EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Screenshot identification by analysis of directional inequality of interlaced video

Ji-Won Lee[1], Min-Jeong Lee[2], Hae-Yeoun Lee[3] and Heung-Kyu Lee[4*]

Abstract

As screenshots of copyrighted video content are spreading through the Internet without any regulation, cases of copyright infringement have been observed. Further, it is difficult to use existing forensic techniques for determining whether or not a given image was captured from a screen. Thus, we propose a screenshot identification scheme using the trace of screen capture. Since most television systems and camcorders use interlaced scanning, many screenshots are taken from interlaced videos. Consequently, these screenshots contain the trace of interlaced videos, combing artifacts. In this study, we identify a screenshot using the characteristics of combing artifacts that appear to be shaped like horizontal jagged noise and can be found around the edges. To identify a screenshot, the edge areas are extracted using the gray level co-occurrence matrix (GLCM). Then, the amount of combing artifacts is calculated in the extracted edge areas by using the similarity ratio (SR), the ratio of the horizontal noise to the vertical noise. By analyzing the directional inequality of noise components, the proposed scheme identifies the source of an input image. In the experiments conducted, the identification accuracy is measured in various environments. The results prove that the proposed identification scheme is stable and performs well.

Keywords: combing artifacts, directional inequality, interlaced video, screenshot identification

1 Introduction

With a more capable Internet than ever before, many people have started to collect and share information about their interests through the Internet. Multimedia content such as movies, television programs, and user generated contents (UGCs) are among the content that attracts the greatest common interest. To collect and share multimedia content information, many people use screenshots as well as the original video content. Since social networking sites (SNSs) such as MySpace, Twitter, and Facebook have become extremely popular, this tendency is growing faster. We can easily find many screenshots of varied video content from these SNSs. The problem is that many screenshots are taken from copyrighted video content without any permission. Further, additional copyright infringements take place, when people share and distribute these screenshots without any notification to the content provider.

The trusted computing group (TCG), a not-for-profit organization of global IT companies, states that releasing screenshots of copyrighted video content to the public is copyright infringement [1]. This means that not only the video content but also the screenshots taken from them are subject to a copyright. However, most people are not aware that it is illegal to use screenshots of copyrighted video content. Even if someone knows that screenshots may have a copyright, it is difficult to distinguish screenshots from nonscreenshots by the naked eye. In here, nonscreenshot means the image that is not a screenshot. To demonstrate that humans have difficulties in distinguishing between screenshots and nonscreenshots, we conducted a subjective test. For the subjective test, we used 100 screenshots and 100 non-screenshots. We shuffled 200 test images, then each image was presented in 3 s and 8 participated observers chose the origin of the given image after watching that image. Table 1 shows the subjective test results. As shown in the results, accuracies were around 50%, which is similar to accuracy of random selection (50%).

* Correspondence: hklee@mmc.kaist.ac.kr
[4]Department of Computer Science and Division of Web Science and Technology, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea
Full list of author information is available at the end of the article

Table 1 Subjective test results when 200 test images (100 screenshots and 100 nonscreenshots) were given

observer number	1	2	3	4	5	6	7	8
# Correct screenshots	41	64	37	34	35	56	33	58
# Correct nonscreenshots	65	41	64	69	69	57	68	49
Accuracy	53%	52.5%	50.5%	51.5%	52%	56.5%	50.5%	53.5%

If there were a technique for identifying screenshot images, people can be cautioned to check first the copyright before uploading a screenshot to Internet. Furthermore, we can retrieve the source video content of that screenshot using video retrieval techniques. A detailed scenario is depicted in Figure 1. Also, we could think of different scenarios. Some people upload screenshots for selling or distributing illegally recorded content using peer to peer (P2P) or torrent sites. In this case, if we could check the origin of the uploaded images, we could send information of malicious users to the webmaster or the content owner for further action against the malicious users. To provide a practical monitoring scheme, we propose an identification scheme that can distinguish whether a given image is a screenshot or nonscreenshot.

There have been a few techniques for identifying the sources of input images. In [2-10], techniques were proposed for distinguishing photographic images and computer graphics (CG) using the statistical characteristics of natural images. Further, the approaches to distinguish recaptured images and natural images were suggested in [11-13]. Similarly, we focused on screenshots as the source of input images. The screenshot identification scheme was first proposed in our previous study [14]. We had extracted features from the wavelet domain and differential histograms to detect screenshots. The extracted features were then used to train and test the support vector machine (SVM) classifier. The identification accuracy in our previous study was high; however, there were inevitable problems related with the SVM classifier. The training process of the classifier took a long time due to time-consuming feature selection and extraction stages. Also, if the test environment of the classifier is different with the trained one, a new training process is needed to get the highest identification accuracy.

Therefore, we propose an identification scheme that distinguishes whether the test image is a screenshot or not without the SVM classifier support. To achieve our purpose, we introduce the concept of "similarity ratio"

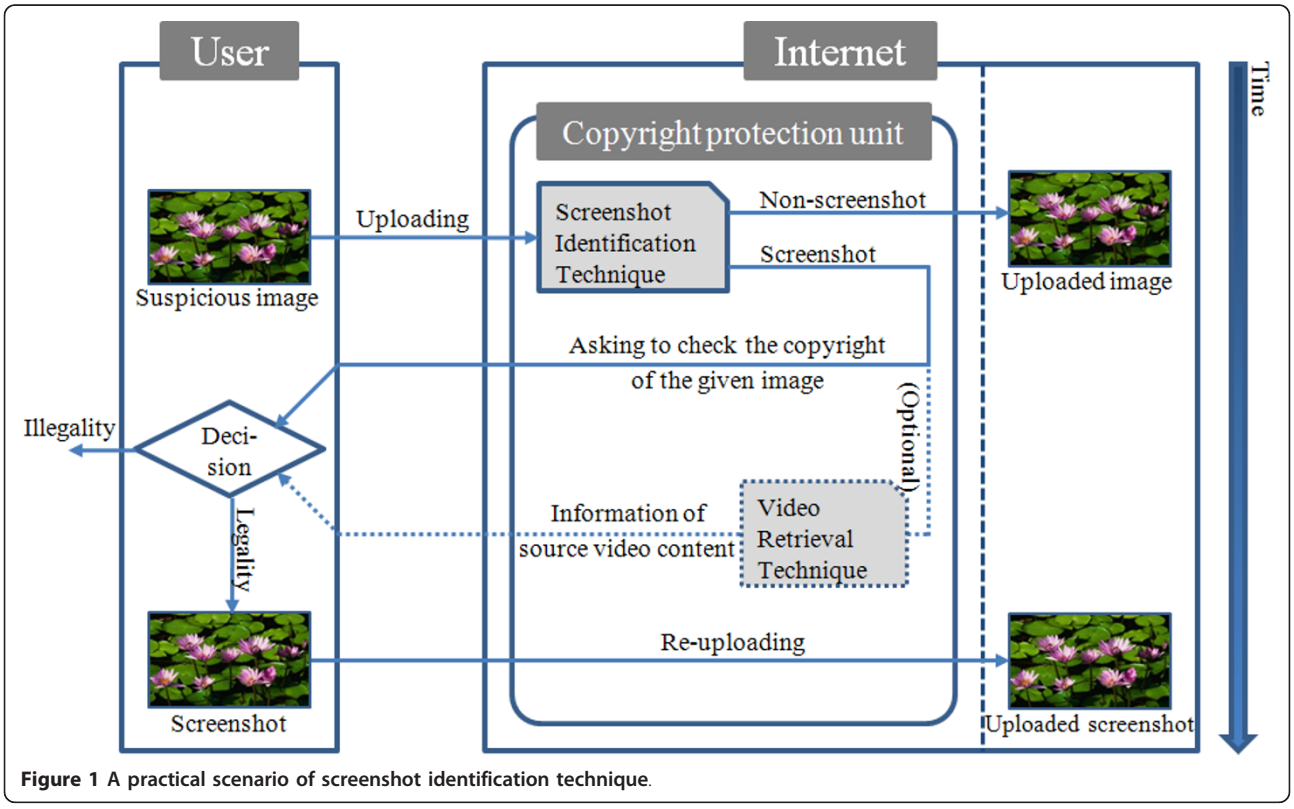

Figure 1 A practical scenario of screenshot identification technique.

(SR) as a wavelet-motivated measure. Since the similarity ratio is statistically calculated by analyzing the innate characteristics of an inter-laced screenshot, the proposed approach achieves good adaptability and does not repeatedly require new training process.

The remainder of this article is organized as follows. Section 2 introduces combing artifacts, a unique characteristic of interlaced video. Section 3 explains three subprocesses of the proposed scheme. Section 4 presents the experimental results to prove the effectiveness and adaptability of the proposed scheme. Finally, Section 5 presents the concluding remarks.

2 Combing artifacts

There are two primary types of scanning modes used in modern display devices: interlaced scanning and progressive scanning. Interlaced scanning draws odd scan lines of the full resolution frame at time t, $F(x, y, t)$, and even scan lines of the full resolution frame at time t+1, $F(x; y, t+1)$. One-half of a full resolution frame at time t is called a field $f(x, y, t)$ [15]. On the other hand, progressive scanning displays all lines of a full resolution frame $F(x, y, t)$ at time t in sequence. Figure 2 illustrates these scanning modes.

