1 \\ 0 & elsewhere \end{cases} \tag{3}$$

2.1.4 Bicubic Interpolation

Bicubic interpolation considers the closest 4x4 neighborhood of known pixels for estimation. Closer pixels are given higher weightings. It assumes that the image follows a polynomial in this neighborhood. The interpolation function is described in Equation (4). Lehmann, Gonner and Spitzer [1] suggest using an optimal value of a = 0.5. Bicubic interpolation performs better than bilinear interpolation [7].

$$h_{bicubic}(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1 & 0 \leq |x| \leq 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & 1 \leq |x| \leq 2 \\ 0 & elsewhere \end{cases} \tag{4}$$

2.1.5 Lanczos3 Interpolation

Lanczos3 is a windowed sinc function equivalent to the multiplication of the sinc function by a rectangular function in the spatial domain [1]. It is described by Equation (5) [6]. The truncation introduces ringing artifacts. These artifacts are reduced if the edges of the window coincide with a pair of the sinc function's zero-crossings [7].

$$h_{lanczos3}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x} \times \frac{\sin(\pi \frac{x}{3})}{\pi \frac{x}{3}} & |x| < 3 \\ 0 & elsewhere \end{cases} \tag{5}$$

2.2 Detection of Interpolation Techniques

Two of the more common detection methods used were the Expectation-Maximization method [2] and the properties of the second difference statistics method [11, 12]. Popescu and Farid [2] found that specific samples in a resampled sequence can be written as a linear combination of their neighboring samples. The Expectation-Maximization method was applied to determine the probability that the samples were correlated and find the specific form of the correlation. This was because the probabilities were arranged in periodic patterns, shown in its Fourier transform, which depended on the resampling rate.

Gallagher [11] introduced a novel algorithm for interpolation detection and estimation in JPEG images. The second derivative signal of interpolated images was proven to have a periodicity related to the resampling factor. The algorithm worked by detecting patterns in the pixel-wise difference. First, the second derivative of each row of the image was computed using Equation (6) followed by the average across all rows.

$$Sp(i,j) = 2p(i,j) - p(i,j+1) - p(i,j-1) \tag{6}$$

The DFT magnitudes of this trace were then plotted. Distinct peaks would be displayed if the image was interpolated. If no peaks occurred, the image was considered authentic. Where peaks occurred, the resampling factors could then be estimated using Equation (7) if the peak occurred at a normalized frequency greater than or equal to 0.5, or Equation (8) for normalized frequency less than 0.5.

$$\text{Resampling Factor} = \frac{1}{(\text{Normalized Peak Frequency})} \tag{7}$$

$$\text{Resampling Factor} = \frac{1}{(\text{1-Normalized Peak Frequency})} \tag{8}$$

Gallagher [11] noted that the algorithm's performance was better for low order interpolators such as bilinear and bicubic, but decreased as the order of interpolator increased. Gallagher [11] was unable to detect resampling factors of 2.0 since phase was preserved. Due to JPEG compression, peaks occurred at normalized frequencies of 1/8, 1/4, 3/8, 5/8, 3/4 and 7/8. These were ignored for all images, however if the image was resampled at a rate corresponding to those frequencies, it was difficult to determine.

Similar techniques to [11] were described in [13] for resampling detection using the second difference properties. The concept follows that when a sequence of P samples is resampled by a factor of M / N, there are now PM / N samples. If $M / N \geq 2$, there is at least one interpolated sample between a pair of original samples. Taking the second difference of the sample between the original pairs will produce a zero at that location. These zeros are periodic and do not occur in untampered sequences. Prasad and Ramakrishnan [13] generated a binary sequence from the second differences where the result was true if the second difference equaled zero as given by Equation (9). The DFT of this sequence would show distinct peaks indicating the presence of periodic zeros.

$$p[k] = \begin{cases} 1 & if|x''[k]| = 1 \\ 0 & otherwise \end{cases} \tag{9}$$

Alternatively, the zero-crossings of the second difference of the image was found before taking the DFT using Equation (10).

$$p[k] = \begin{cases} 1 & if x"[k] > 0 \text{ and } x"[k+1] \leq 0 \\ 1 & if x"[k] < 0 \text{ and } x"[k+1] \geq 0 \\ 0 & otherwise \end{cases} \tag{10}$$

A major weakness in both the algorithms in [11] and [13] is that they cannot detect interpolation if the image was rotated. Mahdian and Saic [14] proposed a new algorithm which was able to detect rotation and skewing using the radon transform. They selected a region of interest by dividing the image into overlapping blocks before applying the method.

Inspired by [11], Wei et al. [15] proposed an algorithm for rotation angle estimation. Nevertheless, it was their efforts in detection that was of interest. They compared the plots generated in [11], where the average of the second difference of each row was taken before computing the DFT, to one where the DFT was computed before taking the average. They found that when an image was scaled then rotated, the curve generated by the latter produced distinct peaks while the former did not. This was just a change in steps from [11] to achieve rotation tolerance. Wei et al. [15] also divided the image into overlapping blocks before applying their method.

Birajdar and Mankar [9] worked towards improving the algorithm in [13] by including estimation as done in [11]. Their input image was the Y component of an YCbCr image but tests were done using other color spaces. Their results showed that detection was better for the green component and grayscale image than it was for the Y component. Additionally, their algorithm was able to detect resampling factors of 2.0 unlike those by [11, 14].

