

Approximate Detection Method for Image Up-Sampling

Ching-Ting Tu¹, Hwei-Jen Lin¹, Fu-Wen Yang¹, Hsiao-Wei Chang²

¹Department of Computer Science and Information Engineering
Tamkang University
Taipei, Taiwan, 25137 ROC
[e-mail: 086204@mail.tku.edu.tw]

²Department of Computer Science and Information Engineering
China University of Science of Technology
Taipei, Taiwan, 11581 ROC
[e-mail: changhw@cc.cust.edu.tw]

Received July 4, 2013; revised December 30, 2013; accepted January 23, 2014; published February 28, 2014

Abstract

This paper proposes a new resampling detection method for images that detects whether an image has been resampled and recovers the corresponding resampling rate. The proposed method uses a given set of zeroing masks for various resampling factors to evaluate the convolution values of the input image with the zeroing masks. Improving upon our previous work, the proposed method detects more resampling factors by checking for some periodicity with an approximate detection mechanism. The experimental results demonstrate that the proposed method is effective and efficient.

Keywords: image forgery, copy-move, resampling rate, resampling factor, zeroing mask

1. Introduction

Rapid development of techniques over the past decades have enabled people to easily acquire images and share or transfer them through the Internet. These images are easily modified or further synthesized by existing image-processing software for certain purposes. Such image manipulation might violate the copyright or illegitimately cause a bad result. To combat these illegitimate practices, we urgently need algorithms that can detect such forgeries.

In recent years, many forgery-detection methods have been proposed. Popescu et al. [1] and Kirchner [2] detected forgeries with resampling or rotation by checking for some periodicity in the images. Nillius et al. [3] and Johnson et al. [4] detected digital forgeries by detecting the lighting consistencies. Li et al. [5] detected tampered watermarked images with the embedded information and recovered the images using methods that are similar to those in [6, 7]. Camera defects such as chromatic aberration [8] and sensor pattern noise [9-12], as well as the color filter arrays [1] that cameras use to interpolate colors, can be used to detect forgeries. Copy-move images are easily made by copying certain regions and pasting them on some other regions. Some methods have been proposed to detect these types of forged images [13-18].

Resampling is frequently involved in image forgeries, which motivates us to study the detection of such image manipulation. Most existing resampling-detection methods [1, 2] can detect resampling but cannot recover the resampling factors (or resampling rates). The method proposed by Lien et al. [19] addresses the recovery of the resampling rates, but it can only recover the resampling factors with the provided corresponding weighting tables and must use weighting tables to recover the possible original image and examine the periodicity. Furthermore, their work is notably time-consuming, and false positive may be caused when the set of weighting tables is not sorted in specific orders.

We have previously proposed an algorithm [20] to detect a wider range of resampling factors in less time by providing a set of zeroing masks. However, the detectable resampling factor must correspond to a zeroing mask in the provided set. In other words, the amount of detectable resampling factors is exactly identical to the number of zeroing masks that are provided for detection. The objective of this paper is to overcome this limitation of our previous work and detect many more factors than the number of provided zeroing masks. If an image is resampled by a factor that corresponds to a provided zeroing mask, then the exact resampling factor can be detected; otherwise, the image will be further detected using an approximate detection mechanism. Therefore, the proposed method is more applicable.

The remainder of this paper is organized as follows. The basic concept of resampling and a review of our previous work are introduced in Section 2. In Section 3, the proposed method of approximate resampling detection is described. Section 4 presents some experimental results, and Section 5 concludes.

2. Overview of Exact Resampling Detection

In this section, we will briefly introduce the basic concept of resampling and the exact resampling detection that was proposed in our previous work [20].

2.1 Resampling

Resampling is the mathematical technique to create a new version of the image with a different width and/or height in pixels. Increasing the image size is called upsampling; reducing its size is called downsampling.

Resampling of a 2-D image can be broken down into two 1-D resampling passes. In one pass, horizontal resampling is performed by producing an image with a different width but the same height. In the next pass, this intermediate image is resampled vertically to change its height while maintaining the width. This process is much more computationally efficient than combining the work into one pass. Upsampling involves interpolating among the existing pixels to estimate their values at the new pixel locations. Downsampling involves computing the weighted average of the original pixels that overlap each new pixel. Color images are treated like three grey-scaled images which are separated from the original image, individually resampled, and finally recombined to create the final image.

For 2-D resampling, each dimension can be separately resampled and detected. To easily explain, we will first describe one-dimensional resampling [1].

Let $x[t]$ be a 1-D discrete signal with n samples to be sampled. The number of samples in signal $x[t]$ can be resampled using a factor of p/q to create a new sample $y[t]$ with m samples as follows:

1. *Up-sample*: Create a new signal $x_u[pt] = x[t]$, where $t = 1, 2, \dots, n$, and $x_u[pt] = 0$ otherwise.
2. *Interpolate*: Convolve $x_u[t]$ with a low-pass filter $h[t]$: $x_i[t] = x_u[t] * h[t]$.
3. *Down-sample*: Create a new signal $x_d[t]$ with m samples, where $x_d[t] = x_i[qt]$, and $t = 1, 2, \dots, m$. Denote the resampled signal as $y[t] = x_d[t]$.

Different types of resampling algorithms (e.g., linear cubic) differ in the form of the interpolation filter in Step 2. In this study, we assume that the linear interpolation filter is used. Because all three steps in the signal resampling are linear, this process can be described with a linear transformation as shown in (1).

$$y = A_{p/q} x, \quad (1)$$

where $A_{p/q}$ is the transformation matrix of size $m \times n$. Depending on the resampling fraction, the resampling process introduces correlations of varying degrees among the neighboring samples. For example, consider the upsampling of a signal by a factor of two. In this case, the resampling matrix takes the following form:

$$A_{2/1} = \begin{bmatrix} 1 & 0 & 0 & & \\ 0.5 & 0.5 & 0 & & \\ 0 & 1 & 0 & & \\ 0 & 0.5 & 0.5 & & \\ 0 & 0 & 1 & & \\ & & & \ddots & \ddots \end{bmatrix} \quad (2)$$

Here, the odd samples of the resampled signal $y = [y_1 y_2 \dots y_m]^t$ take on the values of the original signal $x = [x_1 x_2 \dots x_n]^t$; i.e., $y_{2i-1} = x_i$, $i = 1, 2, \dots, n$, where $y_j = y[j]$ and $x_j = x[j]$. The even samples are the averages of the adjacent neighbors of the original signal:

$$y_{2i} = 0.5 \cdot x_i + 0.5 \cdot x_{i+1}, \quad (3)$$

where $i = 1, 2, \dots, n-1$. Because $x_i = y_{2i-1}$ and $x_{i+1} = y_{2i+1}$, the above relationship can be expressed in terms of only the resampled samples:

$$y_{2i} = 0.5 \cdot y_{2i-1} + 0.5 \cdot y_{2i+1}. \quad (4)$$

In other words, across the entire resampled signal, each even sample is precisely the same linear combination of its adjacent neighbors. According to this description, we will use the relation among the pixels and their neighbors to detect resampling.

2.2 Constructing Resampling Matrices

In this subsection, we describe how our previous work [20] constructed the resampling matrices. For a matrix A , let $A[r_1..r_2, c_1..c_2]$ denote the sub-matrix of A that is formed with the entries between rows r_1 and r_2 , and between columns c_1 and c_2 . It can be observed that the matrix $A_{p/q}$ is periodically composed of its sub-matrix $A_{p/q}[1..p, 1..q+1]$. For example, the matrix $A_{4/3}$ is periodically composed of the sub-matrix $A_{4/3}[1..4, 1..4]$. This result can be described as follows.

$$A_{4/3}[1+4k..4+4k, 1+3k..4+3k] = A_{4/3}[1..4, 1..4], \quad (5)$$

for every integer k with $0 < 4k < m$ and $0 < 3k < n$, or $0 < k < \min\{m/4, n/3\}$. To be more general, we have

$$A_{p/q}[1+pk..p+pk, 1+qk..q+1+qk] = A_{p/q}[1..p, 1..q+1], \quad (6)$$

for every integer k with $0 < pk < m$ and $0 < qk < n$, or $0 < k < \min\{m/p, n/q\}$.