Since interlaced scanning uses just one-half of a frame at any given time, the video quality is worse compared to that for progressive scanning. Further, interlaced scanning has horizontal jagged noise due to weaving of the two fields. The spatial quality of interlaced scanning may be worse than that of progressive scanning, however, the temporal resolution is higher than that of progressive scanning. Also, it consumes only one-half of the bandwidth compared to that in the case of progressive scanning. Further, cathode ray tube (CRT)-based televisions cannot adopt the progressive scanning mode owing to their technical limitations. Thus, interlaced scanning is still widely used in various television encoding systems and camcorder recording modes, in spite of unavoidable shortcomings. Standard definition television (SDTV) uses one of the three analog television encoding standards known as NTSC, PAL, and SECAM. All of them use interlaced scanning. In the case of camcorders, both scanning modes are supported during recording, but interlaced scanning is set as the default scanning mode in most camcorders.

As shown in Figure 3, an interlaced frame $F(x, y, t)$ is created by simply weaving the even field $f(x, y, t\text{-}1)$ and the odd field $f(x, y, t)$. Since an interlaced video is created by weaving two fields together, the video contains some horizontal jagged noise due to motion, this noise is referred to as combing artifacts. The magnitude of combing artifacts is larger when the motion between the adjacent fields is greater, and is commonly seen around the vertical edges of moving objects. Figure 4 shows one such example of combing artifacts caused by interlaced scanning. Since combing artifacts are inherently introduced in an interlaced video, a screenshot of the interlaced video also has traces of these combing artifacts. In this study, we use the combing artifacts of a screenshot as evidence of interlaced video capturing.

3 Proposed scheme

Since the screenshots of an interlaced video have traces of interlaced scanning, we exploit this clue to distinguish a screenshot, when a test image is given. To do this, we

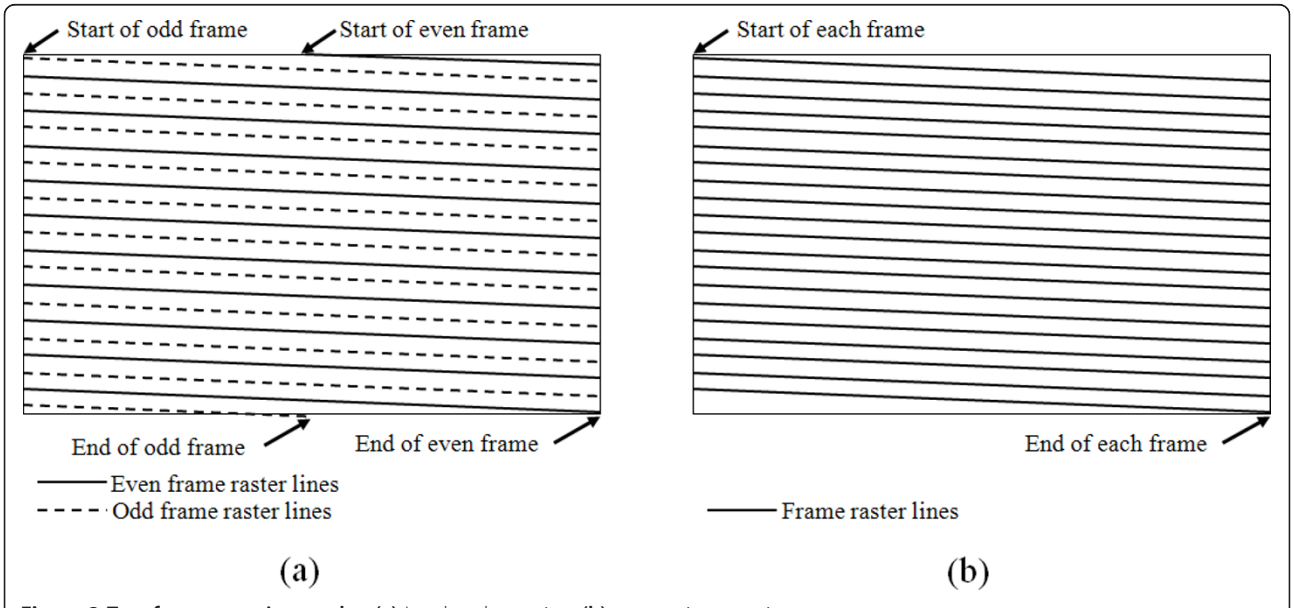

Figure 2 Two frame scanning modes. **(a)** Interlaced scanning, **(b)** progressive scanning.

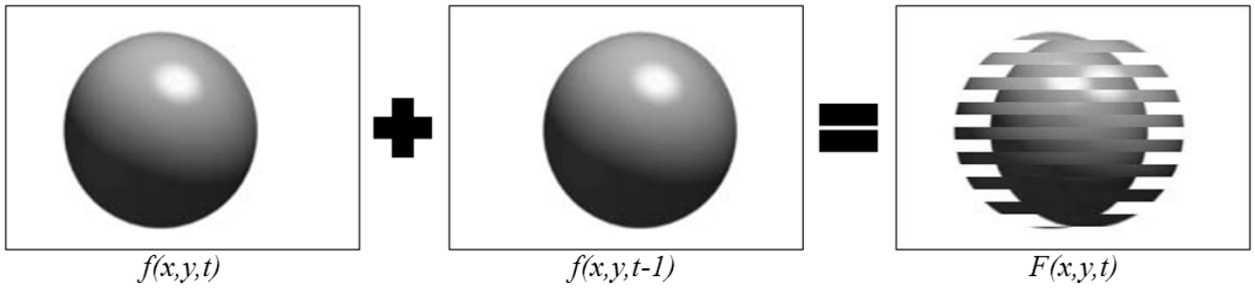

Figure 3 An interlaced frame *F*(*x*, *y*, *t*) is created by weaving the even field *f*(*x*, *y*, *t*) and the odd field *f*(*x*, *y*, *t*-1).

define a measure that expresses combing artifacts clearly. In this study, we define an SR that exploits the directional inequality of the noise distribution due to combing artifacts, in order to identify a screenshot. The screenshot identification process consists of three steps: finding edge blocks, measuring the directional inequality, and determining the image source. An overview of the proposed screenshot identification scheme is presented in Figure 5.

3.1 Screenshot identification process: finding edge blocks

One possible way of identifying a test image as a screenshot is to measure the amount of combing artifacts. To do this, we first extract the areas where combing artifacts may exist. As we mentioned before, combing artifacts are usually found around the edges of an image. Therefore, the first step is to find the edge areas for identifying a screenshot.

A gray level co-occurrence matrix (GLCM) is proposed for the statistical analysis of pixel-based texture [16]. Given direction and distance between two adjacent pixels of an image, the GLCM is defined as the distribution of co-occurring luminance values at a given offset. Since the GLCM can describe the textural characteristics clearly in a given image, we use the GLCM to extract the edge areas from the given image.

To extract the edge areas, the input image I is split into small blocks with an $m \times m$ pixel size, where m is a preset integer. Then, the GLCM is applied in each block B_a, where $0 \le a \le n\text{-}1$ and n is the number of blocks. If m is too small, the calculated GLCM cannot represent the edge areas sufficiently. On the other hand, if m is too large, almost all GLCM features become similar. This means that the selection of block size affects the identification accuracy. In our study, m is experimentally selected to get the highest identification performance. After that, we calculate the two-directional GLCMs$\left(0, \frac{\pi}{2}\right)$ in each block B_a to accurately identify both the horizontal and vertical edges. In mathematical terms, we have

$$\mathrm{GLCM}_H^{B_a}(i,j) = \sum_{p=1}^{m}\sum_{q=1}^{m} \begin{cases} 1, \text{ if } B_a(p,q) = i \text{ and } B_a(p+1,q) = j \\ 0, \text{ otherwise} \end{cases} \quad (1)$$

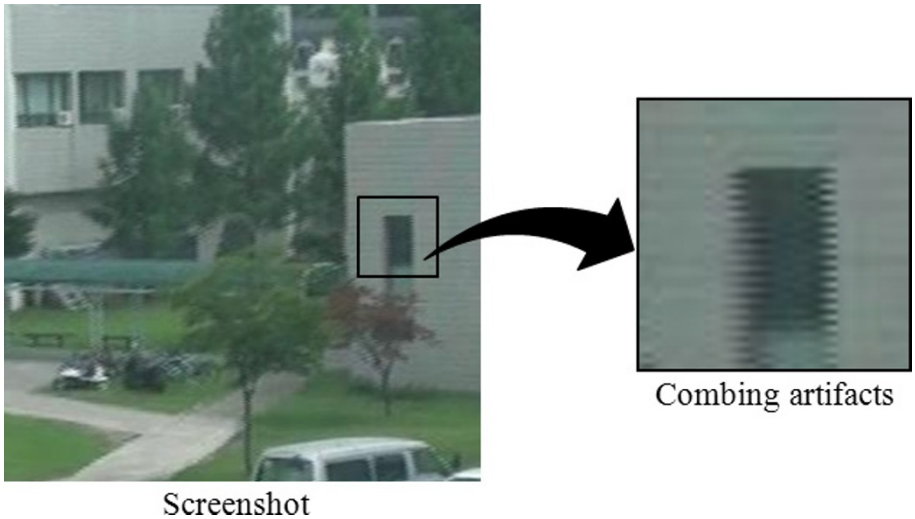

Figure 4 An example of combing artifacts.