3. THE PROPOSED DETECTION ALGORITHM: ADZ

The detection algorithm implemented aimed at detecting locally tampered regions, taking the Averaged Discrete Fourier transform of the Zero-crossing of the second difference of the resampled signal (ADZ). To achieve local tamper detection, the image was divided into blocks. The input image was the green plane of the tampered image, since [9] showed this plane worked better than other color spaces in resampling detection. Image forgeries often involve scaling the alien image to match the size of the objects in the image being tampered. Rotation is done for alignment. The technique which found the DFT then average of the image block used in [15] was chosen for its rotation tolerance. The zero-crossings of the second difference were incorporated as it was shown in [9] to detect resampling factors of 2.0. The algorithm's steps are outlined as follows for an $M \times N$ image:

- Compute the second difference: The green component G of the RGB image was taken and the second difference of each row was computed.

- Generate the binary image: Using (10), the zero-crossings of the second difference was found.

- Divide into blocks: The binary image was divided into overlapping blocks, each sized 64×64 with overlapping area of 32 pixels between each pair of blocks, row-wise and column-wise.

- Block-wise forgery detection: For each block, the DFT of all rows was generated. The average of the DFT magnitudes across the rows was found.

- Peak Detection: A threshold-based peak detector was used to find all peaks in the averaged spectrum that were 1.625 times the average in that array. If peaks existed in the averaged spectrum, the message "Interpolation Detected" and output image showing the interpolated block was displayed. If no peaks existed, the message "No Interpolation Detected" was displayed.

4. METHODOLOGY

4.1. Comparison of Interpolation Techniques

The manuals of five popular image editing software were found and the most commonly used interpolating functions were determined (see Figure 5). For the chosen interpolation techniques in the investigation, the methods of comparison as done in [1] were followed.

4.1.1 Frequency Analysis

Plots of the different interpolating functions truncated within the interval $-3 \leq x \leq 3$ were implemented using Matlab. The Fourier transformations were performed in order to study the frequency domain characteristics over the frequency band $-12 \leq f \leq 12$. The analysis looked at the

passband for deviation from the ideal, the cut-off point for the amplitude of the slope and the stopband for the occurrence and amplitudes of ripples and sidelobes. Attenuation in the passband leads to blurring while amplification will improve the transformed images sharpness as well as image noise. Aliasing effects are caused by small slopes having high cut-off amplitudes, and the presence of sidelobes and ripples aliases the continuous spectrum replicas into the passband [1].

4.1.2 Interpolation Quality

Taking an original image after histogram equalization a forward and backward transformation was done using the different interpolation techniques to produce the transformed image. The forward transformation was a magnification of 2.0 and the backward transformation was a magnification of 0.5. Thus the image returned to its original size after the transformations. The images within a 25-pixel border from both the original and transformed images were taken (to avoid border effects) for the following sequence of operations:

- The pixel-wise, absolute difference of the original and transformed images was found.
- A threshold was created for the resultant image to show true for all pixels equaling zero. This produced a black and white image where white represented all pixels that remained unchanged after the transformation and black represented otherwise.
- The image quality was measured using the normalized cross-correlation coefficient as shown in Equation (11) [1]. This compares the original and transformed images, and returns a value between 0 and 1, where 1 represents perfect similarity [16].

For the interpolation quality test, classic *512 x 512* grayscale “Lena” and “Living Room” TIFF images were used.

$$C = \left| \frac{\sum_{k,l} s(k,l)r(k,l) - KL\bar{s}\bar{r}}{\sqrt{(\sum_{k,l} s^2(k,l) - KL\bar{s}^2(\sum_{k,l} r^2(k,l) - KL\bar{r}^2)}} \right| \quad (11)$$

4.1.3 Runtime Measurements

The average time taken to transform an image using the different interpolation techniques was done in the Matlab environment. The transformation was a scale by 2.0. The time taken to run the transformation was found in seconds.

4.2. Detection of Interpolation Techniques

Preliminary tests were done to verify the best color space of the input image, determine a suitable block size and peak detection threshold. Global detection of an upsampled image was a good starting point since blocks indicating the tampered region should appear over the entire image. The green plane of the RGB image, grayscale image and Y component of the Ycbcr image were used as the input image to detect global resampling factors of 1.2, 2.0, 2.5 and 5. The green plane was chosen since it produced the most blocks. The threshold value was found empirically. The value of 1.625 maintained a good true negative rate on authentic images while detecting the most blocks on the globally scaled image. Values at the beginning and end of the averaged DFT trace were eliminated from the peak search to remove the high DC component which caused false positives. The block size was required to be large enough to detect periodicity in the pixel values, while not being so large that the tampered region was less than half the area. A greater overlapping area would increase the accuracy of the method but it would be computationally expensive. As recommended in [15], a block size of *64 x 64* pixels with 32 pixel overlap in each direction worked efficiently.

90 digital forgeries were created in Adobe Photoshop CS6 using uncompressed TIFF images from the UCID database [17]. Subsets of 30 images had tampered portions that were in

equivalence classes of: scaled only, rotated only, and scaled and rotated. These were resampled at various factors greater than 1.0. For each subset, bilinear and bicubic interpolation were used on 15 images each, with no post-processing done on the tampered regions. Nearest neighbor and Lanczos3 were not considered. This was because nearest neighbor is known to be the poorest interpolator and therefore less likely to be used in tampering. Lanczos3 was not included since it was known that detection accuracy reduces for higher order interpolators. This was not a problem since it is not widely used and this work sought to address interpolating functions found in commonly available image editing software. An additional 30 images were left untampered, giving a total of 120 images to test the algorithm. All images were a fixed size of *384 x 512* pixels. These test images were restricted to TIFF format to avoid the detection and elimination of peaks caused by JPEG compression.