Because of the periodicity, it is sufficient to consider only the $(p+1) \times (q+1)$ sub-matrix $A_{p/q}[1..p+1, 1..q+1]$ for any resampling matrix $A_{p/q}$. We called the sub-matrix $A_{p/q}[1..p+1, 1..q+1]$ the primary sub-matrix of $A_{p/q}$. For simplicity, when we refer to a resampling matrix $A_{p/q}$, we mean its primary sub-matrix. (7) and (8) show the newly defined matrices $A_{2/1}$ and $A_{5/3}$, respectively, from each of which we can see that the entries are symmetric to the center of the matrix; in other words, $A_{p/q}[i, j] = A_{p/q}[p+2-i, q+2-j]$ for $i = 1, 2, \dots, p+1$ and $j = 1, 2, \dots, q+1$.

$$A_{2/1} = \begin{bmatrix} 1 & 0 \\ 0.5 & 0.5 \\ 0 & 1 \end{bmatrix} \quad (7)$$

$$A_{5/3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.4 & 0.6 & 0 & 0 \\ 0 & 0.8 & 0.2 & 0 \\ 0 & 0.2 & 0.8 & 0 \\ 0 & 0 & 0.6 & 0.4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Again, for simplicity, we assume that $q = 1$ or p and q are coprime (or relatively prime) positive integers. For $1 \leq q < p \leq 12$, there are 45 factors p/q , which are composed as listed below:

- $p = 2, q = 1;$
- $p = 3, q = 1, 2;$
- $p = 4, q = 1, 3;$
- $p = 5, q = 1 \sim 4;$
- $p = 6, q = 1, 5;$
- $p = 7, q = 1 \sim 6;$
- $p = 8, q = 1, 3, 5, 7;$
- $p = 9, q = 1, 2, 4, 5, 7, 8;$
- $p = 10, q = 1, 3, 7, 9;$
- $p = 11, q = 1 \sim 10;$
- $p = 12, q = 1, 5, 7, 11.$

It can be further observed from (7) and (8) that every row except the top row and the bottom row in a resampling matrix $A_{p/q}$ has exactly two nonzero entries, which are adjacent in the row. The nonzero entries in row i are $((p - (i + 1) \cdot q) \bmod p) / p$ and $((i + 1) \cdot q \bmod p) / p$. Assume that the nonzero entries in row i ($2 \leq i \leq p - 1$) are located in the j th and $(j + 1)$ th columns. Then, the nonzero entries in row $i + 1$ are in the $(j + 1)$ th and $(j + 2)$ th columns if $(p - (i + 1) \cdot q) \bmod p < (p - (i + 2) \cdot q) \bmod p$; and in the j th and $(j + 1)$ th columns otherwise. The general form of the resampling matrix $A_{p/q}$ is given in (9), where $\beta_i = ((p - i \cdot q) \bmod p) / p, i = 1, 2, \dots, p - 2$, denotes the first nonzero entry in the $(i + 1)$ th row of $A_{p/q}$. Let β_r be in column $c(r)$ of $A_{p/q}$ then $A_{p/q}[(r + 1), c(r)] = \beta_r$. If $\beta_r > \beta_{r+1}$ (or $(p - r \cdot q) \bmod p > (p - (r + 1) \cdot q) \bmod p$), then β_r and β_{r+1} must be in the same column, i.e., $c(r + 1) = c(r)$; otherwise, $c(r + 1) = c(r) + 1$.

$$A_{p/q} = \begin{bmatrix} 1 & 0 & 0 & 0 & & \dots & \dots & \dots & 0 & 0 \\ \beta_1 & 1 - \beta_1 & 0 & 0 & & \dots & \dots & \dots & & 0 \\ 0 & \ddots & \ddots & 0 & & \dots & \dots & \dots & & 0 \\ 0 & 0 & \beta_r & 1 - \beta_r & 0 & \dots & \dots & \dots & & \vdots \\ \vdots & \dots & 0 & \beta_{r+1} & 1 - \beta_{r+1} & 0 & \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & \beta_{r+2} & 1 - \beta_{r+2} & 0 & 0 & & & 0 \\ 0 & \dots & 0 & \vdots & \vdots & 0 & 0 & \dots & \dots & 0 \\ 0 & \dots & 0 & \beta_{r+k} & 1 - \beta_{r+k} & 0 & & \dots & \dots & \vdots \\ \vdots & \dots & & 0 & \beta_{r+k+1} & 1 - \beta_{r+k+1} & 0 & \dots & \dots & 0 \\ 0 & \vdots & & & \ddots & \ddots & & \dots & \dots & 0 \\ 0 & & & & & & & & 0 & \beta_{p-2} & 1 - \beta_{p-2} \\ 0 & 0 & \dots & & \dots & \dots & & & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

As shown in (9), assume that $\beta_i > \beta_{i+1}$ (or $(p - r \cdot q) \bmod p > (p - (r + 1) \cdot q) \bmod p$), for $i = r+1, r+2, \dots, r+k$, and $\beta_{r+k} < \beta_{r+k+1}$ (or $(p - (r+k) \cdot q) \bmod p < (p - (r+k+1) \cdot q) \bmod p$), then $c(r) + 1 = c(r+1) = c(r+2) = \dots = c(r+k) = c(r+k+1) - 1$. For each row between the 3rd row and the $(p-1)$ th row where its first nonzero entry is located at the same column as the first nonzero entry in the preceding row, we called it the *following row*. It can be observed that the matrix $A_{p/q}$ possesses a *following row* only when $p > q+1$. For instance, the 4th row of the matrix $A_{5/3}$ is a *following row*.

The sequence of the entry pairs $((\beta_1, 1 - \beta_1), (\beta_2, 1 - \beta_2), \dots, (\beta_{p-2}, 1 - \beta_{p-2}))$ is symmetric, i.e., $\beta_i = 1 - \beta_{p-1-i}$ for $i = 1, 2, \dots, \lfloor (p-1)/2 \rfloor$.

With the above observation, we may determine the content of any resampling matrix $A_{p/q}$. The algorithm *Resampling Matrix Construction (RMC)* [20] that derives the resampling matrices is given as follows.

Algorithm RMC

Input : two positive coprime integers p and q with $p > q$

Output : Matrix $A[1..1+p, 1..q+1]$

for $i \leftarrow 1$ to $p+1$

 for $j \leftarrow 1$ to $q+1$

$A[i, j] \leftarrow 0$;

$A[1, 1] \leftarrow 1; A[p+1, q+1] \leftarrow 1; j \leftarrow 1$;

for $i \leftarrow 2$ to $\lfloor \frac{p+1}{2} \rfloor$

 if $(p-iq) \bmod p < (p-(i+1)q) \bmod p$

 then $j \leftarrow j+1$;

$A[i, j] \leftarrow \frac{(p-(i+1)q) \bmod p}{p}$;

$A[i, j+1] \leftarrow 1 - A[i, j]$;

$A[p+2-i, q+2-j] \leftarrow A[i, j]$;

$A[p+2-i, q+1-j] \leftarrow A[i, j+1]$

2.3 Deriving Zeroing Masks

With the above assumption for resampling matrices, we consider only the first $q+1$ samples of the signal x to be resampled, say $x[t]$, $t = 1, 2, \dots, q+1$, and the first $p+1$ samples of the resampled signal y , say $y[t]$, $t = 1, 2, \dots, p+1$. As a result, we write the relationship in (4) (with $i = 1$) for the signal that is resampled by a fraction of $2/1$ to:

$$-0.5 \cdot y_1 + y_2 - 0.5 \cdot y_3 = 0 \quad (10)$$

which indicates that the convolution of signal y with the 3×1 filter $F_{2/1} = [-0.5 \ 1 \ -0.5]$ zeroes the samples of y . We call this filter a zeroing mask for the resampling factor $2/1$. Of course,

any resampled signal can be zeroed using a zero filter (a zero vector), which cannot be used to detect resampling. From now on, we are only interested in nonzero zeroing filters.