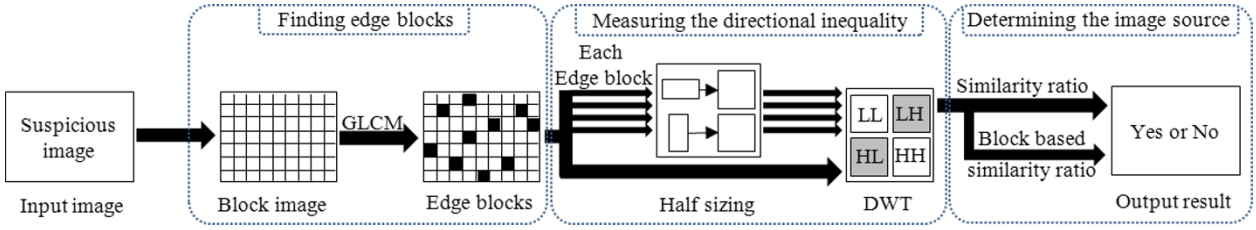

Figure 5 Overview of screenshot identification scheme.

$$\mathrm{GLCM}_V^{B_a}(i,j) = \sum_{p=1}^{m}\sum_{q=1}^{m}\begin{cases} 1, \text{ if } B_a(p,q) = i \text{ and } B_a(p,q+1) = j \\ 0, \text{ otherwise} \end{cases} \quad (2)$$

Here, $\mathrm{GLCM}_H^{B_a}$ and $\mathrm{GLCM}_V^{B_a}$ are 0 and $\frac{\pi}{2}$ directional GLCMs of block B_a, respectively.

Figure 6 shows the distributions of GLCMs for the case in which plain, slightly textured, and strongly textured blocks are input. Here, slightly textured and strongly textured blocks indicate the blocks which have small and large amount of edge components, respectively. As shown in the figure, the distribution of GLCM is more dispersed from the line with a slop of $\frac{\pi}{4}$, when the block has a larger textured area. We use this property to discriminate the edge blocks from other given blocks. For a block B_a, the decision formula D for identifying an edge block is as follows:

$$D = \begin{cases} \text{edge}, & \text{if} \dfrac{\sum_{|i-j|\geq Th_1}\left(\mathrm{GLCM}_H^{B_a}(i,j) + \mathrm{GLCM}_V^{B_a}(i,j)\right)}{\sum_{i,j}\left(\mathrm{GLCM}_H^{B_a}(i,j) + \mathrm{GLCM}_V^{B_a}(i,j)\right)} \geq Th_2 \\ \text{non - edge, otherwise} \end{cases} \quad (3)$$

Here, Th_1 represents the maximum allowable luminance difference between two adjacent pixels. If exceeds Th_1, we decide that an edge component exists in that block. Th_2 represents the proportion of the edge component in a block. Briefly, Th_1 and Th_2 represent the quality and quantity of the edge component, respectively. If a certain block satisfies the above decision formula D, we decide that the block is an edge block. Each extracted edge block is denoted as E_b, where $0 \leq b \leq k\text{-}1$ and k is the number of total edge blocks. These extracted edge blocks are used in the next step to calculate the directional inequality.

3.2 Screenshot identification process: measuring the directional inequality

There are two basic types of de-interlacing algorithms: field combination and field extension [15]. In the field extension type, there is a de-interlacing method called vertical half-sizing. In this method, each interlaced field is displayed separately, resulting in a video with half the vertical resolution of the original one, this alleviates the problem of combing artifacts. This method is implemented by deleting all the even or odd lines of the interlaced frame. It can eliminate most combing artifacts but it severely degrades the video quality and breaks the aspect ratio, and hence, it is not widely used for de-interlacing. We focused on the powerful de-interlacing ability of vertical half-sizing and used it as the basis of our scheme to separate the screenshots and nonscreenshots.

In general, the luminance value of a certain pixel of block E_b from a nonscreenshot is highly correlated with that of the vertically and horizontally adjacent pixels, so the difference value between the adjacent pixels is around zero. The values of horizontally adjacent pixels of E'_b from the screenshot are also highly correlated with each other. However, the values of vertically adjacent pixels are not correlated, owing to the combing artifacts [14]. If E_b is vertically downsized by a factor of 2:1 and then interpolated, we get a similar interpolated blocks E_{b_v} to E_b. On the other hand, if E'_b undergoes the same process as E_b, we get the block E'_{b_v} without the combing artifacts from E'_b because most of the horizontal jagged noise is removed by the vertical half-sizing. On the other hand, if E_b and E'_b are horizontally downsized by a factor of 2:1 and then interpolated, we get similar interpolated blocks E_{b_h} and E'_{b_h} to the input blocks E_b and E'_b, respectively. The reason is that the pixel values of both the nonscreenshot and screenshot are highly correlated for horizontally adjacent pixels. Thus, the amount of vertical jagged noise removed by horizontal half-sizing is small. Figure 7 shows the example images of the two processes mentioned above. As shown in Figure 7a, two interpolated blocks E_{b_v} and E_{b_h} are similar to the edge block E_b from a nonscreenshot. In contrast, in Figure 7b, the horizontally interpolated block E'_{b_h} is similar to the edge block E'_b from a screenshot, whereas the vertically interpolated block E'_{b_v} is quite different from E'_b. We exploit this dissimilarity between E'_b and E'_{b_v} to identify the image source.

To calculate the similarity of directional noise between a given edge block and its vertically and horizontally interpolated blocks, we use the low-high (LH) and high-low (HL) subband images of the discrete wavelet

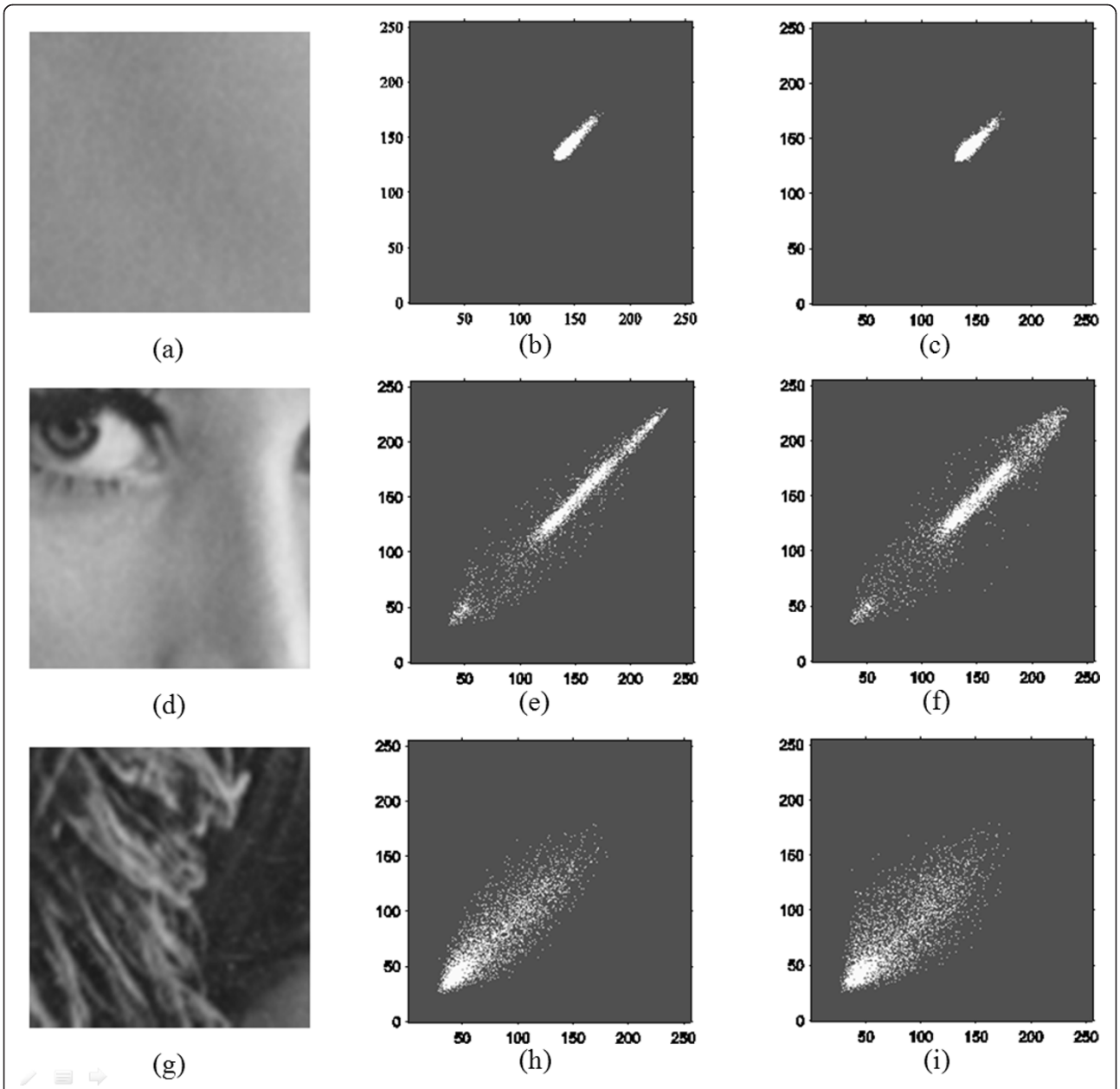

Figure 6 The distributions of $GLCM_H$ and $GLCM_V$ of various blocks. **(a)** Plain block, **(b)** $GLCM_H$ of (a), **(c)** $GLCM_V$ of (a), **(d)** slightly textured block, **(e)** $GLCM_H$ of (d), **(f)** $GLCM_V$ of (d), **(g)** strongly textured block, **(h)** $GLCM_H$ of (g), **(i)** $GLCM_V$ of (g).

