2

The Copyright Protection System for Android Platform

Yueh-Hong Chen[1] and Hsiang-Cheh Huang[2]

[1] Department of Computer Science and Information Engineering,
Far East University,
Tainan 744, Taiwan, R.O.C.
yuehhong@gmail.com
[2] Department of Electrical Engineering,
National University of Kaohsiung,
Kaohsiung 811, Taiwan, R.O.C.
huang.hc@gmail.com
https://sites.google.com/site/hch888dr/

Summary. In recent years, Camera Smartphone has become a popular consumer electronics product. Young people like to use it to record their daily lives; moreover, they will share these photos and information to others. But the photos may be used without consent after they are uploaded to Internet. To avoid this problem, one can embed visible and invisible watermarks into images. However, an additional process for embedding watermarks should be performed before an image is uploaded. When the number of photos becomes large, the process for embedding watermarks will bother the user. Thus, in this chapter, we propose a copyright embedding system for Android platform. Using this system, pre-specified copyright information is automatically embedded into pictures with digital watermark technology when these pictures are taken. In addition, original images (i.e., images without watermarks) can be preserved selectively. Proposed system has following features: (1) computational complexity of watermark embedding process is possibly reduced for handheld mobile system; (2) the watermark can be extracted without the use of the original image; (3) the watermark embedded into an image would not be removed by commonly used image processing operations; (4) embedding copyright information, resizing the images, and uploading images to Internet are automatically performed without manual intervention. Therefore, this system is very suitable for the protection of the photographs taken by Android phones to prevent piratical behaviors.

2.1 Introduction

In recent years, the Camera Smartphone has become a popular consumer electronics product. Young people like to use that to record their daily lives; moreover, they will share these photos and information to others. However, the photos may be used without consent after they are uploaded to Blogs. Since digital images can easily be to redistributed without agreement, related researches on digital right protection

J.-S. Pan et al. (Eds.): Recent Advances in Information Hiding and Applications, ISRL 40, pp. 19–32.
springerlink.com

have attracted much attention in recent years. Among all schemes, digital watermarkin [1]–[7] can still protect digital images when they are displayed. Thus, it can be used as one of the approaches to preventing image piracy. Digital watermarking is a process to embed some information (i.e., watermarks) to image data. The watermarked image can still be displayed, and the embedded information is used for owner identification in the future. In practice, the watermark embedding algorithm can be designed to resist re-encoding, compression, D/A converting and image processing operations. Therefore, digital watermarking has been considered as the underlying technology in several digital rights management (DRM) applications [2]. For instance, In copy prevention, digital watermarking may be used to embed license information so that hardware devices and software can detect illegal use of digital contents. In copyright protection applications, the watermark may be used to identify the copyright holder and ensure proper payment of royalties. Moreover, there are a number of other applications for which watermarking has been used or suggested. These include broadcast monitoring, transaction tracking, authentication, copy control, and device control [2].

Although one can embed visible and invisible watermarks into pictures to prevent image piracy, an additional process for embedding watermarks should be performed before a picture is uploaded while necessary. When the number of photos becomes large, the process for the embedding of watermarks will bother the user due to the routine process. Thus, in this chapter, we propose a copyright embedding system for Android platform. Using this system, pre-specified copyright information is embedded into pictures with digital watermark technology when these pictures are taken. In addition, original images (i.e., images without watermarks) can be preserved selectively. Since computational complexity of watermark embedding procedure is possibly reduced for handheld mobile system, processes including embedding copyright information and resizing the photos along with uploading these photos to Internet can be automatically performed. Thus, this system is very suitable for the protection of the photographs taken by Android phones to prevent piratical behaviors.

2.2 Related Works

Android is a software stack including an operating system, middleware and key applications. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. Based on Linux kernel, Android provided a set of libraries, Android runtime and an application framework. For application developers, the Android SDK provides the tools and APIs necessary to develop applications on the Android platform using Java programming language.

Watermarking techniques can be briefly classified into *additive*, *multiplicative*, *quantization-based*, and *relationship-based* schemes. They are briefly described as follows.