We have proved in our work that there is a nonzero zeroing mask for any resample factor p/q , where p and q are coprime positive integers and $p > q$, or that there is a $(p+1) \times 1$ zeroing filter $F_{p/q} = [\alpha_1 \alpha_2 \dots \alpha_{p+1}]$, where α_i 's are not all zeros, such that for any resampled signal y by factor p/q , the following relation holds.

$$\sum_{t=1}^{p+1} \alpha_t \cdot y_t = 0, \quad (11)$$

which is equivalent to the convolution of y with $F_{p/q}$:

$$F_{p/q} * y = 0, \quad (12)$$

where $F_{p/q} * y = F_{p/q} y$ if y is of size $(p+1) \times 1$ and $F_{p/q} y^t$ if y is of size $1 \times (p+1)$. Assuming that y is of size $(p+1) \times 1$, substituting (1) in (12) yields:

$$F_{p/q} A_{p/q} x = 0, \quad (13)$$

Because the relation in (13) holds for every vector x , $A_{p/q}^t F_{p/q}^t$ must be a zero vector, as shown in (14), which can be regarded as a linear system of $q + 1$ equations with $p + 1$ unknowns.

$$A_{p/q}^t F_{p/q}^t = \mathbf{0} \quad (14)$$

As an example, considering the resampling by a fraction of 5/3; then, (11) becomes

$$\sum_{t=1}^6 \alpha_t \cdot y_t = 0. \quad (15)$$

Substituting $y = A_{5/3}x$ in (15) yields

$$\begin{aligned} & \alpha_1 \cdot x_1 + \alpha_2 \cdot (0.4 \cdot x_1 + 0.6 \cdot x_2) + \alpha_3 \cdot (0.8 \cdot x_2 + 0.2 \cdot x_3) + \\ & \alpha_4 \cdot (0.2 \cdot x_2 + 0.8 \cdot x_3) + \alpha_5 \cdot (0.6 \cdot x_3 + 0.4 \cdot x_4) + \alpha_6 \cdot x_4 = 0, \end{aligned} \quad (16)$$

which is equivalent to

$$\begin{aligned} & (\alpha_1 + 0.4 \cdot \alpha_2) \cdot x_1 + (0.6 \cdot \alpha_2 + 0.8 \cdot \alpha_3 + 0.2 \cdot \alpha_4) \cdot x_2 + \\ & (0.2 \cdot \alpha_3 + 0.8 \cdot \alpha_4 + 0.6 \cdot \alpha_5) \cdot x_3 + (0.4 \cdot \alpha_5 + \alpha_6) \cdot x_4 = 0. \end{aligned} \quad (17)$$

Because (17) holds for any signal x , the coefficient for each x_i in (17) must be zero, i.e.,

$$\begin{cases} \alpha_1 + 0.4 \cdot \alpha_2 = 0 \\ 0.6 \cdot \alpha_2 + 0.8 \cdot \alpha_3 + 0.2 \cdot \alpha_4 = 0 \\ 0.2 \cdot \alpha_3 + 0.8 \cdot \alpha_4 + 0.6 \cdot \alpha_5 = 0 \\ 0.4 \cdot \alpha_5 + \alpha_6 = 0 \end{cases} \quad (18)$$

(18) forms a system of 4 linear equations with 6 unknowns to solve the zeroing filter $F_{p/q}$ and is equivalent to the relation $A_{5/3}' F_{5/3}' = \mathbf{0}$ or the matrix equation:

$$\begin{bmatrix} 1 & 0.4 & 0 & 0 & 0 & 0 \\ 0 & 0.6 & 0.8 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0.8 & 0.6 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (19)$$

Thus, the matrix equation that corresponds to the fraction p/q , as shown in (14), represents a system of $q+1$ linear equations with $p+1$ unknowns.

Two masks $M = [m_1 m_2 \dots m_K]$ and $M' = [m_1' m_2' \dots m_K']$ are considered equivalent if $M = cM'$ (or $m_i/m_i' = c$ for $i = 1, 2, \dots, K$) for some constant c . Two equivalent masks zero the same set of signals. Hence, a solution for a zeroing mask corresponds to a set of parallel vectors (M and M' are considered two parallel vectors), which can be considered a 1-D vector space. Therefore, for $p \geq q+2$ (or $p-q \geq 2$), there is an infinite amount of nonequivalent zeroing masks for the resample fraction p/q . To unify equivalent zeroing masks and exclude the zero zeroing mask, we fix the first element $\alpha_1 = 1$ for every mask $F_{p/q} = [\alpha_1 \alpha_2 \dots \alpha_{p+1}]$. This action is equivalent to adding the equation $\alpha_1 = 1$ in each of the aforementioned systems. This setting can also avoid trivial solutions or the solution of the zero filter. For example, adding the equation $\alpha_1 = 1$ into (18) forms a nonhomogeneous linear system, as shown in (20), which is equivalent to the matrix form in (21).

$$\begin{cases} \alpha_1 = 1 \\ \alpha_1 + 0.4 \cdot \alpha_2 = 0 \\ 0.6 \cdot \alpha_2 + 0.8 \cdot \alpha_3 + 0.2 \cdot \alpha_4 = 0 \\ 0.2 \cdot \alpha_3 + 0.8 \cdot \alpha_4 + 0.6 \cdot \alpha_5 = 0 \\ 0.4 \cdot \alpha_5 + \alpha_6 = 0 \end{cases} \quad (20)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0.4 & 0 & 0 & 0 & 0 \\ 0 & 0.6 & 0.8 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0.8 & 0.6 & 0 \\ 0 & 0 & 0 & 0 & 0.4 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (21)$$

The zeroing masks for the resample factor 5/3 can be obtained by solving the nonhomogeneous linear system in (20) or (21). In general, the zeroing mask $F_{p/q}$ for the resample factor p/q can be obtained by solving the following nonhomogeneous linear system.

$$E_{p/q} F_{p/q}^t = I_{q+2}, \quad (22)$$

where $E_{p/q} = \begin{bmatrix} I_{p+1}^t \\ A_{p/q}^t \end{bmatrix}$, $I_k = [1\ 0\ 0 \dots 0]_k^t$ denotes a unit column matrix with a size of $k \times 1$, and $F_{p/q} = [\alpha_1\ \alpha_2 \dots \alpha_{p+1}]$ is a row matrix of size $1 \times (p+1)$ that consists of $p+1$ variables. Therefore, $E_{p/q}$ is a matrix of size $(q+2) \times (p+1)$ with the submatrix I_{p+1}^t at its top row, i.e., $E_{p/q} [1..1, 1..p+1] = I_{p+1}^t$, and $E_{p/q} [2..q+1, 1..p+1] = A_{p/q}^t$. The general form of $A_{p/q}$, which is given in (9), yields the expanded form of $E_{p/q}$.