transform (DWT) decomposition. For the orthogonal wavelet transform, one level of decomposition is used, and the wavelet employed is Daubechies' symmlet with sixteen vanishing moments [17]. LH and HL subband images represent the horizontal and vertical noise of the input image, respectively. When an edge block E_b and its vertically and horizontally interpolated blocks, i.e., E_{b_v} and E_{b_h}, respectively, are given, the sum of the absolute values of each LH or HL subband image element is calculated to measure the amount of the directional noise component of each input block. Using the ratio of the calculated sum values, we can estimate the similarities of directional noise between E_b and E_{b_v}, and E_b and E_{b_h}. More precisely, we have

$$Sim_{b\_h} = \frac{\sum |LH(E_{b\_v})|}{\sum |LH(E_b)|}$$
$$Sim_{b\_v} = \frac{\sum |LH(E_{b\_h})|}{\sum |LH(E_b)|} \quad (4)$$

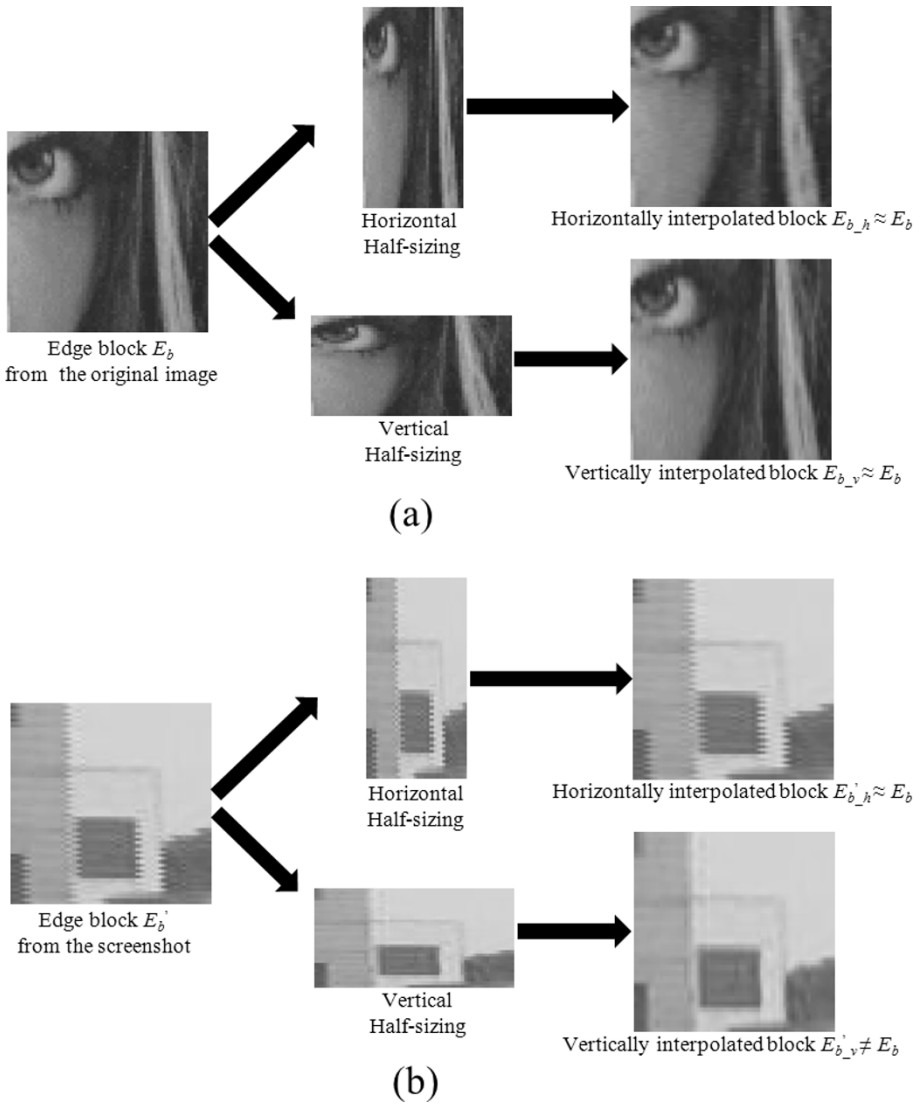

Figure 7 Horizontal and vertical half-sizing process of the given edge block. **(a)** Half-sizing process of an edge block from the nonscreenshot, **(b)** half-sizing process of an edge block from the screenshot.

where Sim_{b_h} is the similarity of the horizontal noise between E_b and E_{b_v}, and it measures a change in the horizontal noise component before and after vertical half-sizing. In the same manner, Sim_{b_v} is the similarity of the vertical noise between E_b and E_{b_h}, and it measures a change in the vertical noise component before and after horizontal half-sizing. If E_b is from a nonscreenshot, both Sim_{b_h} and Sim_{b_v} are similar to each other. On the other hand, if E_b is from a screenshot, Sim_{b_h} is much lower than Sim_{b_v} owing to the removal of combing artifacts by the vertical half-sizing process. Each edge block E_b has its similarities Sim_{b_h} and Sim_{b_v}. The directional inequality of the noise component is inferred using the *SR* of all edge blocks.

$$SR = \frac{\sum_{b=0}^{k-1} Sim_{b\_h}}{\sum_{b=0}^{k-1} Sim_{b\_v}} = \frac{\sum_{b=0}^{k-1} \left(\frac{\Sigma |LH(E_{b\_v})|}{\Sigma |LH(E_b)|} \right)}{\sum_{b=0}^{k-1} \left(\frac{\Sigma |HL(E_{b\_h})|}{\Sigma |HL(E_b)|} \right)} \quad (5)$$

where k is the number of edge blocks. If the input image is a nonscreenshot, the numerator and denominator of the *SR* have similar values. However, the numerator and denominator of the *SR* are quite different when

the input image is a screenshot. Thus, we can infer the directional noise inequality by using the calculated *SR*.

3.3 Screenshot identi cation process: determining the image source

3.3.1 Global directional inequality detection

When an edge block E_b is given, E_{b_h} and E_{b_v} have a lower edge component value compared to E_b owing to the half-sizing. If E_b is from a typical nonscreenshot, the edge component of E_b does not have specific directionality. This means that the loss ratios of the edge component of both E_{b_h} and E_{b_v} are similar. As a result, Sim_{b_h} and Sim_{b_v} have similar values, and the *SR* is close to 1. However, if E_b is from a screenshot, the edge component of E_b has a horizontal directionality caused by the combing artifacts. Owing to the horizontal directionality of E_b, the loss ratio of the edge component of E_{b_v} is significantly larger than that of E_{b_h}. Consequently, Sim_{b_h} has a lower value than Sim_{b_v}, and the calculated *SR* is lower than 1. The *SR* is close to 0 when the directional noise inequality is large (i.e., the difference between the two interlaced fields is large). Figure 8 shows the distributions of the numerator and denominator of the *SR* of 1000 sample images (500 screenshots and nonscreenshots each). As can be seen from the figure, the screenshot distributions deviate from the slope of $\frac{\pi}{4}$ according to the magnitude of their combing artifacts, whereas the slope of the nonscreenshot distributions are close to $\frac{\pi}{4}$. The *SR* value is calculated from the whole edge block of a given image. Thus, the *SR* value is smaller when the amount of combing artifacts in the whole edge blocks is larger.