1. In additive schemes, a very weak W is added into original signal x, as shown in Eq. (2.1):
$$Y = X + \alpha W, \tag{2.1}$$
where X is the original signal, Y is the watermarked signal and α is a constant, referred to as *watermark strength*.
2. In multiplicative schemes, samples of the original data are multiplied by an independent signal $(1 + \alpha W)$. Precisely, multiplicative schemes can be described by Eq. (2.2):
$$Y = X \times (1 + \alpha W). \tag{2.2}$$
3. In quantization based watermarking schemes, X is modified such that the quantization indices imply a watermark with a certain quantization step q. For example, a binary watermark W can be embed into the signal X with following Eq. (2.3):
$$\begin{cases} \lfloor \frac{Y}{q} + 0.5 \rfloor \bmod 2 = 0, & \text{if } W = 0; \\ \lfloor \frac{Y}{q} + 0.5 \rfloor \bmod 2 = 1, & \text{if } W = 1. \end{cases} \tag{2.3}$$
In this example, signal X is modified into Y such that its quantization index is an even number to imply a binary value '0', and vice versa.
4. The basic idea of relationship-based watermarking is to use two pixel values or transform domain coefficients in an image to represent each bit of a binary watermark. If the first value is larger than the second, then an '1' is encoded; otherwise, a '0' is encoded. Hsu and Wu [8] proposed an approach using middle frequency coefficients chosen from one or more 8×8 DCT blocks to embed watermarks. Quantization operation is taken into account in this approach so that watermarks can survive the JPEG lossy compression. In [9], six coefficients are selected from a DCT block and then the first coefficient is exchanged with the largest or smallest coefficient among them according to to watermark bit value. A closely related approach was proposed in [10]. These approaches may also achieve good robustness.

2.3 Automatic Photograph Publishing System

In this section, we describe how to watermark a photograph with the proposed system, perform the resizing of images, and uploads the output image to Internet. The watermarking approach adopted in the system is also introduced briefly.

2.3.1 Procedures for Automatic Photograph Publishing

The automatic photograph publishing process proposed in this chapter involves three major tasks: (1) watermarking embedding, (2) image resizing, and (3) image uploading. These three tasks are integrated appropriately into the automatic photograph publishing process to reduce computational complexity. A flow diagram of the proposed automatic photograph publishing process is shown in Fig. 2.1. The

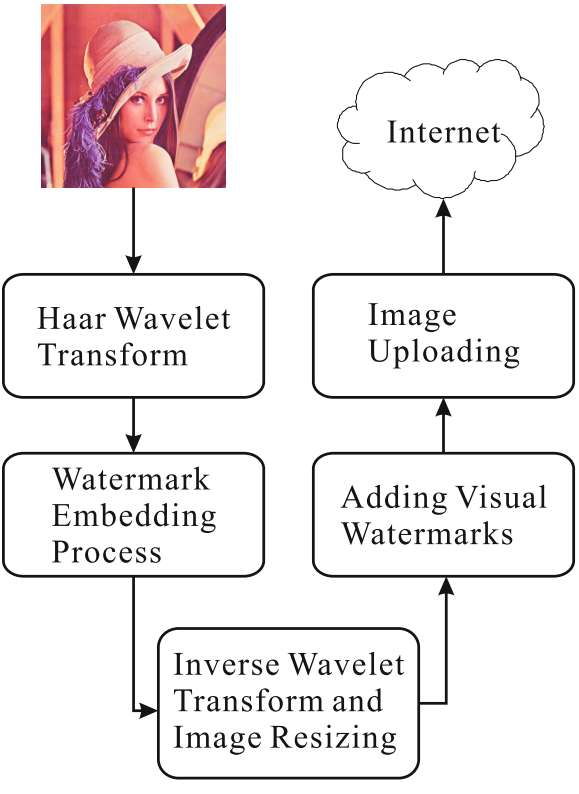

Fig. 2.1. Flow diagram of the proposed Automatic photograph publishing process

five steps included in the automatic photograph publishing process are described as follows.

1. Transform the photograph into Haar wavelet domain.
2. Embed the pre-specified watermark into the photograph.
3. Transform the photograph to pixel domain according a pre-specified resize factor.
4. Add a visible watermark (i.e., copyright information) to the photograph.
5. Upload the watermarked photograph to Internet.