2.4 Detecting Resampling with Zeroing Masks

Image resampling by any factor can be detected by evaluating the convolution of the image with a sequence of zeroing masks. As follows, the algorithm *ZMD* (*zeroing mask derivation*) [20] generates a zeroing mask that corresponds to any resampling factor p/q and the algorithm *UD* (*up-sampling detection*) [20] detects resampling with a set of provided zeroing masks. To detect resampling by factor p/q on an image region I , we use a resampling score function rs , as provided in (23), to evaluate the confidence value of resampling, where $B(I)$ is the set of divided blocks of the image region I , and $*$ denotes the convolution operator. Block B_{ij} is considered resampled by factor p/q if the convolution value $M_{p/q} * B_{ij}$ is sufficiently small. The number of resampled blocks is counted using the function h : $h(x) = 1$ if $x \leq s$, and 0 otherwise, where s is a given threshold. If the resampling score value $rs(I, p/q)$ is greater than a threshold t , we consider the image region I as having been resampled by fraction p/q . The score value rs falls between 0 and 100.

$$rs(I, p/q) = \frac{100}{|B(I)|} \sum_{B_{ij} \in B(I)} h(M_{p/q} * B_{ij}) \quad (23)$$

Algorithm ZMD

Input two positive coprime integers p and q with $p > q$

Output coefficients a_1, a_2, \dots, a_{p+1}

$a_1 \leftarrow 1$

$\Delta \leftarrow 1 - \beta_1$

$i \leftarrow 2$

while $i \leq \lfloor (p+2)/2 \rfloor$ **do**

$k \leftarrow 1$

$\alpha \leftarrow \beta_i$

while $\beta_{i+k-1} \geq \beta_{i+k}$ **do**

$\alpha \leftarrow \alpha + \beta_{i+k}$

$k \leftarrow k + 1$

$a_i \leftarrow (-\Delta) / \alpha$

$\ell \leftarrow 1$

while $\ell \leq k - 1$ **do**

$a_{i+\ell} \leftarrow a_{i+\ell-1}$

$\ell \leftarrow \ell + 1$

$\Delta \leftarrow a_i \cdot (k - \alpha)$

$i \leftarrow i + k$

if p is odd & $i - 1 = \lfloor (p+2)/2 \rfloor$

then

while $i \leq p + 1$ **do**

$a_i \leftarrow -a_{p+2-i}$

$i \leftarrow i - 1$

else

while $i \leq p + 1$ **do**

$a_i \leftarrow a_{p+2-i}$

$i \leftarrow i - 1$

Algorithm UD

$k \leftarrow 1$; *resampled* \leftarrow *false*;

while ($k \leq K$) **and** (**not** *resampled*) **do**

if $rs(I, p[k]/q[k]) > t_2$

then *resampled* \leftarrow *true*

else $k \leftarrow k + 1$;

if *resampled*

then return $p[k]$ & $q[k]$

else return "not resampled"

3. Recovery of Approximate Resampling Rates

Our previous method can detect only the resampling factors that corresponds to the provided zeroing masks. To detect more resampling factors, one must provide more zeroing masks; thus, the required detection time is proportional to the number of provided zeroing masks. As a result, the detectable resampling factors are limited. In this section, we present an approximate detection method that can detect many more resampling factors than the previous method using the same set of zeroing masks.

Assume that $y = (y_1, y_2, y_3, \dots)$ is a resampled signal of the origin signal $x = (x_1, x_2, x_3, \dots)$ with the resampling factor p/q ; thus, $y = A_{p/q}x$. Let $y[\alpha..\beta]$ denotes the sub-signal $(y_\alpha, y_{\alpha+1}, y_{\alpha+2}, \dots, y_\beta)$ of y . Then, by (23), for any nonzero integer k , the convolution value $\text{Conv}(M_{p/q}, y, 1+kp)$, which is defined in (24), of a sub-signal $y[1+kp..1+(k+1)p]$ with the zeroing mask $M_{p/q}$ must be zero.

$$\text{Conv}(M_{p/q}, y, i) = \frac{1}{\|M_{p/q}\|} \left| \sum_{j=1}^{p+1} m_j y_{i-1+j} \right| \quad (24)$$

Table 1(a) shows an original signal x , and **Table 1(b)** shows a resampled signal y of x with factor $4/3$. The convolution value $\text{Conv}(M_{4/3}, y, 1+4k)$ of the sub-signals $y[1+4k .. 5+4k]$ (or $y[1..5], y[5..9], y[9..13], \dots$) with the zeroing mask $M_{4/3}$ shown in **Table 1(c)** must be zero. In this example, $\text{Conv}(M_{4/3}, y, 1) = \sum_{j=1}^5 m_j y_j = 0$, but $\text{Conv}(M_{4/3}, y, 2) = \sum_{j=1}^5 m_j y_{j+1} = 5.50 \neq 0$. **Table 1(d)** provides more values for more sub-sequences.

Table 1. (a) signal x ; (b) signal y ; (c) zeroing mask $M_{4/3}$; (d) convolution values of y with $M_{4/3}$

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
20	48	64	40	72	40	64	72	100	96	76
x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	...	
116	140	120	120	100	116	80	96	100	...	

(a)

y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}
20	41	56	58	40	64	56	46	64	70	86	99	96
y_{14}	y_{15}	y_{16}	y_{17}	y_{18}	y_{19}	y_{20}	y_{21}	y_{22}	y_{23}	y_{24}	y_{25}	y_{26}
81	96	122	140	125	120	115	100	112	98	84	96	...

(b)

m_1	m_2	m_3	m_4	m_5
0.25	-1	1.5	-1	0.25

(c)

i	1	2	3	4	5	6	7
$\text{Conv}(M_{4/3}, y, i)$	0	8.25	16.26	12.43	0	8.37	7.41
i	8	9	10	11	12	13	14
$\text{Conv}(M_{4/3}, y, i)$	4.18	0	2.03	4.54	7.29	0	0.72
i	15	16	17	18	19	20	21
$\text{Conv}(M_{4/3}, y, i)$	8.13	6.33	0	5.62	10.76	9.44	0

(d)

Table 1(d) shows that $\text{Conv}(M_{4/3}, y, i) = 0$ for only $i = 1 + 4k$, i.e., the convolution value of zero appears periodically. Therefore, the previously proposed method can detect an image if it is resampled by a factor that corresponds to a provided zeroing mask for detection, and simultaneously, it can correctly detect the resampling factor. The recovery rate approaches 99%. In our experiment, we follow our previous work [20] with the considered factors p/q , where $1 \leq q < p \leq 12$, such that $q = 1$ or p and q are coprime (or relatively prime) positive integers. Thus, there are 45 zeroing masks for the 45 provided factors for detection. As previously mentioned, this method is limited. It can only detect a provided fixed set of resampling factors. To overcome this limitation, we propose an approximate detection mechanism to detect many more resampling factors than those whose zeroing masks are provided. First, we will describe how to use the 45 zeroing masks to detect a wider range of resampling factors.

For example, the signal z shown in **Table 2(a)** is a resampled signal of the original signal x in **Table 1(a)** with the resampling factor 13/10, whose resampling rate can be successfully detected if the zeroing mask $M_{13/10}$ is provided. However, the detection will fail because the zeroing mask $M_{13/10}$ is not included in the set of zeroing masks.

Table 2. (a) signal z ; (b) convolution values of z with $M_{4/3}$

z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_9	z_{10}	z_{11}	z_{12}
20	42	57	57	42	67	52	49	65	71	91	98
z_{13}	z_{14}	z_{15}	z_{16}	z_{17}	z_{18}	z_{19}	z_{20}	z_{21}	z_{22}	z_{23}	...
91	76	107	129	134	120	120	108	106	110	83	...

(a)

i	1	2	3	4	5	6	7
$\text{Conv}(M_{4/3}, z, i)$	0.96	6.57	16.14	15.78	5.38	4.3	6.33
i	8	9	10	14	11	12	13
$\text{Conv}(M_{4/3}, z, i)$	6.1	3.11	0.84	0.72	5.74	13.03	5.62
i	15	16	17	18	19	20	...
$\text{Conv}(M_{4/3}, z, i)$	4.18	7.05	5.74	3.11	3.94	12.31	...

(b)

The resampled signals y and z are shown in **Table 3** (x_0 is a dummy signal). The composition coefficients are periodic and symmetric in each period, such as those of $y[1..5]$, $y[5..9]$, and $z[1..14]$. Although $y[1..5]$ and $y[5..9]$ are compositions of different portions of the original signal x , they can both be zeroed using the zeroing mask $M_{4/3}$. It can be further observed that the composition coefficients in a prefix and a postfix of a period of z are near those in each period of y . For example, the composition coefficients in $z[1..5]$ (a prefix of $z[1..14]$) and those in $z[10..14]$ (a postfix of $z[1..14]$) are near those in $y[1..5]$.