Figure 9 presents the *SR* histograms of the above sample screenshots and nonscreenshots. In Figure 9a, since the magnitude of the combing artifacts in the screenshot is changed on a case-by-case basis, the histograms of the screenshots do not follow a specific probability model. On the other hand, the histograms of nonscreenshots follow a Laplace model, as shown in Figure 9b. A random variable has a Laplace(μ, b) distribution if its probability density function is

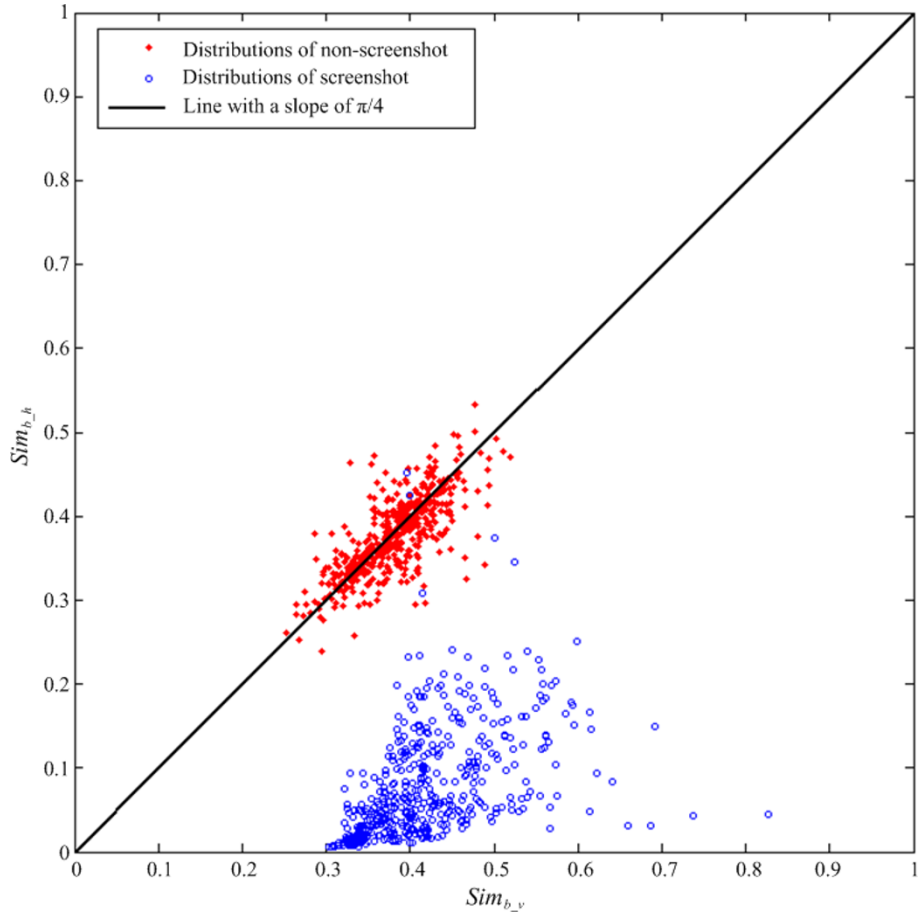

Figure 8 Similarity distributions of sample nonscreenshots and screenshots.

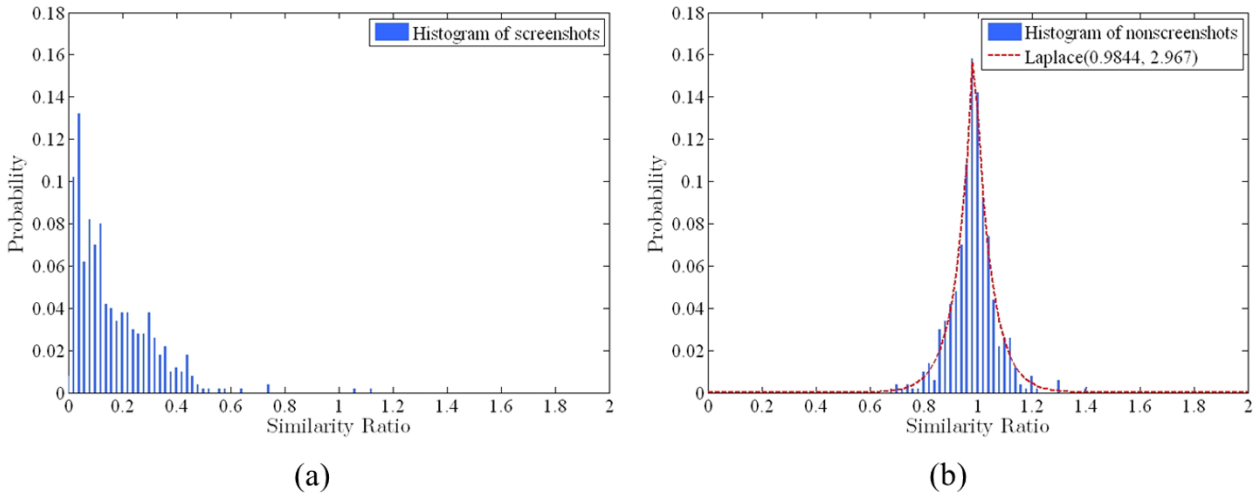

Figure 9 Similarity ratio histograms of screenshots and nonscreenshots. **(a)** *SR* histogram of screenshots, **(b)** *SR* histogram of nonscreenshots.

$$f(x|\mu,\ b) = \frac{1}{2b}e^{\left(-\frac{|x-\mu|}{b}\right)} = \frac{1}{2b}\begin{cases} e^{\left(-\frac{\mu-x}{b}\right)} & \text{if } x < \mu \\ e^{\left(-\frac{x-\mu}{b}\right)} & \text{if } x \geq \mu \end{cases} \quad (6)$$

Here, μ is a location parameter and $b > 0$ is a scale parameter, μ and b are calculated as follows.

$$\mu = E(x)$$
$$b = \sqrt{\frac{\mathrm{Var}_{(x)}}{2}} \quad (7)$$

where $E(x)$ and $\mathrm{Var}(x)$ are the expected value and the variance of the histogram, and x is a random variable for the *SR* values of nonscreenshots. In the histogram, $E(x)$ and $\mathrm{Var}(x)$ are calculated as 0.9844 and 17.6062, respectively. Therefore, the values of μ and b are 0.9844 and 2.967, respectively.

Since the *SR* histogram of screenshots does not follow a specific probability model, the probability model of the *SR* histogram of nonscreenshots, Laplace(0.9844, 2.967), can be used to identify the image source. For example, the screenshot identifier will flag the input image as a screenshot when the *SR* value of a given image is lower than 0.7523, 0.6167, or 0.5029, which corresponds to a false positive rate of less than 10^{-2}, 10^{-3}, or 10^{-4}, respectively.

3.3.2 Local directional inequality detection

The global directional inequality detection has certain advantages. In the case of non-screenshots, since there is little horizontal noise, which might be misinterpreted as combing artifacts, the misclassification rate is very small. Further, we can control the false positive rate because the *SR* values of nonscreenshots follow a Laplacian distribution.

However, some screenshots in which combing artifacts are shown as localized may be misclassified as nonscreenshots during the global directional inequality detection stage. Figure 10 shows two identification examples for the case in which the global directional inequality detection uses 10^{-3} as the false positive rate. For the screenshots that have local combing artifacts like Figure 10b, sometimes the global directional inequality detection misclassifies the image source. Therefore, we have to examine the existence of local combing artifacts in the images that were classified as nonscreenshots in the first stage. From now on, we refer to this method as local directional inequality detection. By this method, we can improve the identification accuracy of our screenshot identification scheme.

To find local combing artifacts in a given image, we pick candidate blocks that may contain combing artifacts, from the edge blocks. Since the distribution of GLCM follows a linear representation $y = x$ and the combing artifacts are horizontal noise, the GLCM result of a block that has combing artifacts satisfies the following condition:

$$(y_i = x_i + \varepsilon_i) \text{ and } (\mathrm{var}(\mathrm{GLCM}_V) > \mathrm{var}(\mathrm{GLCM}_H)) \quad (8)$$

where (x_i, y_i) is an element of GLCM, ε_i is an error term, and $\mathrm{var}(A)$ is the variance of A. When an edge block is given, we use the coefficient of determination (R^2) of simple linear regression to compare the variance of GLCM_H and GLCM_V of a given block [18]. R^2 of the data set A is calculated as follows:

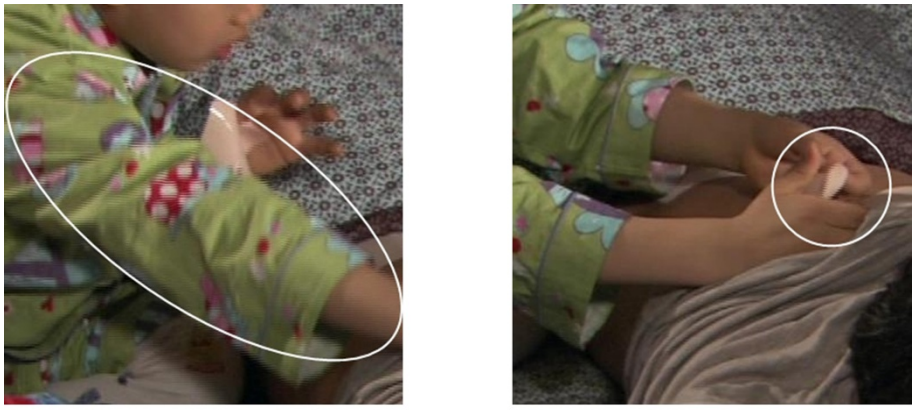

Figure 10 Identification results after applying global directional inequality detection when the screenshots that have different amount of combing artifacts are given. Combing artifacts are circled with white circles. **(a)** A screenshot that has large amount of combing artifacts is identified as a screenshot, **(b)** a screenshot that has small amount of combing artifacts is identified as a nonscreenshot.

$$R^2(A) = \frac{\text{Regression sum of squares } (SSR)}{\text{Total sum of squares}(SST)} = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \quad (9)$$
$$\text{where } \hat{y}_i = \textit{linear function}(x_i), \bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$$