Since Haar wavelet with some modifications may be performed without floating-point operations, it is very suitable for constrained devices such as handheld sets. Moreover, Haar wavelet transform is an orthonormal transform, so the visual quality of the watermarked image can be evaluated directly in the transform domain. In the second step, the watermarking scheme proposed in [11] is adopted to embed the watermark into photographs. This watermarking scheme will be introduced in follow subsections. After the watermark is embedded, the watermarked image may be resized so that it is applicable to the Internet environment. To resize the

watermarked image, we can use only the low and middle frequency coefficients in Haar wavelet domain to perform inverse Haar wavelet transform. Thus, if $HH1$, $HL1$ and $LH1$ coefficients are discarded, the image is resized to half the original size. With the same manner, discarding $HH1$, $HL1$, $LH1$, $HH2$, $HL2$ and $LH2$ will result in a quarter-size watermarked image. Therefore, image resizing and watermark embedding can be applied simultaneously. If the resize factor specified by the user is not a power of 2, the resizing operation may not be performed by discarding some wavelet coefficients. In the circumstances, the method proposed in [12] is used to resize the image. Finally, we add a visible watermark with pixel domain image processing techniques and upload the watermarked image to a pre-specified web site.

2.3.2 The Watermark Embedding Process

To embed a watermark, an image is firstly transformed into Haar wavelet domain. For each bit of the watermark, a number of coefficients in pre-specified subband (e.g., $LH3$, $HL3$ or $HH3$) are then randomly chosen and modified. Finally, inverse wavelet transform is applied to obtain the watermarked image.

When one bit of the watermark is to be embedded, an user-specified number of coefficients are chosen randomly. These coefficients are then changed such that the first coefficient, in the order of being chosen, becomes the largest one if an '1' is to be embedded. If a '0' is to be embedded, the coefficients should be modified such that the first coefficient becomes the smallest one. Suppose c_i, $i = 1, \cdots, n$, are the chosen coefficients, n is the number of chosen coefficients, $W = \{w_i \mid w_i \in \{1,0\}, 1 \leq i \leq L\}$ is the watermark to be embedded, and L is the length of the watermark W. Precisely, after modification step, the relationship behind the coefficients is as Eq. (2.4)

$$\begin{cases} c_1^{'} \geq \max\left(c_2^{'}, c_3^{'}, \cdots, c_n^{'}\right) + \delta, & \text{if } w_i = 1, \\ c_1^{'} \leq \min\left(c_2^{'}, c_3^{'}, \cdots, c_n^{'}\right) + \delta, & \text{if } w_i = 0, \end{cases} \tag{2.4}$$

where $c_i^{'}$, $i = 1, \cdots, n$ are the modified coefficients, w_i is a particular bit of the watermark code, and δ, $\delta \geq 0$, is the strength parameter specifying the difference between the first coefficient and the largest (smallest) one among remaining coefficients. Intuitively, the larger the value of δ, the more robust the watermark. However, the perceptual fidelity of the watermarked image will decrease when a larger δ is adopted. For different applications, the value of δ should be specified by the user.

To clarify the description, a simple example of implying a watermark bit with coefficients is given. Suppose an '1' is to be embedded and five coefficients, -5, 112, -1, 107 as well as 13, are chosen randomly. A straightforward manner is to increase the value of the first coefficient, -5, to a value equal to or larger than 112, and other coefficients are left unchanged. By this manner, the first coefficient, -5, should be increase to $112 + \delta$ to embed an '1'.

2.3.3 Optimal Method for Embedding Binary Value

In order to obtain enhanced result in the image quality, more than one coefficient should be considered at the same time. To embed an '1', if the first coefficient c_1 is increased to $x+\delta$, all coefficients larger than x should be decreased to x to fit the rule shown in Eq. (2.4). Therefore, it is possible to find the optimal value of x such that the watermarked image have the best quality according to an appropriate quality metric.