Thus, if we let $y_i = a_i x_{r(i)+1} + (1-a_i)x_{r(i)}$ and $z_i = b_i x_{s(i)+1} + (1-b_i)x_{s(i)}$, then $a_i \doteq b_i$ for $i = 1, 2, \dots, 5$ and $a_i \doteq b_{i+9}$, $i = 1, 2, \dots, 5$. The prefix sub-signal $z[1..5]$ has a similar composition relation to $y[1..5]$. In other words, $z[1..5]$ and $y[1..5]$ have notably similar sets of coefficients. Because the set of coefficients for $z[10..14]$ is symmetric to the set of coefficients for $z[1..5]$ and the coefficients in each set are symmetric, $z[10..14]$ and $y[1..5]$ also have similar sets of coefficients. Therefore, the convolution values of $z[1..5]$ and $z[10..14]$ with the zeroing mask $M_{4/3}$ are near zero. In general, the convolution values of the prefix sub-signal $z[1+13k..5+13k]$ and the postfix sub-signal $z[10+13k..14+13k]$ of each sub-signal $z[1+13k..14+13k]$ can be approximately zeroed using the zeroing mask $M_{4/3}$.

Table 3. (a) relations of the resampled signal y and the original signal x through the resampling factor $4/3$; (b) relations of the resampled signal z and the original signal x through the resampling factor $13/10$

$y_1 = 0x_0 + 1x_1$	$z_1 = 0x_0 + 1x_1$
$y_2 = (1/4)x_1 + (3/4)x_2$	$z_2 = (3/13)x_1 + (10/13)x_2$
$y_3 = (2/4)x_2 + (2/4)x_3$	$z_3 = (6/13)x_2 + (7/13)x_3$
$y_4 = (3/4)x_3 + (1/4)x_4$	$z_4 = (9/13)x_3 + (4/13)x_4$
$y_5 = 1x_4 + 0x_5$	$z_5 = (12/13)x_4 + (1/13)x_5$
$y_6 = (1/4)x_4 + (3/4)x_5$	$z_6 = (2/13)x_4 + (11/13)x_5$
$y_7 = (2/4)x_5 + (2/4)x_6$	$z_7 = (5/13)x_5 + (8/13)x_6$
$y_8 = (3/4)x_6 + (1/4)x_7$	$z_8 = (8/13)x_6 + (5/13)x_7$
$y_9 = 1x_7 + 0x_8$	$z_9 = (11/13)x_7 + (2/13)x_8$
$y_{10} = (1/4)x_7 + (3/4)x_8$	$z_{10} = (1/13)x_7 + (12/13)x_8$
$y_{11} = (2/4)x_8 + (2/4)x_9$	$z_{11} = (4/13)x_8 + (9/13)x_9$
$y_{12} = (3/4)x_9 + (1/4)x_{10}$	$z_{12} = (7/13)x_9 + (6/13)x_{10}$
$y_{13} = 1x_{10} + 0x_{11}$	$z_{13} = (10/13)x_{10} + (3/13)x_{11}$
$y_{14} = (1/4)x_{10} + (3/4)x_{11}$	$z_{14} = 1x_{11} + 0x_{12}$
⋮	⋮
(a)	(b)

According to the above observation, we expect that the resampled signal z can be approximately zeroed periodically. For example, the small convolution values $\text{Conv}(M_{4/3}, z, i)$ would appear for $i = 1, 10, 14, 23, \dots, 1+13k, 10+13k, \dots$, as shown in **Table 2(b)**. This fact can be used for the approximate resampling detection. In general, we assume that the signal z is the resampled signal of x with a resampling factor p'/q' . When detecting signal z using the zeroing mask $M_{p'/q'}$, where $p' > p$, the small convolution values $\text{Conv}(M_{p'/q'}, z, i)$ will appear for $i = 1, p' - p, p', 2p' - p, \dots, kp', (k+1)p' - p, \dots$. We define a convolution score cs in (25). Here, we set $t = 4$, so that the convolution score is 1 for a small convolution value and 0 otherwise. **Table 4** shows the convolution scores of a sequence of samples in the resampled signal z with the zeroing mask $M_{4/3}$. **Fig. 1** illustrates an ideal sketch of cs of a signal that is resampled using factor p'/q' and the zeroing mask $M_{p'/q'}$.

If we calculate the difference of the consecutive indices i 's where the scores of 1 (or small convolution values) appear, we can obtain the sequence $p'-p, p, p'-p, p, p'-p, p, \dots$. Moreover the values $p'-p$ and p can be extracted. Because p and q are known, and p' can be calculated using $p' = (p'-p) + p$, q' can be approximately calculated using $q' = \text{round}(p'(q/p))$ because of the assumption of $p'/q' \doteq p/q$. For example, if $p/q = 4/3$ and $p'/q' = 13/10$ (initially, the factor p'/q' is unknown), the scores of 1 will appear in $i = 0, 9, 13, 22, 26, 35, 39, \dots$, and the differences of two consecutive indices are 9, 4, 9, 4, 9, 4, \dots . We can obtain $p' = 13$ for $p = 4$ and $p'-p = 9$. Then, $q' = \text{round}(p'(q/p)) = \text{round}(13(3/4)) = \text{round}(9.75) = 10$. Therefore, we have the detected resampling factor $p'/q' = 13/10$.

$$cs(M_{p'/q'}, z, i) = \begin{cases} 1 & \text{if } |\text{Conv}(M_{p'/q'}, z, i)| < t \\ 0 & \text{else} \end{cases} \quad (25)$$

Table 4. convolution values and the corresponding scores.

i	1	2	3	4	5	6
$\text{Conv}(M_{4/3}, z, i)$	0.24	4.30	-6.70	-3.35	5.86	7.17
$cs(M_{4/3}, z, i)$	1	0	0	1	0	0
i	7	8	9	10	...	
$\text{Conv}(M_{4/3}, z, i)$	-17.57	21.51	-14.84	-3.82	...	
$cs(M_{4/3}, z, i)$	0	0	0	1	...	

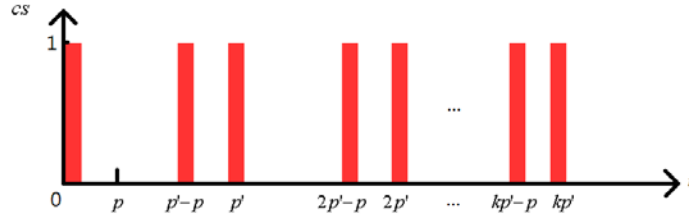


Fig. 1. An ideal sketch of $cs(M_{p/q}, z, i)$.

The proposed method to detect the approximate factor is as follows. We assume that the desired detected image is resampled using some unknown factor $f = p'/q'$, whose corresponding zeroing mask is not included in the provided set and f satisfies $12/11 < f < 12$, where 12/11 and 12 are the minimum and maximum values, respectively, of the factors that correspond to the 45 provided zeroing masks. The detections for horizontal resampling and vertical resampling are similar. Here, we only discuss the former. When the zeroing mask $M_{p/q}$ is used to detect an image I , the scores are evaluated, and the positions where 1s occur are extracted. The detection method is successful if the resampling factor is near one in the set of 45 factors; otherwise it tends to fail. In fact, the 45 factors are unevenly distributed on the interval [1, 12]. If we plot the factor values on the real line, and if f falls in a sparse region, the detection tends to fail. Half of the factors are in the subinterval [1, 2], and only 11 factors fall in [4, 12]. For $f = 1.82$, because it is in a dense region, where the two factors 9/5 and 11/6 are notably near it, it can be successfully detected. For $f = 6.5$, because it is in a sparse region and neither of the closest factors (6 and 7) is sufficiently near it, the detection will fail. To address this problem, we add two zeroing masks, $M_{19/3}$ and $M_{20/3}$, for detection, the factors of which are both near 6.5. Similarly, in other sparse regions, we add some more corresponding zeroing masks, to raise the detection rate. For each sparse region $[j, j+1]$, $j = 6, 7, \dots, 11$, we add two factors $j + 1/3$ and $j + 2/3$. Thus, 12 zeroing masks are derived and added to the set of zeroing masks for detection. As a result, there are 57 ($= 45 + 12$) provided zeroing masks for detection, so that the resampling factors between 12/11 and 12 can be detected. This set is sufficient for practical use.