Here, $y = \textit{linear function}\ (x)$ is $y = x$ and $(x_i, y_i) \in A$. $R^2\ (A)$ is in inverse proportion to the variance of the error between the linear function and the data of set A. As shown in Figure 11, larger values of R^2 tend to indicate that the data points are closer to the fitted regression line. We extract the candidate blocks from the edge blocks by comparing the two R^2 values, i.e., GLCM_H and GLCM_V, of each edge block. In mathematical terms, we have

$$E_b = \begin{cases} \text{Candidate}, & \text{if } R^2\left(\text{GLCM}_H^{E_b}\right) > R^2\left(\text{GLCM}_V^{E_b}\right) \\ \text{Non} - \text{candidate}, & \text{otherwise} \end{cases} \quad (10)$$

The discriminated candidate blocks are then used to calculate the block-based similarity ratio (BSR) to identify the existence of combing artifacts in each block.

$$BSR_b = \frac{Sim_{b\_h}}{Sim_{b\_v}} \quad (11)$$

Here, BSR_b is the BSR value of E_b. If the BSR_b of a certain block exceeds a preset threshold, then the block is classified as a local combing artifact block. We reclassify a given image, which was classified as a nonscreenshot in the first stage, as a screenshot when the percentage of the local combing artifact blocks is more than a chosen percentage of the total edge blocks. Here, we experimentally chose 8% as the percentage for identifying the local screenshots.

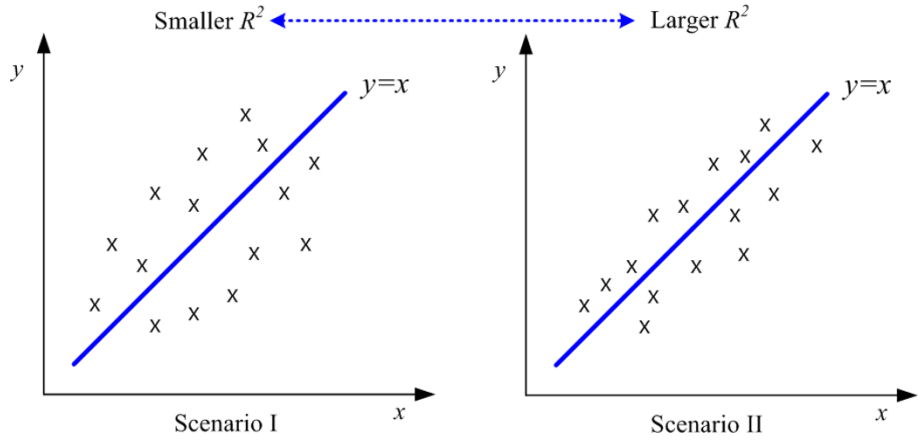

Figure 11 Scenarios I and II show the sample data points and its linear regression line $y = x$. The coefficient of determination R^2 is larger in Scenario II than in Scenario I.

4 Experimental results

This section presents the experimental results to evaluate the accuracy and efficiency of the proposed screenshot identification scheme. To do this, we gathered several nonscreenshots and screenshots. Table 2 shows the list of the source cameras of nonscreenshots and the source camcorders of screenshots used for the experiments. For the nonscreenshot set, 3,000 images of size 1,920 × 1,080 were used. For the screenshot set, first, we collected 10 TV programs encoded in NTSC format for various genres and 10 sets of camcorder recorded content. Here, the term "camcorder recorded content" refers to video content such as home videos or UGCs, which is recorded personally by amateur cameramen. This video content is one-hour long and has a resolution of 1,920 × 1,080. Then, we took 3,000 screenshots each from TV programs and camcorder recorded content. Thus, we got two screenshot sets consisting of 3,000 images: one from the camcorder recorded content, and the other from TV programs. From now on, the nonscreenshot set and screenshot set are denoted by *NS* and *SS*, respectively. In the set *SS*, there are two subsets *CS* and *PS*: *CS* is the set of screenshots taken from camcorder recorded content and *PS* is the set of screenshots taken from TV programs. In the experiments, the block size m was experimentally set to 32, and Th_1 and Th_2 of the edge decision formula D were experimentally selected as 10 and 0.1, respectively. *Undetermined* in the experimental results means that the input image is edgeless, so the proposed process cannot extract any edge blocks. Practically, since many screenshots are taken from meaningful scenes of video content, most screenshots that can be found on the Internet have the edge component. This means that the probability of a test image being *Undetermined* is negligible.

4.1 Comparative test with and without applying local directional inequality detection

To verify the improved performance when employing the local directional inequality detection stage, we carried out a comparative test with and without employing the local directional inequality detection. In the test, the test image sets consisted of 512 × 512 *NS*, *PS*, and *CS*, and we compressed the given image sets using JPEG and MPEG-4. Table 3 summarizes the image source identification results for the comparative test when we used a false positive rate of 10^{-3} as a threshold. As shown in Table 3 both identification results of with and without employing the local directional inequality detection stage were similar in the cases of *NS* and *CS*. However, the identification accuracy that employed local directional inequality detection stage was much lower than the identification accuracy that did not employ the local directional inequality detection stage in the case of *PS*. It is because combing artifacts tend to be localized, and the magnitude of combing artifacts fluctuates significantly in the case of PS. Thus, the misclassification rate became unavoidably high when we did not employ the local directional inequality detection stage. From now on, all experimental results were obtained after employing both global and local directional inequality detection.

Table 3 Identification results of *NS*, *PS*, and *CS* with and without employing the local directional inequality detection.

	Out					
	Without employing			With employing		
	NS	*SS*	*Undetermined*	*NS*	*SS*	*Undetermined*
In						
NS	98.45	0.06	1.49	97.98	0.48	1.54
PS	30.51	69.12	0.37	15.69	83.91	0.40
CS	1.18	97.73	1.09	0.39	98.53	1.10

(unit: %)

4.2 Format conversion

In order to evaluate the performance, we compared the proposed scheme with our previous study [14] under the three most widely used image and video formats, i.e., JPEG, BMP, and TIFF for images and MPEG-2, MPEG-4, and H.264 for videos. In the NSs, the center area of size 512 × 512 was cropped from each image and saved in the JPEG, BMP, and TIFF formats. In total, we got 3,000 JPEG, 3,000 BMP, and 3,000 TIFF images of *NS*s. To make the format-converted *SS*, TV programs and camcorder recorded content were first converted to MPEG-2, MPEG-4, and H.264. Then, we took 3,000 screenshots for each video format and cropped them in the same manner as for the *NS*. These *SS*s were saved to JPEG, BMP, and TIFF. Since there are two sources of video content (TV programs and camcorder recorded content),

Table 2 Sources of nonscreenshot set and screenshot set used for the experiments

Source of nonscreenshots	# of used	Source of screenshots	# of used images
Nikon D90	600	Sony HDR-CX550	800
Olympus E420	600	Sony HDR-FX1	800
Canon 500D	600	Sony XR520	700
Sony α380	600	Samsung HMX-H205	700
Dresden image database [19]	600	TV programs	3000

a total of 18 screenshot sets (= 3(# video formats) × 3(# image formats) × 2(# sources of video content)) were made. In this experiment, the compression ratio of JPEG was 90%, and BMP and TIFF were encoded in 24 bits. We compressed the MPEG-2 and MPEG-4 format video clips at 5,000 and 3,000 kbps, respectively, and the compression ratio of H.264 was 90%. Table 4 summarizes the experimental results for the various formats, these results were obtained using the abovementioned sets for the threshold that was set to have a false positive rate of 10^{-3}.

As shown in Table 4 the overall identification results of the proposed scheme were much better than those of the previous study. Since our previous method uses the SVM classifier, there is no *Undetermined* part. This means that they have to select the image source unconditionally. Thus, the false positive of the previous scheme is significantly higher than that of the proposed scheme.

At the bottom of Table 4 the screenshot identification results from two different video sources are shown. As seen in the results, the identification accuracy is not influenced by a specific image or video format in a certain video source. This means that the directional noise inequality of the given screenshot is not affected by a specific image and video format. In other words, combing artifacts are not easily removed by image or video format conversion. However, combing artifacts are affected by the video source. While the misidentification rate of *PS* is around 15%, the misidentification rate of *CS* is only around 0.5%. This difference is due to the characteristics of the source of the content. Generally, this difference arises from the purpose for which the content is created and the recording skills of the cameraman. Figure 12 shows the *SR* histograms of *PS* and *CS*. In the case of TV programs, most of the content was recorded to be shown to the audiences and the scenes were recorded by professional cameramen. Further, most camcorders for recording the content are fixed to prevent shaking, and the recorded content is edited to provide a comfortable viewing experience for the viewers. Thus, the movement of the object in a scene is relatively slow and localized. Further, only objects, rather than the whole background, move frequently. Because of these characteristics of TV programs, combing artifacts are localized and the magnitude of combing artifacts fluctuates significantly. Consequently, the *SR* distribution of screenshots from TV programs is randomly spread from 0 to 1. On the other hand, most scenes of camcorder recorded content are more dynamic and the size of motion is also more globalized than that of TV programs because most of the content is recorded by amateurs. Since combing artifacts reflect these tendencies, the *SR* distribution in Figure 12b is localized around the low *SR* values.