The following terms were used in the evaluation of the ADZ algorithm: true positive, true negative, false positive and false negative. Where an interpolation-detected block fell on a tampered region, it was a true positive. If the block fell on a region that had not been resampled, it was a false positive. A false negative is counted for every block that should have fell on a tampered region, while a true negative is counted for every block that did not indicate interpolation on authentic areas. To measure the performance of the detection method, the precision and recall metrics were used on tampered images, and the specificity metric was used on authentic images. Precision is a measure of accuracy. That is, how well the algorithm determined a true positive region. Recall corresponds to the true positive rate, while specificity corresponds to the true negative rate [18].

$$\text{Precision} = \frac{TP}{(\text{TP} + \text{FP})} \tag{12}$$

$$\text{Recall} = \frac{TP}{(\text{TP} + \text{FN})} \tag{13}$$

$$\text{Specificity} = \frac{TN}{(\text{TN} + \text{FP})} \tag{14}$$

Where: TP - True Positive; FP - False Positive; TN - True Negative; FN - False Negative.

Since algorithms in previous works were not evaluated in this way, algorithms for [11, 9, and 15] were implemented using a common platform for comparison. The color space, block size and peak threshold identified in the respective papers were used. Preliminary tests were done to determine a range of the averaged DFT searched for peaks, to reduce false positives. For algorithms that considered only global detection, the entire image was used as a single block. The grayscale image was used as the input image in [11] as the color space was not specified. The peak threshold value used in [9] was also not specified but thought to be 10 since the algorithm followed [11], which used the same value. Only 15 bilinear interpolated images, and 15 authentic images were tested.

5. RESULTS

5.1. Comparison of Interpolation Techniques

The results of the Fourier transformations of the interpolation equations are shown in Figure 6. For each interpolation function, the graph shows its shape in the spatial domain (left graph), and the frequency domain (right graph). In the frequency domain, the deviation from the ideal interpolator, which is a rectangular function, is observed. The lanczos3 filter had the closest resemblance to the ideal, producing sharp images, while nearest neighbor showed the least with its high sidelobes, producing blurred images. Judging by the slopes at the cutoff point, and the sidelobes, the bilinear filter resulted in more aliasing than the bicubic filter.

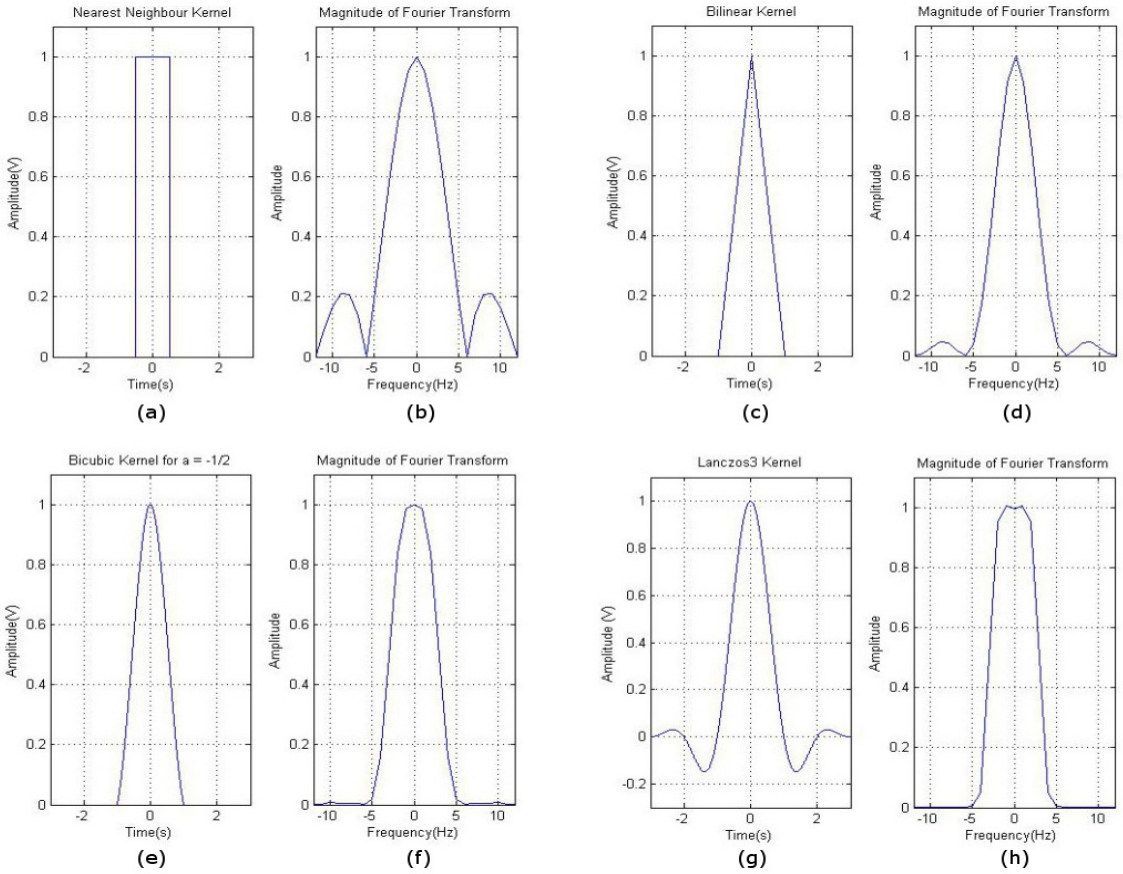

FIGURE 6: Fourier Analysis: (a) Nearest neighbor function in the spatial domain (b) Magnitude of nearest neighbour Fourier transform (c) Bilinear function in the spatial domain (d) Magnitude of bilinear Fourier transform (e) Bicubic function in the spatial domain (f) Magnitude of bicubic Fourier transform (g) Lanczos3 function in the spatial domain (h) Magnitude of lanczos3 Fourier transform.