In this chapter, we adopt PSNR as the image quality metric due to its simplicity. If the mean square error (MSE) of the modified coefficients is minimized, the PSNR value is maximized simultaneously. Suppose a bit of '1' is to be embedded into n coefficients. If c_1 is increased to $x+\delta$ and all coefficients larger than x are decreased to x, the square error (SE) value can be calculated as Eq. (2.5):

$$SE(x) = ((x+\delta)-c_1)^2 + \sum_{c_i>x} (c_i - x)^2. \tag{2.5}$$

Then the minimum of $SE(x)$ can be obtained by finding out the value of x where the first derivative of $SE(x)$ is equal to 0. The first derivative of $SE(x)$ is shown in Eq. (2.6), and the optimal value of x is shown in Eq. (2.7).

$$\frac{d}{dx}SE(x) = 2\times(x+\delta-c_1) + 2\times\sum_{c_i>x}(x-c_i), \tag{2.6}$$

$$x = \frac{\left(\sum\limits_{c_i>x} c_i\right) + c_1 - \delta}{k+1}, \quad i = 1,\cdots,n, \tag{2.7}$$

where k is the number of coefficients larger than c_1. In Eq. (2.7), it is assumed that only k largest coefficients and c_1 be modified. Therefore, the value of x should be larger than the $(k+1)$-th largest coefficient but smaller than k-th largest coefficient. The algorithm to find the optimal value x is as follow:

Obtain $d_1, d_2, \cdots, d_n$ *by sorting* $c_1, c_2, \cdots, c_n$
  such that $d_1 \geq d_2 \geq \cdots \geq d_n$
Suppose c_1 *is the* $(k+1)$*-th largest value*
If $(k+1) = 1$
  $x_{opt} = d_2$, *Stop*
End If
For $i = 1$ *to* k
  $x = \frac{\left(\sum_{j=1}^{i} d_j\right) + d_{k+1} - \delta}{i+1}$
  If $d_{i+1} < x \leq d_i$
   $x_{opt} = x$, *Stop*
  End If
End For
$x_{opt} = c_1$, *Stop*

After completing the algorithm, the optimal value of x can be found. c_1 can then be modified to $x+\delta$, and all coefficients larger than x be modified to x to embed a bit of '1'. A similar algorithm to find the optimal value to embed a '0' is as follow:

Obtain $d_1, d_2, \cdots, d_n$ *by sorting* $c_1, c_2, \cdots, c_n$
  such that $d_1 \le d_2 \le \cdots \le d_n$
Suppose c_1 *is the* $(k+1)$*-th smallest value*
If $(k+1) = 1$
  $x_{opt} = d_2$, *Stop*
For $i = 1$ *to* k
  $x = \frac{\left(\sum_{j=1}^{i} d_j\right) + d_{k+1} + \delta}{i+1}$
  If $d_{i+1} > x \ge d_i$
   $x_{opt} = x$, *Stop*
  End If
End For
$x_{opt} = c_1$, *Stop*

Finally, c_1 is decreased to $x-\delta$, and all coefficients smaller than x be increased to x to embed a bit of '0'.

Continuing the example in previous subsection, if $\delta = 0$, the $SE(x)$ value is:

$$SE(x) = \begin{cases} (x-112)^2 + (x+5)^2 \\ \quad \text{if } 107 \le x < 112; \\ (x-112)^2 + (x-107)^2 + (x+5)^2 \\ \quad \text{if } -1 \le x < 107; \\ (x-112)^2 + (x-107)^2 + (x+1)^2 + (x+5)^2 \\ \quad \text{if } -5 \le x < -1. \end{cases} \tag{2.8}$$

By applying the proposed algorithm, the optimal value of x, about 71.3, can be obtained. The curve of $SE(x)$ is shown in Fig. 2.2. It is clear that SE(x) is the minimum when $x = 71.3$.

2.3.4 The Watermark Extraction Process

The simplest extracting method is to pick up the same coefficients and determine if the first coefficient is largest or smallest. However, the watermarked image may be distorted due to some image-processing operations, and the first coefficient is possibly no longer the largest or the smallest one. Hence, the purposed extracting method is to compare the first coefficient with the largest and smallest ones among remaining coefficients. If the value of the first coefficient is closer to the largest one among remaining coefficients, an '1' will be extracted; otherwise, a '0' will be extracted. This method can be described as Eq. (2.9):

$$w_i' = \begin{cases} 1, & \text{if } c_1'' \ge \frac{1}{2}\left(c_{\max}'' + c_{\min}''\right); \\ 0, & \text{otherwise.} \end{cases} \tag{2.9}$$