4. Experimental Results

In our experiments, 100 natural images were used for testing. Each of these 100 images was first resampled using one of 110 different resampling factors, which ranged from 1.1 to 12 with a step size of 0.1. As a result, there were 11,000 resampled test images. The exact detection is always performed first. If the exact detection fails, the approximate detection is performed. If an image is resampled using a factor that corresponds to one of the provided zeroing masks, it can be 100% successfully detected in the first stage. Otherwise, it would be successfully approximately detected in the second stage if its resampling rate is near one of the factors that correspond to the provided zeroing masks. As defined in (26), the error rate of the detected factor in this example is 10%. The recovery rates of the factors vs. the tolerance of error rates, which ranged from 1.1 to 3.0, are illustrated in Fig. 2. We can observe that a higher error rate for tolerance corresponds to a higher recovery rate.

$$\text{error rate} = \left| \frac{\text{detected factor} - \text{correct factor}}{\text{correct factor}} \right| \quad (26)$$

For a resampled image whose resampling factor is not sufficiently near the factor of any provided zeroing mask, the approximate detection tends to fail. This problem can be addressed by providing more zeroing masks so that the resampling factors of all provided masks are distributed over the specific interval with no sparse portions. Because $1 \leq q < p \leq 12$, the minimum and maximum values of the provided p/q are 12/11 and 12, respectively, and there are 45 factors in the interval [1, 12]. More than half of the provided factors are in the interval [1, 3], and the remainder is sparsely distributed in the interval [3, 12]. A factor in [3, 12] is likely far from all provided factors and may be more difficult to detect. To address this problem, we provide some more zeroing masks for the factors in the interval [3, 12], as follows. In the interval $[k, k+1]$, $6 \leq k \leq 11$, zeroing masks are added for factors $k+1/3$ and $k+2/3$. As a result, 12 zeroing masks are added, and the total number of zeroing masks for detection is increased to 57.

After adding 12 more zeroing masks to eliminate the sparse regions in the interval [1.1, 12], any resampling factor in the interval [1.1, 12] is near at least one of the factors that correspond to the provided zeroing masks, so that it can be detected.

Fig.s 3(a)-(c) and **3(d)-(f)** show the recovery rates for the resampling factors in [6, 12] using the set of 45 zeroing masks and the increased set of 57 zeroing masks, respectively. The use of more zeroing masks indeed improves the recovery rates. As shown in **Fig. 3(a)**, with 45 zeroing masks, the recovery rates with a tolerance of 0% error rate for the resampling factors 6.3 and 6.6 are 0% and 1%, respectively. As shown in **Fig. 3(d)**, with 57 zeroing masks, the detection rates with a tolerance of 0% error rate for the resampling factors 6.3 and 6.6 are increased to 35% and 22%, respectively.

As shown in **Table 5**, the recovery rate for the resampling factors is up to 97.74%, with a tolerance of error rate of 20%. However, the detection rate of resampling without any tolerance is 100%.

In most cases, the errors are produced by the “round” operation. For example, when performing the approximate detection on an image that was resampled using a factor of 2.1 (= 21/10) with the zeroing mask of factor 2, which is near 2.1, the periodicity of $p' = 21$ is found, and q' is evaluated as $round(21/2) = round(10.5) = 11$ to yield the approximate factor 21/11 (≈ 1.9).

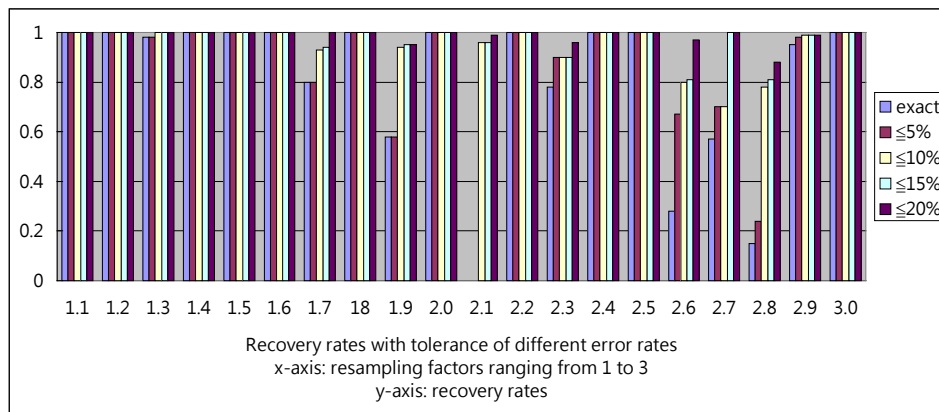
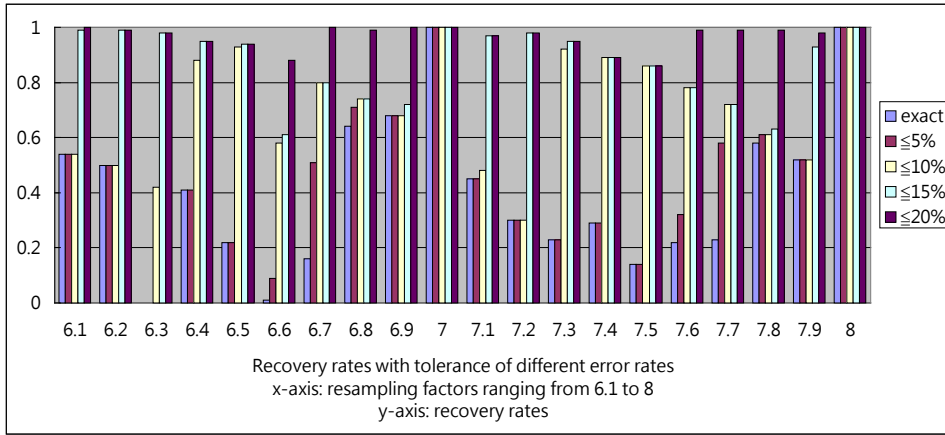
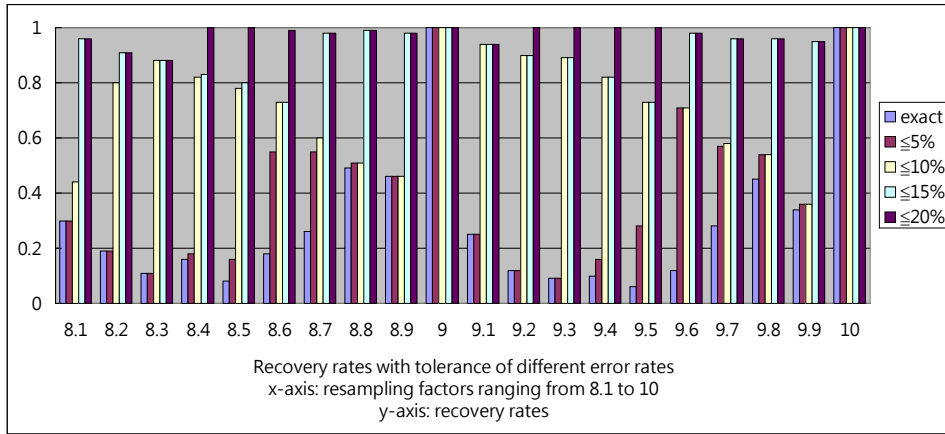