The interpolation quality test results are shown in Figure 7. The binary images show those pixels that remained the same after the forward and backward transformations in white, and those that were changed in black. The greater the number of white pixels there are, the better the quality of the image. In both test cases, the nearest neighbor interpolator returned the image exactly. The result of bilinear interpolation showed significant loss of edge information, having more pixel values changed after transformation than bicubic interpolation. Lanczos3 interpolation showed the least error, excluding nearest neighbor.

FIGURE 7: Interpolation Image Error: (a) Original Lena image (b) Original Living room image (c) Lena image nearest neighbor error (d) Lena image bilinear error (e) Lena image bicubic error (f) Lena image lanczos3 error (g) Living room image nearest neighbor error (h) Living room image bilinear error (i) Living room image bicubic error (j) Living room image lanczos3 error.

TIFF Image	Nearest	Bilinear	Bicubic	Lanczos3
Lena	1.0	0.9973	0.9994	0.9997
Living room	1.0	0.9918	0.9979	0.999
Average	1.0	0.9946	0.9987	0.9994

TABLE 1: Normalized Cross-Correlation Similarity Measure.

TIFF Image	Nearest	Bilinear	Bicubic	Lanczos3
Lena	0.0101	0.0198	0.0296	0.10
Living room	0.0047	0.0135	0.0208	0.0240
Average	0.0074	0.01665	0.0252	0.062

TABLE 2: Average Time Taken to Perform a Transformation.

This visual quality assessment is supported by the quantitative measurement. Table 1 shows the results of the normalized cross-correlation coefficient. In both cases, the nearest neighbor function had perfect similarity, 1.0. On average, the transformation done using bilinear interpolation had the lowest similarity of 0.9946. The similarity increased with the bicubic function producing a value of 0.9987, and 0.9994 had the lanczos3 function.

A measurement of computational time (in seconds) to perform a transformation using the various interpolation techniques was performed. The results are shown in Table 2. On average, using nearest neighbor interpolation took the least time of 0.0074s. The time taken for bilinear, bicubic and lanczos3 interpolation increased with each one. These values were 0.01665s, 0.0252s and 0.062s respectively.

5.2. Detection of Interpolation Techniques

Figure 8 presents the precision, recall and specificity values. The overall precision is the percentage accuracy of the proposed ADZ algorithm to detect true positive regions, for all 30 images in each tampered subset. The same goes for the overall recall which shows the true positive rate. The bilinear and bicubic precision and recall values are those for the 15 images interpolated by the respective functions, in each subset. The specificity is the average true negative rate found for the 30 authentic images.

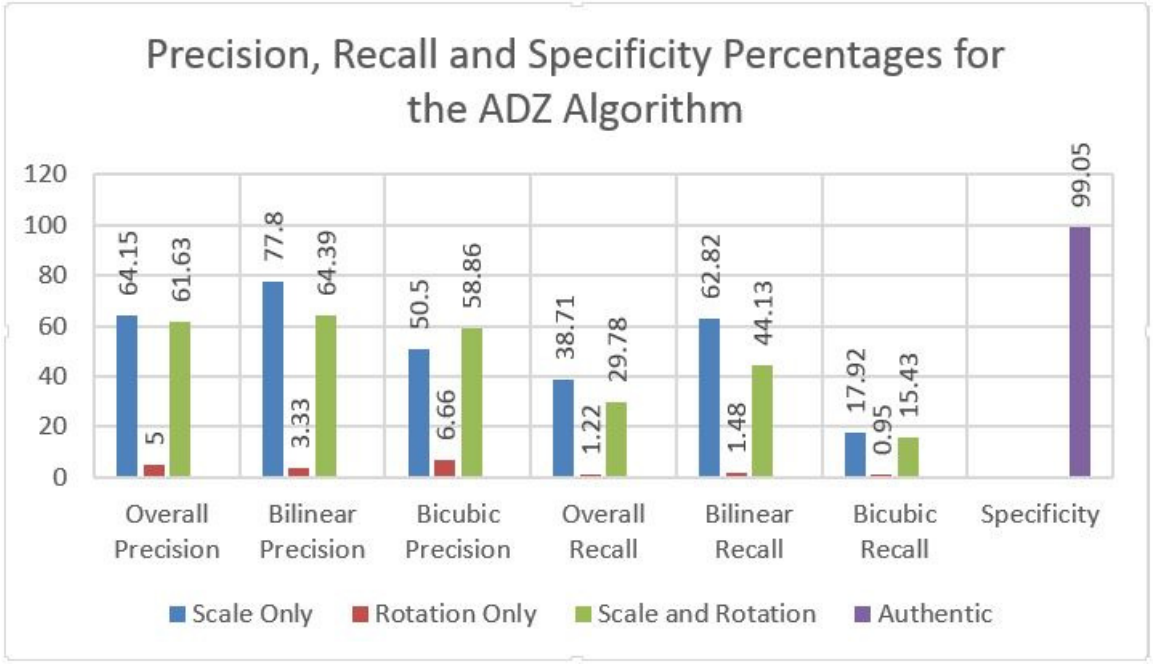

FIGURE 8: Precision, Recall and Specificity Percentages for the ADZ Detection Algorithm.