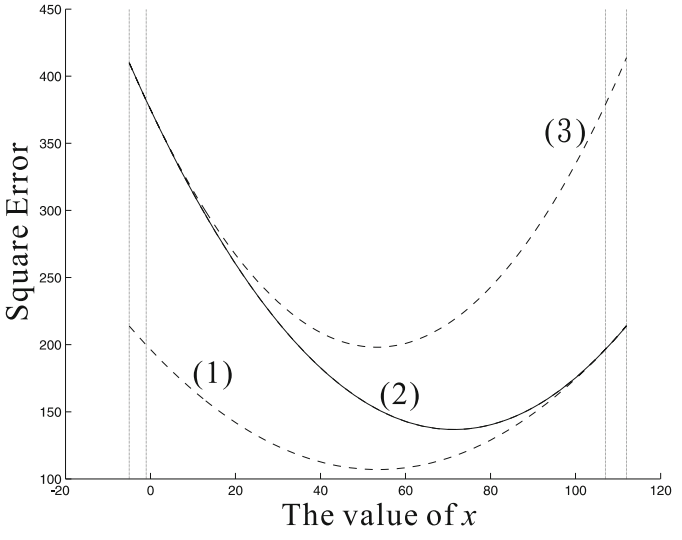

Fig. 2.2. Curves of $(x-112)^2+(x+5)^2$ (curve 1), $(x-112)^2+(x-107)^2+(x+5)^2$ (curve 2) and $(x-112)^2+(x-107)^2+(x+1)^2+(x+5)^2$ (curve 3)

$$c^{"}_{\max} = \max\left(c^{"}_2, c^{"}_3, \cdots, c^{"}_n\right) \tag{2.10a}$$

$$c^{"}_{\min} = \min\left(c^{"}_2, c^{"}_3, \cdots, c^{"}_n\right) \tag{2.10b}$$

where $c^{"}_i, i = 1, \cdots, n$ are the coefficients obtained from an image to be judge, and $w^{'}_i$, $1 \le i \le L$, is the extracted binary value. A comparison between the extracted binary string $W^{'}$, $W^{'} = \{w^{'}_i \mid w^{'}_i \in \{1,0\}, 1 \le i \le L\}$, and the watermark W is then performed. Finally, the number of correct bit is compared with a certain threshold to determine if the watermark exists or not.

2.4 Experimental Results

In this section, some experimental results are presented to illustrate the practicality of our system. In the first part of the experiments, the robustness of the adopted watermarking algorithm was evaluated. An 1000-bit watermark was generated randomly and used in this experiment. To embed one bit of the watermark, twelve coefficients were chosen from $LH2$, $HL2$ or $HH2$ subband. In other words, n is equal to 12 in our experiments. The strength parameter δ is assigned to 0.

To determine the threshold of the number of correct watermark bits, 58600 images chosen from Corel Gallery 1000000 were watermarked using the embedding algorithm described in Section 2.3.3. Then, the threshold was chosen such that watermarked and unwatermarked images could be well separated. The experimental result was shown in Fig. 2.3. According to the result, the value of threshold was assigned to 0.6 in all following experiments.