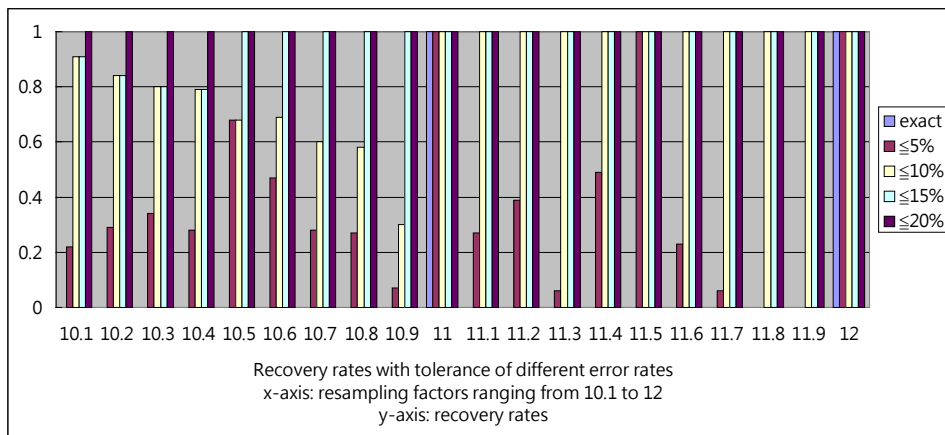
Fig. 2. Recovery rates vs. tolerance of error rates.



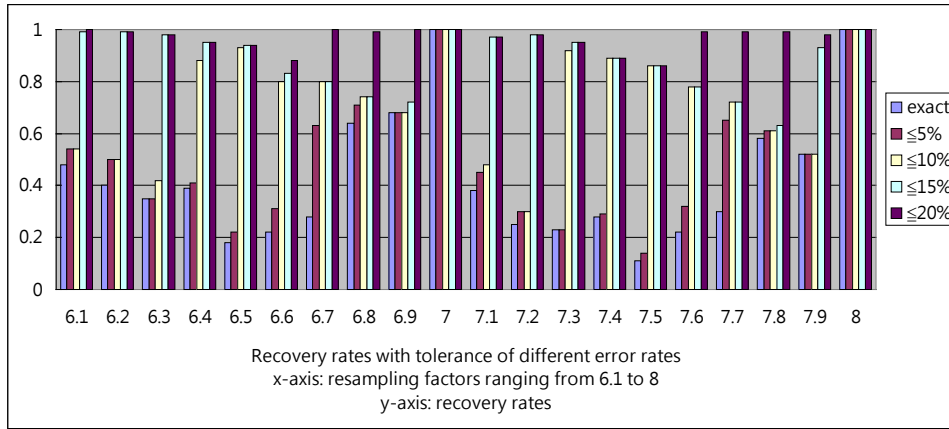
(a)



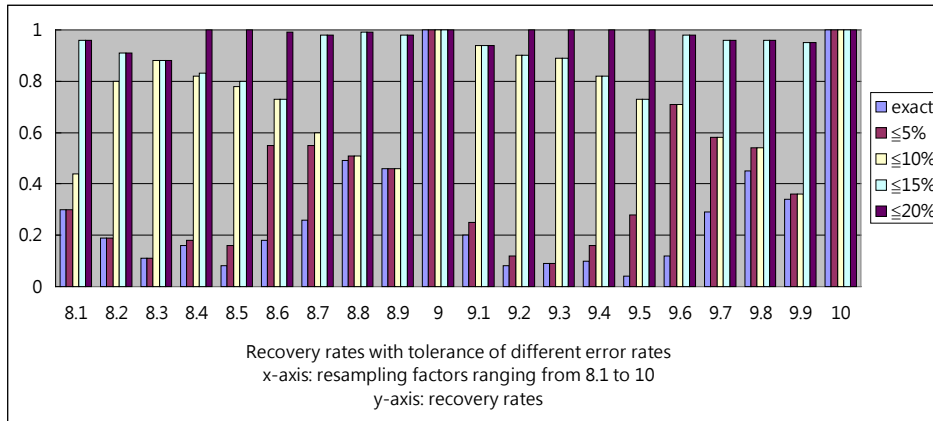
(b)



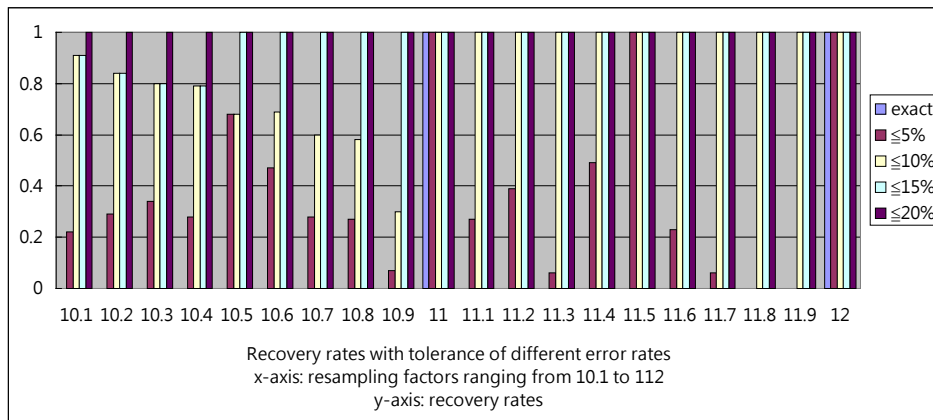
(c)



(d)



(e)

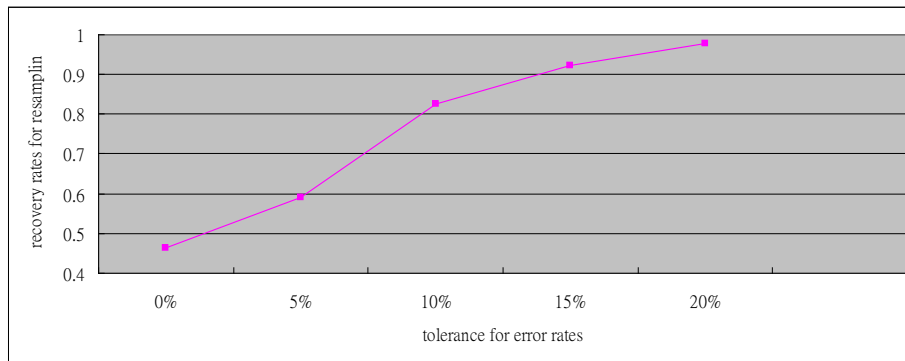


(f)

Fig. 3. (a)-(c) detection results with 45 zeroing masks (d)-(f) detection results with 57 zeroing masks.

Table 5. Recovery rates of resampling factors vs. tolerance of error rates (11100 test images).

tolerance of error rates	0%	$\leq 5\%$	$\leq 10\%$	$\leq 15\%$	$\leq 20\%$
recovery rates of resampling factors	46.31%	59.13%	82.58%	92.17%	97.74%

**Fig. 4.** Sketch of recovery rates of resampling factors vs. tolerance of error rates

5. Conclusion and Future Work

Improving upon our previous work, we have presented a new method to detect image resampling and recover the resampling factors, which detects many more resampling factors using the same set of zeroing masks. This improved method detects image resampling by checking for some periodicity that is generated by convolution with zeroing masks using an approximate mechanism. The proposed method detects whether an image has been resampled and recovers an approximate corresponding resampling factor.

Although the method that was proposed by Lien et al. [19] can recover the resampling factors, it can only recover the resampling factors whose weighting tables are provided, and it must use weighting tables to recover the possible original image and examine the periodicity. Furthermore, their work is notably time-consuming, and false positives may occur when the set of weighting tables is not sorted in specific orders.