Figure 8 shows that on average, the ADZ algorithm implemented correctly identified 99.05% of blocks per authentic image as not resampled. That leaves an approximated 1% false positive

rate. The overall precision showed that for tampered, scaled images, only 64.15% of blocks detected were correctly identified as interpolated. This was closely followed at 61.63% in tampered images that had scaled and rotated regions. The algorithm performed poorly in detection of tampered portions that were rotated, with an accuracy of 5%. To be clear, the remainder of these percentages was for false positive blocks. The trend followed for the overall recall values. That is, the algorithm identified 38.71%, 29.78% and 1.22% of the tampered region for scaled only, scaled and rotated, and rotated only images respectively. The remainder of these percentages was for false negative blocks. For all subsets, the precision and recall values were higher for bilinear interpolation than bicubic. The only exception was in the precision of rotated only tampered regions, where bicubic interpolation had 6.66% accuracy and bilinear had 3.33%.

The comparison of the proposed ADZ algorithm to previous algorithms is shown in Figure 9. The proposed algorithm had 77.8% precision in scaled only tampered images, followed by 64.39% in scaled and rotated images, and 3.33% in rotated images. Likewise, the recall values were 62.82%, 44.13% and 1.48% for scaled only, scaled and rotated, and rotated only tampered images. The algorithm had a 99.5% specificity. The algorithms in [11] and [9] both had a specificity of 100%. However, the algorithm in [9] outperformed that in [11] for scaled only tampered images, having precision and recall values of 33.3% as compared to 6.67%. They both failed to detect rotated only, or scaled and rotated tampered regions, having 0%. The proposed ADZ algorithm is better matched by that in [15], which had its highest precision value of 58.67% in scaled and rotated tampered images, followed by 46.67%, and 6.67% in scaled only and rotated only tampered images respectfully. The recall values were 46.67% for scaled and rotated tampered images, followed by 18.32% and 0.83% in scaled only and rotated only tampered images. The specificity value for the 15 authentic images was 98.06%. The proposed ADZ algorithm performed the best as described in Section 6.

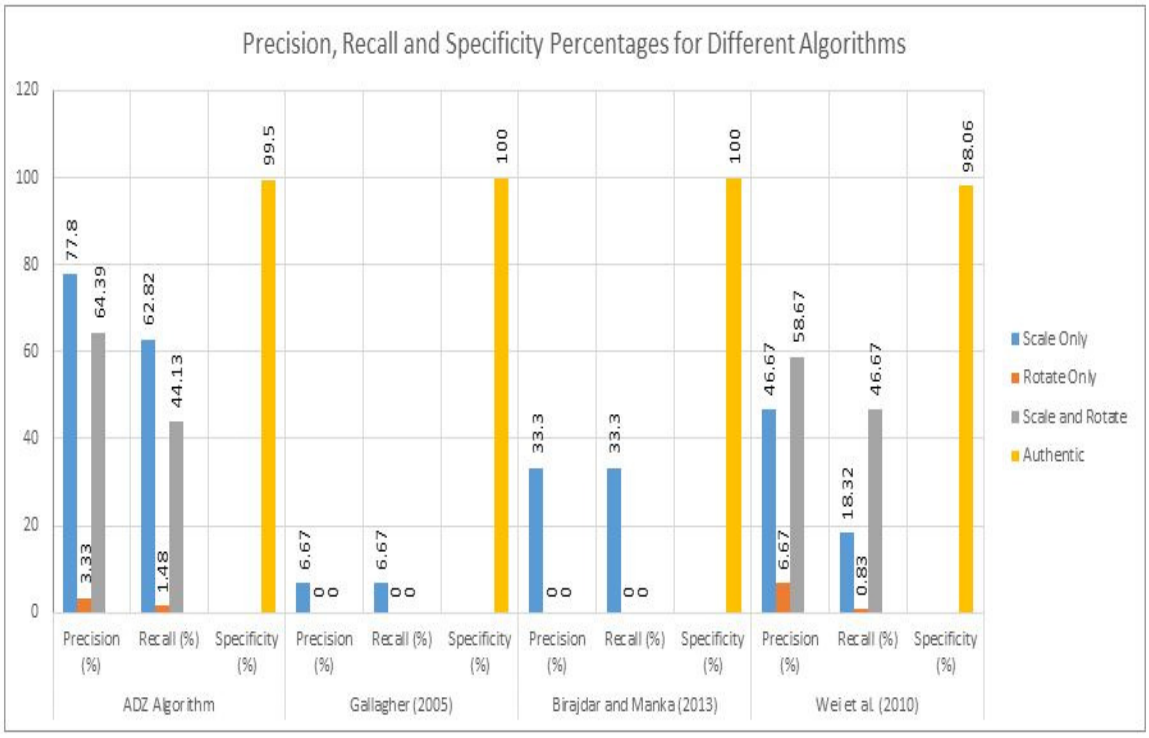

FIGURE 9: Precision, Recall and Specificity Percentages for Different Algorithms.

One of the objectives of the paper was to summarize where on the image the tampered region was detected. Figure 10 shows overlapping blocks in green indicating the detected regions. This is an example of a 100% accurate detection. It was noticed that on authentic images, where a

block fell on an all-white or textured region, false positives were detected. These are shown in Figure 11 and Figure 12 respectfully.