Table 4 Identification results of *NS*, *PS*, and *CS* under various image and video formats.

	Out				
	Proposed scheme			Lee's scheme [14]	
	NS	*SS*	*Undetermined*	*NS*	*SS*
In					
NS_{JPEG}	97.83	0.50	1.67	94.93	5.07
NS_{BMP}	97.90	0.60	1.50	94.91	5.09
NS_{TIFF}	98.20	0.33	1.47	94.93	5.07
$PS_{MPEG\text{-}2/JPEG}$	16.60	83.00	0.40	25.91	74.09
$PS_{MPEG\text{-}2/BMP}$	15.80	83.83	0.37	25.81	74.19
$PS_{MPEG\text{-}2/TIFF}$	15.67	83.83	0.50	25.90	74.10
$PS_{MPEG\text{-}4/JPEG}$	14.50	85.23	0.27	23.33	76.67
$PS_{MPEG\text{-}4/BMP}$	14.10	85.60	0.30	23.14	76.86
$PS_{MPEG\text{-}4/TIFF}$	14.50	85.23	0.27	23.33	76.67
$PS_{H.264/JPEG}$	16.97	82.47	0.57	25.09	74.91
$PS_{H.264/BMP}$	16.43	83.00	0.57	23.15	76.86
$PS_{H.264/TIFF}$	16.67	83.00	0.33	24.33	75.67
$CS_{MPEG\text{-}2/JPEG}$	0.07	98.43	1.50	3.61	96.39
$CS_{MPEG\text{-}2/BMP}$	0.03	98.47	1.50	3.61	96.39
$CS_{MPEG\text{-}2/TIFF}$	0.07	98.43	1.50	3.51	96.49
$CS_{MPEG\text{-}4/JPEG}$	0.53	98.53	0.93	5.16	94.84
$CS_{MPEG\text{-}4/BMP}$	0.53	98.57	0.90	4.60	95.40
$CS_{MPEG\text{-}4/TIFF}$	0.67	98.43	0.90	5.01	94.99
$CS_{H.264/JPEG}$	0.43	98.67	0.90	4.36	95.64
$CS_{H.264/BMP}$	0.43	98.67	0.90	4.89	95.11
$CS_{H.264/TIFF}$	0.53	98.57	0.90	4.36	95.64

(A_{α}: α image-formatted A, $B_{\alpha/\beta}$: B with α video format and compressed by β image format, unit: %)

4.3 Compression

Since most of the images and videos that can be found on the Internet are compressed, the proposed method has to be robust against the compression of frequently used image and video formats. To measure the robustness of the proposed technique under image and video compression, we compressed the given image sets using JPEG and MPEG-4, which are the most widely used image and video formats, respectively, and we measured the directional noise inequalities. Here, the test image sets consisted of 512 × 512 *NS* and *CS*.

Firstly, to gauge the effect of image compression, we changed only the JPEG compression ratio of *NS* and *CS*. Table 5 shows the confusion matrices of various JPEG compression ratios obtained when we used a false positive rate of 10^{-3} as a threshold. In the table, the cells that have an identification accuracy larger than 95% are colored dark gray, the other cells are colored light gray. The identification results show that combing artifacts have robustness under JPEG compression. In particular, the identification accuracy is similar value when the JPEG compression ratio is greater than 50%. The identification accuracy of screenshots is low when the JPEG

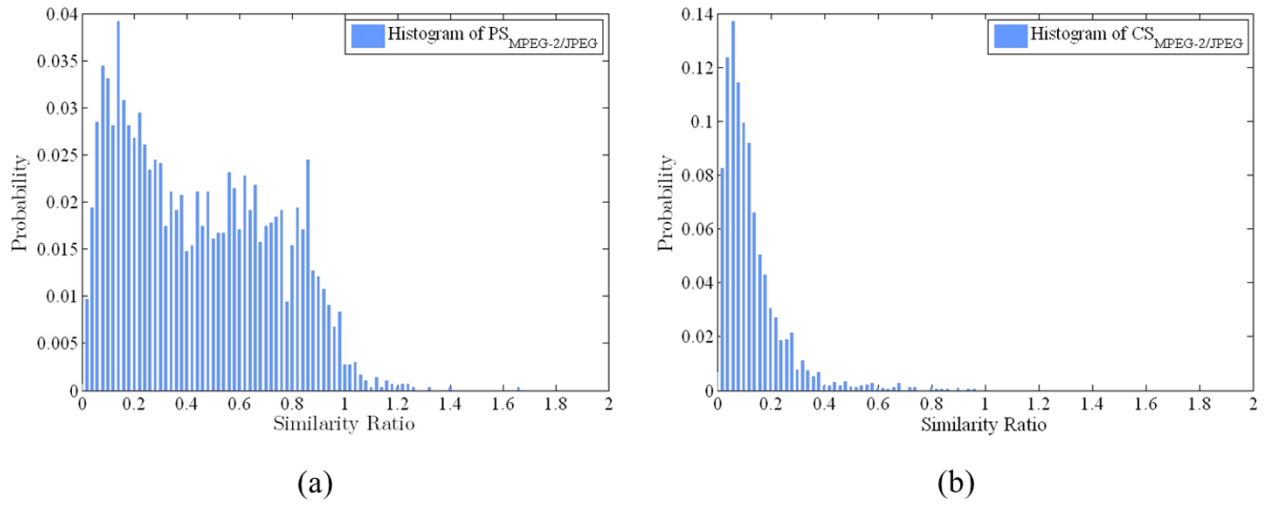

Figure 12 *SR* histogram of screenshot sets from two different video source. **(a)** *SR histogram of* $PS_{MPEG\text{-}2/JPEG}$, **(b)** *SR* histogram of $CS_{MPEG\text{-}2/JPEG}$.

compression ratio is low, whereas the identification accuracy of nonscreenshots is still high when the JPEG compression ratio is low. The reason is that the edge components of the textured area are weakened owing to strong JPEG compression, thus, the difference between the vertical and horizontal similarity values becomes smaller than that before JPEG compression. Therefore, some test images were identified as nonscreenshots because severe JPEG compression decreased the horizontal noise including combing artifacts. However, strong JPEG compression harms the image quality, so people are usually unwilling to perform JPEG compression with a compression ratio of less than 50%.

In the case of video compression, we compressed the camcorder recorded content using the MPEG-4 encoding technique. We took 3,000 screenshots with a size of 512 × 512 using uncompressed JPEG to eliminate the JPEG compression effects. The identification results obtained using a false positive rate of 10^{-3} as the threshold are shown in Table 6. Both the identification accuracy and the rate of *Undetermined* show that combing artifacts are slightly influenced by the MPEG-4 compression. However, the

Table 5 Confusion matrices of various JPEG compression.

		out				out				out	
	NS	*SS*	*UN*		*NS*	*SS*	*UN*		*NS*	*SS*	*UN*
in				**in**				**in**			
NS	**97.80**	0.50	1.70	*NS*	**97.53**	0.70	1.77	*NS*	**97.33**	0.77	1.90
CS	0.07	**98.43**	1.50	*CS*	0.07	**98.53**	1.40	*CS*	0.27	**97.50**	2.23
	JPEG quality: 100				JPEG quality: 90				JPEG quality: 80		
		out				**out**				**out**	
	NS	*SS*	*UN*		*NS*	*SS*	*UN*		*NS*	*SS*	*UN*
in				**in**				**in**			
NS	**97.73**	0.60	1.67	*NS*	**97.90**	0.87	1.23	*NS*	**98.13**	0.97	0.90
CS	0.03	**98.77**	1.20	*CS*	0.13	**98.93**	0.93	*CS*	0.17	**98.83**	1.00
	JPEG quality: 70				JPEG quality: 60				JPEG quality: 50		
		out				**out**				**out**	
	NS	*SS*	*UN*		*NS*	*SS*	*UN*		*NS*	*SS*	*UN*
in				**in**				**in**			
NS	**98.90**	0.77	0.33	*NS*	**99.17**	0.50	0.33	*NS*	**99.37**	0.10	0.53
CS	1.30	**95.43**	3.10	*CS*	1.17	**95.73**	3.10	*CS*	2.43	**93.93**	3.63
	JPEG quality: 40				JPEG quality: 30				JPEG quality: 20		

(*UN: Undetermined*, unit: %, Dark gray cell: identification accuracy is larger than 95%, Light gray cell: identification accuracy is smaller than or equal to 95%)

Table 6 Identification results under various MPEG-4 compression.

	Out		
	NS	*SS*	*Undetermined*
In			
CS_{30}	0.10	96.20	3.70
CS_{40}	0.23	95.77	4.00
CS_{50}	0.40	96.77	2.83
CS_{60}	0.07	97.90	2.03
CS_{70}	0.03	97.83	2.13
CS_{80}	0.10	98.30	1.60
CS_{90}	0.03	98.00	1.97
CS_{100}	0.00	99.10	0.90

(CS_{α}: *CS* with α MPEG-4 quality, unit: %)

screenshot identification results are higher than 96% under severe MPEG-4 compression such as 30% compression ratio. This results shows that combing artifacts are not easily removed by MPEG-4 compression.