Although our previous work is much more efficient and effective than the method proposed by Popescu and Farid [1], it can only detect the resampling factors that correspond to the factors of the provided zeroing masks. For example, 57 factors in the interval [1.1, 12] were used in the experiment. Associated with the approximation detection mechanism, the proposed method in this paper can detect all resampling factors in the interval [1.1, 12]. Theoretically, there is an infinite number, many more than 57, of factors in the interval [1.1, 12]. The experimental results have demonstrated that the proposed method is indeed effective and efficient. The accuracy rate for resampling detection is 100%. The recovery rates for the resampling factors are 46.31%, 59.13%, 82.58%, 92.17%, and 97.74% for tolerance of error rates of 0%, 5%, 10%, 15%, and 20% respectively. Many false positives are generated for the test images that are too smooth because the resampling score for a smooth region with a zeroing mask tends to be high. Thus, images that are too smooth are likely to be zeroed using any zeroing mask. The average time required to detect an image is 0.52 seconds. In the future, we would like to improve our method to combat other types of modifications, such as rotation, Gaussian noise, and gamma correction.

References

- [1] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Transactions on Signal Processing*, Vol. 53, pp. 758-767, 2005. [Article \(CrossRef Link\)](#)
- [2] M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue," in *Proc. of ACM Multimedia and Security Workshop*, pp. 11-20, 2008. [Article \(CrossRef Link\)](#)
- [3] Peter Nillius and Jan-Olof Eklundh, "Automatic estimation of the projected light source direction," in *Proc. of 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 1076-1083, 2001. [Article \(CrossRef Link\)](#)
- [4] M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *Proc. of ACM Multimedia and Security Workshop*, New York, pp. 1-9, 2005. [Article \(CrossRef Link\)](#)
- [5] Kwang-Fu Li, Tung-Shou Chen, and Seng-Cheng Wu, "Image tamper detection and recovery system based on Discrete Wavelet Transformation," in *Proc. of 2001 IEEE Pacific Rim Conference on Intelligent Conference Communications, Computers and Signal Processing*, pp. 26-28, Vol. 1, 2001. [Article \(CrossRef Link\)](#)
- [6] Phen Lan Lin, Chung-Kai Hsieh, and Po-Whei Huang, "A hierarchical digital watermarking method for image tamper detection and recovery," *Pattern Recognition*, Vol. 38, Issue 12, pp. 2519-2529, 2005. [Article \(CrossRef Link\)](#)
- [7] Ming-Shi Wang and Wei-Che Chen, "A majority-voting based watermarking scheme for color image tamper detection and recovery," *Computer Standards & Interfaces*, Vol. 29, No. 5, pp. 561-570, July 2007. [Article \(CrossRef Link\)](#)
- [8] M. K. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *Proc. of ACM Multimedia and Security Workshop*, New York, 2006, pp. 48-55. [Article \(CrossRef Link\)](#)
- [9] N. Khanna, A. K. Mikkilineni, G. T. C. Chiu, J. P. Allebach, and E. J. Delp, "Scanner identification using sensor pattern noise," in *Proc. of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents IX*, Vol. 6505, No. 1, pp. 65051K, 2007. [Article \(CrossRef Link\)](#)
- [10] J. Lukas, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," in *Proc. of the SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, pp. 362-372, Vol. 6072, January 2006. [Article \(CrossRef Link\)](#)
- [11] J. Lukas, J. Fridrich, and M. Goljan, "Determining digital image origin using sensor imperfections," in *Proc. of SPIE Electronic Imaging, Image and Video Communication and Processing*, pp. 249-260, January 16-20, 2005. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.117.6754> [Article \(CrossRef Link\)](#)
- [12] N. Khanna, G. T. C. Chiu, J. P. Allebach, and E. J. Delp, "Scanner identification with extension to forgery detection," in *Proc. of SPIE Electronic Imaging*, San Jose, CA, Vol. 6819, Feb. 2008. [Article \(CrossRef Link\)](#)
- [13] E. S. Gopi, N. Lakshmanan, T. Gokul, S. KumaraGanesh, and P. R. Shah, "Digital Image Forgery Detection using Artificial Neural Network and Auto Regressive Coefficients," in *Proc. of Electrical and Computer Engineering*, 2006 Canadian Conference (CCECE'06), pp. 194-197, May 7-10, 2006. [Article \(CrossRef Link\)](#)
- [14] A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting duplicated image regions," *Technical Report*, TR2004-515, Dartmouth College, Computer Science 2004. [Article \(CrossRef Link\)](#)
- [15] Guohui Li, Qiong Wu, Dan Tu, and Shaojie Sun, "A sorted neighborhood approach for detecting duplicated regions in image forgeries based on DWT and SVD," in *Proc. of IEEE International Conference on Multimedia and Expo.*, Beijing China, pp. 1750-1753, July 2-5, 2007. [Article \(CrossRef Link\)](#)
- [16] Weiqi Luo, Jiwu Huang, and Guiping Qiu, "Robust detection of region-duplication forgery in digital image," in *Proc. of the 18th International Conference on Pattern Recognition*, Vol. 4, 2006, pp. 746-749. [Article \(CrossRef Link\)](#)

- [17] Hwei-Jen Lin, Chun-Wei Wang, and Yang-Ta Kao, "Fast copy-move forgery detection," *WSEAS Transactions on Signal Processing*, Vol. 5, No. 5, pp. 188-197, 2009. [Article \(CrossRef Link\)](#)
- [18] A. N. Myna, M. G. Venkateshmurthy, and C. G. Patil, "Detection of region duplication forgery in digital images using Wavelets and Log-Polar Mapping," in *Proc. of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, pp. 371-377, December 13-15, 2007. [Article \(CrossRef Link\)](#)
- [19] Cheng-Chang Lien, Cheng-Lun Shih, Chih-Hsun Chou, "Fast Forgery Detection with the Intrinsic Resampling Properties," *Journal of Information Security*, 2010, 1, pp. 11-22. [Article \(CrossRef Link\)](#)
- [20] Y. T. Kao, H. J. Lin, C. W. Wang, and Y. C. Pai, "Effective Detection for Linear Up-Resampling by Factor of Fraction," *IEEE Transactions on Image Processing*, Vol. 21, Issue 8, pp. 3443-3453, August, 2012. [Article \(CrossRef Link\)](#)



Ching-Ting Tu received her B.S. and Ph.D. degrees in computer science and information engineering from National Cheng Kung University, Tainan, Taiwan, in 2004 and 2010. In 2011, she joined the department of computer science and information engineering at Tamkang University, Taiwan, as an assistant professor. Her research interests include computer vision and pattern recognition, human-computer interaction, and multimedia information analysis and retrieval.



Hwei-Jen Lin received the B.S. degree in Applied Mathematics from National Chiao Tung University, Hsinchu, Taiwan in 1981 and the M.S. and the Ph.D. degrees in Mathematics from Northeastern University, Boston, U.S.A. in 1983 and 1989, respectively. She is currently a professor at the department of computer science and information engineering of Tamkang University. Her current research interests include Pattern Recognition, Image Processing, and Computer Algorithms.



Fu-Wen Yang received the B.S. degree in Industrial Education from National Taiwan Normal University, Taipei, Taiwan in 1988 and the M.S. and the Ph.D. degrees in Information Engineering from Tamkang University, Taipei, Taiwan in 1998 and 2005, respectively. He is currently a teacher at the Department of Information Engineering of New Taipei Municipal Tam-Shui Vocational High School, and a part-time assistant professor at the department of computer science and information engineering of Tamkang University as well. His research interests include image processing, pattern recognition and computational intelligence.



Hsiao-Wei Chang received the B.S. degree in Applied Mathematics from Fu-Jen Catholic University, Taipei, Taiwan in 1980, the M.S. degree in Computer Science from Texas A&I University, Texas, U.S.A. in 1986 and the Ph.D. degree in Computer Science and Information Engineering from Tamkang University, Taipei, Taiwan, R.O.C. in 2011. He is currently an associate professor at the department of computer science and information engineering of China University of Science and Technology, Taipei, Taiwan. His current research interests include Pattern Recognition, Image Processing and Logic Design.