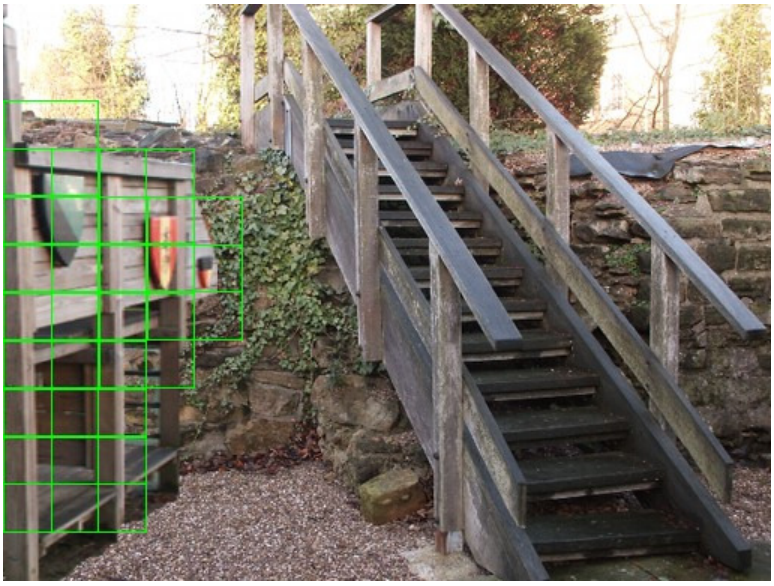

FIGURE 10: Detected Tampered Regions Indicated by the Green Boxes.

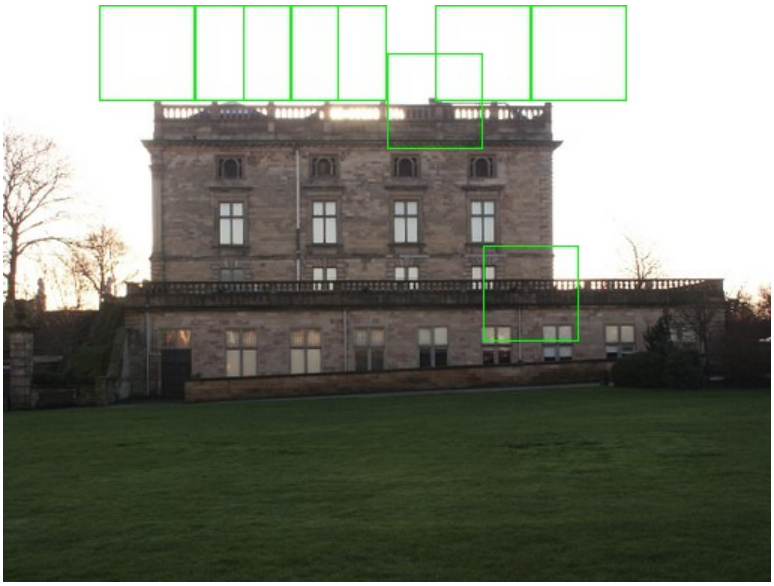

FIGURE 11: False Detected Regions of Solid White Indicated by the Green Boxes.

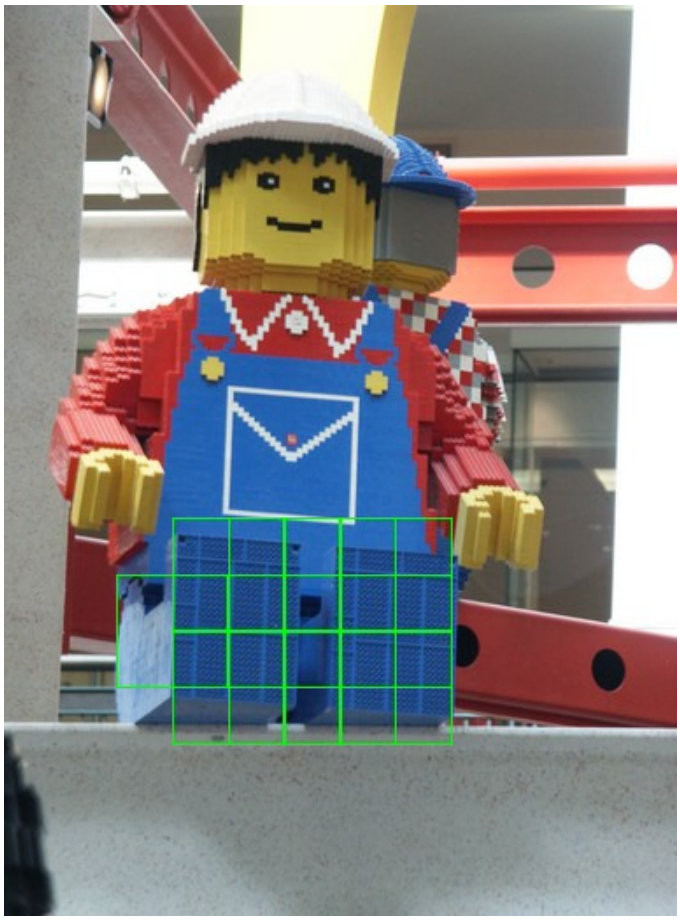

FIGURE 12: False Detected Textured Regions Indicated by the Green Boxes.

6. ANALYSIS

The results of the comparison of interpolation techniques were supported by the literature [1, 19]. The interpolation functions in order of increasing complexity were: nearest neighbor, bilinear, bicubic and lanczos3. The Fourier transformation plots were similar to those produced in [1]. The visual quality tests showed a deviation from what was expected where the nearest neighbor function reproduced an exact image. This was because the backward transform compensated the error introduced by the forward transform. In this case, the quality of nearest neighbor interpolation compared to others was not best represented since nearest neighbor performs the worst. The Fourier plots showed that nearest neighbor interpolation was not an effective filter. The prominent sidelobes would alias the continuous spectrum in the frequency domain, producing blocky images. The nearest neighbor was the simplest interpolator, considering only one pixel for its output, while bilinear considered four and bicubic considered sixteen. The result of the runtime measurement shows that the time taken to perform the same transformation increased for the more complex interpolation techniques. Therefore a better quality image is produced using increasingly complex interpolation techniques at the cost of computational time.