The state-of-the-art image and video formats can express much more information of original content compared with JPEG and MPEG-4 under the same compression ratio. This means that the combing artifacts of a screenshot may remain after the state-of-the-art image and video compression techniques have been implemented.

4.4 Cropping

Most screenshots include whole frames of video content, but the screenshots may have only parts of a video frame. From now on, we refer to the screenshots that include parts of a video frame as *partial screenshots*. To measure the efficiency of the proposed method for *partial screenshots*, we tested five *NSs* and *CSs* with different cropping portions. Apart from the cropping portion, we controlled the other variables such as image and video formats, crop position, and video source. The image format was set to uncompressed JPEG, and the video format was MPEG-2, whose bit rate is 5,000 kbps. Further, we cropped the center area of the given image, and we used the camcorder recorded content as the video source. The cropping portions for this test were selected as 1/2, 1/8, 1/32, 1/128, and 1/512. When the original size of a screenshot is 1920 × 1080, the size of a *partial screenshot* with 1/512 cropping portion is about 64 × 64.

Figure 13 shows the ROC curve of five *NSs* and *CSs* with different cropping portions. As shown in Figure 13, the overall identification accuracy is satisfactory under any cropping portion of *partial screenshots*. The enlarged ROC curve shows that the degree of cropping portion influences the performance of the screenshot detector distinctly. Further, Table 7 shows that the number of *Undetermined* images is increased when the cropping portion is smaller. However, since most people take a screenshot of meaningful scenes of video content, the screenshot would have enough edge information even if it is a *partial screenshot*. Thus, the actual rate of *Undetermined* may be negligible. At this point, the proposed screenshot identifier can operate well under the *partial screenshot*.

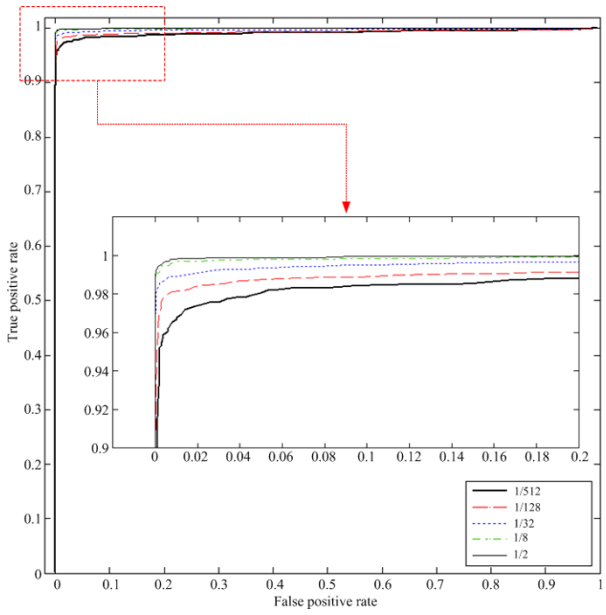

Figure 13 ROC curves of various cropping portions.

5 Conclusion

An interlaced frame is generated by weaving the even and odd fields. In this process, the horizontal jagged noise called the combing artifact is produced because of the temporal differences between the even and odd fields. Combing artifacts are one of the representative characteristics of interlaced videos, and hence, screenshots of interlaced video content inherently have combing artifacts. In this study, we present a scheme for screenshot identification using the properties of combing artifacts. Since combing artifacts are easily found around the edge areas, we extract the edge areas from the input image using the GLCM. Then, since combing artifacts are horizontal noise, we use this property to

Table 7 Rate of *Undetermined* images under the given cropping portion.

	Cropping portion				
	1/2	**1/8**	**1/32**	**1/128**	**1/512**
Source					
NS	0.97	1.87	3.74	8.58	19.50
CS	0.75	1.67	3.67	8.60	19.06

(unit: %)

define the *SR* and *BSR*, the global and local directional noise inequality identifying measure, using the LH and HL subbands of the DWT in the extracted edge areas. The proposed two-stage directional in-equality detection method identifies the source of test images stably in various environments: various image or video formats, cropping portion, and image or video compression.

The proposed scheme shows good performance, though there are a few drawbacks to resolve. The two-stage directional inequality detection method does not apply to screenshots of motionless video content. Further, if the screenshot does not have any edge component, we cannot apply the proposed scheme. To solve these problems, not only combing artifacts but also other inherent characteristics of video content should be used to design the screenshot identifying measure. The above considerations will provide the direction for future studies.

Acknowledgements
This research was supported by WCU (World Class University) program (Project No: R31-30007) and NRL (National Research Lab) program (No. R0A-2007-000-20023-0) under the National Research Foundation of Korea and funded by the Ministry of Education, Science and Technology of Korea, and also was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the CYBER SECURITY RESEARCH CENTER supervised by the NIPA (National IT Industry Promotion Agency), NIPA-C1000-1101-0001.

Author details
[1]Department of Computer Science, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea [2]Information Technology R&D Center, SK Telecom, 11 Euljiro 2-ga, Jung-gu, Seoul, Republic of Korea [3]Department of Computer Software Engineering, Kumoh National Institute of Technology, Sanho-ro 77, Gumi, Gyeongbuk, Republic of Korea [4]Department of Computer Science and Division of Web Science and Technology, Korea Advanced Institute of Science and Technology, 291 Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea

Competing interests
The authors declare that they have no competing interests.

Received: 2 September 2011 Accepted: 2 May 2012
Published: 2 May 2012

References

1. **Copyright issues of screenshots.** [http://www.trustedcomputinggroup.org/].
2. Lyu S, Farid H: **How realistic is photorealistic?** *IEEE Trans Signal Process* 2005, **53(2)**:845-850.
3. Chen W, Shi YQ, Xuan G: **Identifying computer graphics using hsv color model and statistical moments of characteristic functions.** *Proc IEEE Int Conf Multimedia and Expo* Beijing, China; 2007, 1123-1126.
4. Ng TT, Chang SF, Hsu J, Xie L: **Physics-motivated features for distinguishing photographic images and computer graphics.** *Proc ACM Multimedia* Singapore; 2005, 239-248.
5. Ng TT, Chang SF: **An online system for classifying computer graphics images from natural photographs.** In *Proc SPIE Electronic Imaging. Volume 6072.* San jose, CA; 2006:397-405.
6. Wang Y, Moulin P: **On discrimination between photorealistic and photographic image.** *Proc IEEE Int Conf Acustics, Speech, and Signal Processing* Toulouse, France; 2006, 161-164.
7. Wu J, Kamath MV, Poehlman S: **Detecting differences between photographs and computer generated images.** *Proc IASTED Int Conf Signal Processing, Pattern Recognition, and Applications* Innsbruck, Austria; 2006, 268-273.
8. Chen W, Shi YQ, Xuan G, Su W: **Computer graphics identification using genetic algorithm.** *Proc Int Conf Pattern Recognition* Tampa, FL; 2008, 1-4.
9. Li W, Zhang T, Zheng E, Ping X: **Identifying photorealistic computer graphics using second-order difference statistics.** *Proc IEEE Int Conf Fuzzy Sistems and Knowledge Discovery* Yantai, China; 2010, 2316-2319.
10. Sutthiwan P, Cai X, Shi YQ, Zhang H: **Computer graphics classification based on markov process model and boosting feature selection technique.** *Proc IEEE Int Conf Image Processing* Yantai, China; 2009, 2913-2916.
11. Yu H, Ng T-T, Sun Q: **Recaptured photo detection using specularity distribution.** *Proc IEEE Int Conf Image Processing* San Diego, CA; 2008, 3140-3143.
12. Cao H, Kot AC: **Identification of recaptured photographs on lcd screens.** *Proc IEEE Int Conf Acustics, Speech, and Signal Processing* Dallas, TX; 2010, 161-164.
13. Gao X, Ng T-T, Qiu B: **Single-view recaptured image detection based on physics-based features.** *Proc IEEE Int Conf Multimedia and Expo* Suntec City, Singapore; 2010, 1469-1474.
14. Lee J-W, Lee M-J, Oh T-W, Ryu S-J, Lee H-K: **Screenshot identification using combing artifact from interlaced video.** *Proc ACM Multimedia and Security* Rome, Italy; 2010, 49-54.
15. Wang W, Farid H: **Exposing digital forgeries in interlaced and de-interlaced video.** *IEEE Trans Inf Forensics Security* 2007, **2(3)**:438-449.
16. Haralick RM, Shanmugam K, Dinstein I: **Textural features for image classification.** *IEEE Trans Syst Man Cybernet* 1973, **3(6)**:610-621.
17. Daubechies I: *Ten Lectures on Wavelets* SIAM, Philadelphia; 1992.
18. Hayter A: *Probability and Statistics for Engineers and Scientists* Thomson, Belmont, CA; 2007.
19. Gloe T, Bohme R: **The 'Dresden Image Database' for benchmarking digital image forensics.** In *Proc ACM Symposium on Applied Computing. Volume 2.* Sierre, Switzerland; 2010:1584-1590.

doi:10.1186/1687-5281-2012-7
Cite this article as: Lee *et al.*: Screenshot identification by analysis of directional inequality of interlaced video. *EURASIP Journal on Image and Video Processing* 2012 **2012**:7.