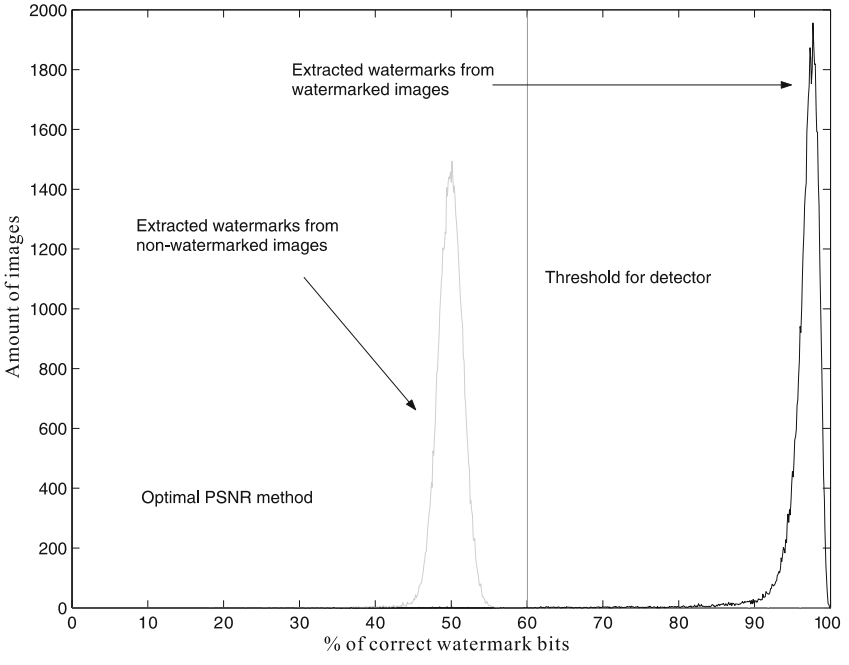

Fig. 2.3. Results of applying watermark detection process to 58600 watermarked and non-watermarked images

To evaluate the robustness of the proposed approach, six popular testing images: Lena, Baboon, F16, Fishing Boat, Pentagon, and Peppers are watermarked with the proposed watermarking approaches. Then, four image processing operations, JPEG compression, Gaussian filtering, sharpening, line removing and rescaling were applied on the watermarked images. The result are shown in Fig. 2.4–Fig. 2.7. As shown in Fig. 2.4, it is obvious that the watermark was still detectable until JPEG quality was lower than 15%. Similar results can be obtained after line removing, Gaussian filtering or sharpening as well as rescaling. These experimental results show that the adopted watermarking approach are robust on minimizing the perceptual distortion.

In the second part of experiments, an HTC desire HD smartphone with Android 2.3.3 was used as a test bed. A 512-bit watermark was used in the experiments, and every watermarked image was resized to quarter size of the original photograph. No visible watermark was added to the image before it was uploaded. In our experiment, six 3264×2448 photographs were taken, watermarked, resized and uploaded to a pre-specific web site with the smartphone. In order to evaluate the image quality, the uploaded images were saved with lossless compression. The photographs uploaded to the web site are shown in Fig. 2.8. After JPEG compression was applied, the number and the percentage of correct bits of each watermark was presented in Table 2.1.