In the detection of authentic images, two cases were found to be incorrectly classified as interpolated; the image consisted of regions that were all-white within the block or had heavy texture. Detecting the white regions can be explained using Equation (6). Since the pixels values were the same, both the first and second difference of neighboring pixels would be zero. The algorithm searched for periodic zeros in the sequence to determine if it was tampered. This would be the same for all blocks of uniform color. The detected block can be ignored by human judgment. Alternatively, the algorithm may be modified to include a pre-processing phase to determine if the detected block had a majority of uniform color pixels using a threshold and if so,

ignore it. However, the computational time to detect the tampered block in the latter would increase. The false detection of textured regions was encountered in [14]. This occurred because textured regions displayed the same periodic patterns as resampled regions. Human judgment determined if the block should be ignored.

The ADZ algorithm worked best at detecting tampered regions that were scaled only, followed by regions that were scaled and rotated, but did not perform well for those that were rotated only. It was explained in [20] that in a rotated image, the periodicity that would exist in a row was skewed onto other rows. When each row was summed, the noise would be strong and no periodicity would be detected [21]. This explains why scaled only regions were detected better than those rotated. To overcome this issue, Qian et al. [20] proposed an algorithm that eliminated rows that were not factors of a constant, before searching for periodicity.

The results of the precision and recall values for bilinear and bicubic interpolation were as expected. Several sources [11, 14] indicated that bilinear interpolation produced larger peaks than bicubic interpolation. When the number of pixels considered producing the interpolated value increased, detection accuracy decreased. Currently, the use of higher order interpolators is not common so this detection is not an issue.

The proposed ADZ algorithm performed much better in detecting tampered regions than those in previous works, as it was able to detect scales of 2.0 and rotated tampered regions. The algorithm in [9] worked better than that in [11] since it was able to detect scales of 2.0, which [11] could not. These algorithms could not detect tampered regions that were rotated. The algorithm in [15] could detect rotated tampered regions, but not scales of 2.0. The color space used for the input image also played a role in the detection, as noted in [9].

The peak threshold and block size had an impact on the results. For undetected blocks, with or without rotated regions, peaks may have existed but the magnitudes fell below the threshold value. The block size determines the amount of data provided for peak detection, apart from restricting the range of values searched for peaks. The blocks were required to be large enough to detect the periodic patterns but not so large that the tampered region occupied less than half the area, leaving insufficient data to detect the patterns. Using a fixed block size would limit the applicability. Overlapping blocks were used to increase the accuracy of detection by shifting the blocks to cover more of the tampered regions however the block size may still be inappropriate to detect the periodicity. Furthermore, the entire image may not be searched for tampering if there were not enough columns or rows lefts to make a block. Increasing the overlap by having a one pixel shift would solve this issue at a cost of computational time. Therefore, improvements include development of optimal peak detection and adaptable block size algorithms. The adaptable block size algorithm can also ease evaluation where counting blocks is concerned.

In the investigation of the detection methods, it was seen that most algorithms incorporated estimation of the resampling factor. The evaluation of these techniques would tend towards the accuracy of the detected resampling factor after a region of interest was selected or discovered. In the algorithm proposed in this paper, estimation was not a concern. The evaluation focused more on the coverage of tampered regions. That is, how many blocks were detected? How many fell on tampered regions? How many did not fall on tampered regions? How many more should have been detected to cover the tampered regions? This meant that the creator of the tampered images for testing would be required to take the measurements since other persons would be unable to identify the tampered region, unless ground truth data was provided.

Counting blocks was a task done manually and therefore human error existed. The low precision and recall values recorded did not mean that the algorithm failed to detect tampered images, but that it did not detect every area of the tampered region. In fact, the specificity value showed that the detection algorithm had a near perfect rate of determining that authentic images were not interpolated. Implementing estimation and evaluating the detection accuracy for various resampling factors would give more comparable statistics to that in the literature. Doing so will

determine the range of resampling factors within the algorithm's limits. Furthermore, a proper investigation of different image formats and input image color spaces can be done as in [9]. Testing can be done on more complex forgeries made with post-processing like blurred edges or regions.

7. CONCLUSION

Digital forgeries often require an image to be scaled and/or rotated. The transformations involve re-sampling, which employs interpolation. Interpolating functions conceal the areas of tampering by varying degrees. While these changes may not be seen with the naked eye, interpolation introduces correlations in the underlying statistics of the image. This paper compared interpolation techniques used in digital image editing software and presented a novel algorithm (ADZ) for its automatic detection.

The comparison of the interpolation techniques followed the methods outlined in [1]. The interpolating functions considered were nearest neighbor, bilinear, bicubic and lanczos3 interpolation. The investigations looked at the Fourier transformations of the interpolating functions, the qualitative and quantitative image quality produced by the different functions, and runtime measurements. The results were comparable to that in the literature. Lanczos3 interpolation was the closest to ideal interpolation. Nearest neighbor performed the worst, producing pixelated images. Bilinear interpolation produced a smoother image with artifacts, while bicubic interpolation produced a sharper image with fewer artifacts. Computational time increased as the complexity of the interpolator increased.

This paper proposed the ADZ algorithm that was able to identify local tampered regions by dividing the image into blocks and searching for periodicity in the zero-crossings of the second derivative of the resampled signal. This method was chosen over the Expectation-Maximization method as it was faster and based on a simpler principle. The algorithm was tested on upsampled images only, as suggested in [22], on authentic and tampered images, where the tampered regions were scaled, rotated or scaled and rotated using bilinear and bicubic interpolation. The performance was measured using the specificity metric for authentic images, and the precision and recall metrics for tampered images. The algorithm worked well to determine that authentic images were not resampled. However, the accuracy of detecting tampered regions was satisfactory for those that were scaled only, and scaled and rotated, but failed for rotation only regions. It was found that bilinear interpolation was detected better than bicubic. The algorithm performed better than those in previous works.

Several improvements can be made in the detection of interpolation as discussed in section 6. Therefore, future work includes:

- Estimation of resampling factors
- Investigation and application of techniques to improve rotation detection
- Investigation and application of optimal peak detection strategies
- Development of an adaptable block size algorithm
- Experiments using JPEG compressed Tampered images
- Experiments using different input image color spaces

This work shows that there may be a great impact on the detection of locally tampered images using resampling methods. It also outlines performance metrics-driven methodology which can be used in assessing any future work in tampered image detection, in order to have a similar

benchmark across the varying techniques. The Interpolation techniques are always evolving, so research into detection methods must continue [23].

8. ACKNOWLEDGMENT

The authors would like to thank the Department of Electrical and Computer Engineering at The University of the West Indies for supporting this research.

9. REFERENCES

[1] T. M. Lehmann, C. Gonner, and K. Spitzer, “Survey: Interpolation methods in medical image processing,” *Medical Imaging, IEEE Transactions on*, vol. 18, no. 11, pp. 1049-1075, 1999.

[2] 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.

[3] H. Farid, “Image forgery detection," *Signal Processing Magazine, IEEE*, vol. 26, no. 2, pp. 16-25, March 2009.

[4] T. Qazi, K. Hayat, S. U. Khan, S. A. Madani, I. A. Khan, J. Ko lodziej, H. Li, W. Lin, K. C. Yow, and C.-z. Xu, “Survey on blind image forgery detection," *Image Processing, IET,* vol. 7, no. 7, pp. 660-670, 2013.

[5] J. A. Parker, R. V. Kenyon, and D. Troxel, “Comparison of interpolating methods for image resampling*," Medical Imaging, IEEE Transactions on,* vol. 2, no. 1, pp. 31-39, 1983.

[6] T. Acharya and P.-S. Tsai, “Computational foundations of image interpolation algorithms," *ACM Ubiquity,* vol. 8, no. 42, 2007.

[7] P. Thevenaz, T. Blu, and M. Unser, “Image interpolation and resampling," *Handbook of medical imaging, processing and analysis*, pp. 393-420, 2000.

[8] K. Turkowski, “Filters for common resampling tasks," in Graphics Gems. Academic Press Professional, Inc., 1990, pp. 147-165.

[9] G. K. Birajdar and V. H. Mankar, “Blind authentication of resampled images and rescaling factor estimation," in *Cloud & Ubiquitous Computing & Emerging Technologies (CUBE), 2013 International Conference on. IEEE*, 2013, pp. 112-116.

[10] T. M. Lehmann, C. Gonner, and K. Spitzer, “Addendum: B-spline interpolation in medical image processing," *Medical Imaging, IEEE Transactions on*, vol. 20, no. 7, pp. 660-665, 2001.

[11] A. C. Gallagher, “Detection of linear and cubic interpolation in jpeg compressed images," in *Computer and Robot Vision, 2005. Proceedings. The 2nd Canadian Conference on*. IEEE, 2005, pp. 65-72.

[12] D. Zhu and Z. Zhou, “Resampling tamper detection based on jpeg double compression," in *Communication Systems and Network Technologies (CSNT), 2014 Fourth International Conference on*. IEEE, 2014, pp. 914-918.

[13] S. Prasad and K. Ramakrishnan, “On resampling detection and its application to detect image tampering," in *Multimedia and Expo, 2006 IEEE International Conference on.* IEEE, 2006, pp. 1325-1328.

[14] B. Mahdian and S. Saic, “Blind authentication using periodic properties of interpolation," *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 3, pp. 529-538, 2008.

[15] W. Wei, S. Wang, X. Zhang, and Z. Tang, “Estimation of image rotation angle using interpolation-related spectral signatures with application to blind detection of image forgery," *Information Forensics and Security, IEEE Transactions on,* vol. 5, no. 3, pp. 507-517, 2010.

[16] T. Lehmann, A. Sovakar, W. Schmiti, and R. Repges, “A comparison of similarity measures for digital subtraction radiography," *Computers in Biology and Medicine,* vol. 27, no. 2, pp. 151-167, 1997.

[17] G. Schaefer and M. Stich, “Ucid: an uncompressed color image database," in *Electronic Imaging 2004.* International Society for Optics and Photonics, 2003, pp. 472-480.

[18] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks*," Information Processing & Management*, vol. 45, no. 4, pp. 427-437, 2009.

[19] E. Maeland, “On the Comparison of Interpolation Methods," *Medical Imaging, IEEE Transactions on,* vol. 7, no. 3, pp. 213-217, 1988.

[20] R. Qian, W. Li, N. Yu, and Z. Hao, “Image forensics with rotation-tolerant resampling detection," in *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*. IEEE, 2012, pp. 61-66.

[21] B. Mahdian and S. Saic, “Detection of Resampling Supplemented with Noise Inconsistencies Analysis for Image Forensics," in *Computational Sciences and Its Applications (ICCSA 2008), International Conference on*. IEEE, 2008, pp. 546-556.

[22] Y.T. Kao, H.J. Lin, C.W. Wang and Y.C. Pai, “Effective Detection for Linear Up-Sampling by a Factor of Fraction," *Image Processing, IEEE Transactions on,* vol. 21, no. 8, pp. 3443-3453, 2012.

[23] S. Math and R.C. Tripathi, “Digital Forgeries: Problems and Challenges," *International Journal of Computer Applications,* vol. 5, no. 12, pp. 9-12, 2010.