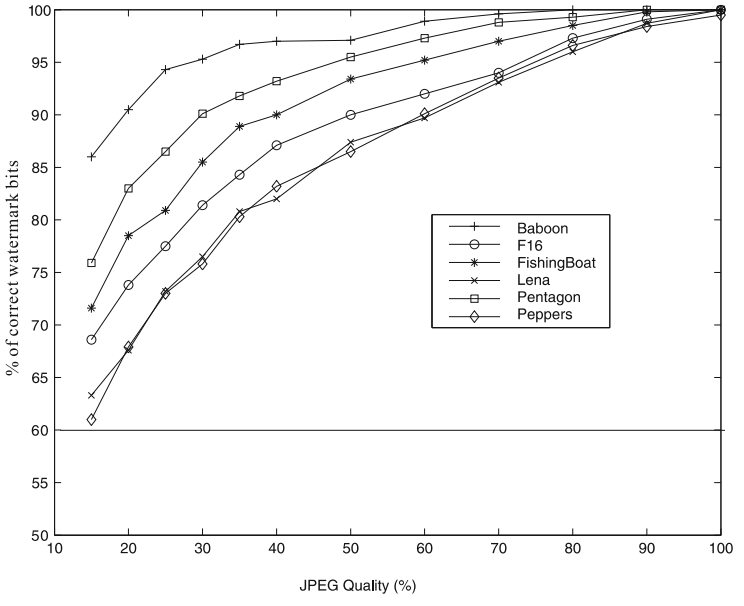

Fig. 2.4. Results of watermark detection after JPEG compression

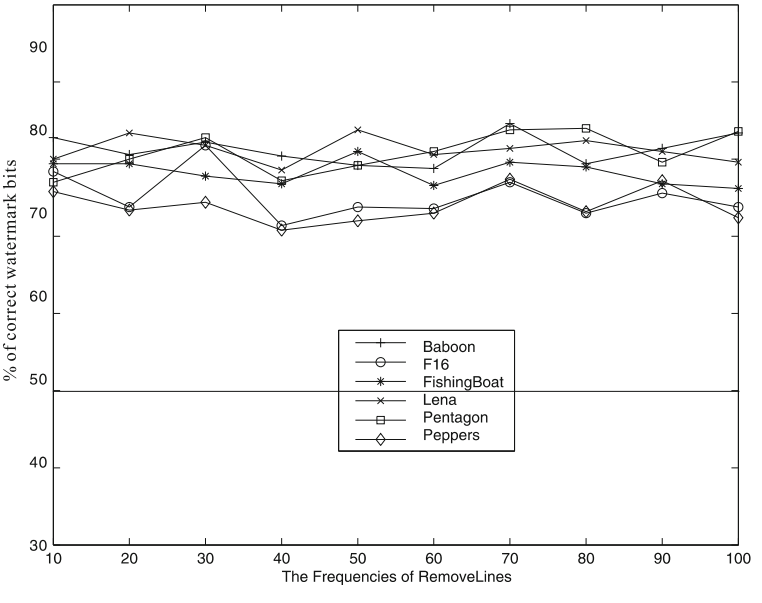

Fig. 2.5. Results of watermark detection after line removing attack

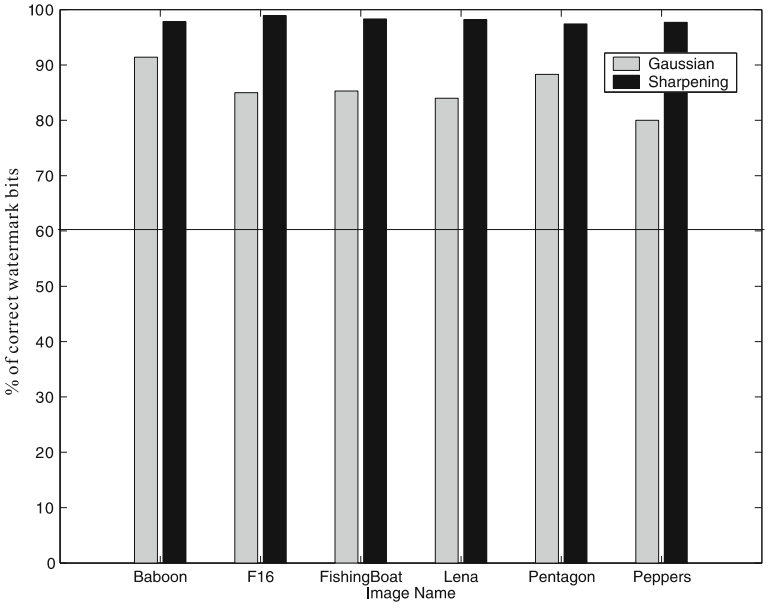

Fig. 2.6. Results of watermark detection after Gaussian filtering and sharpening

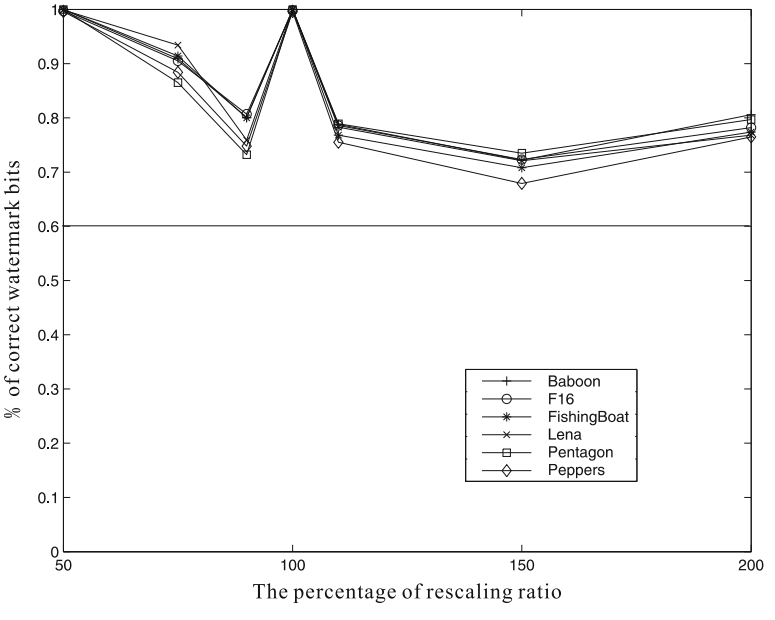

Fig. 2.7. Results of watermark detection after rescaling

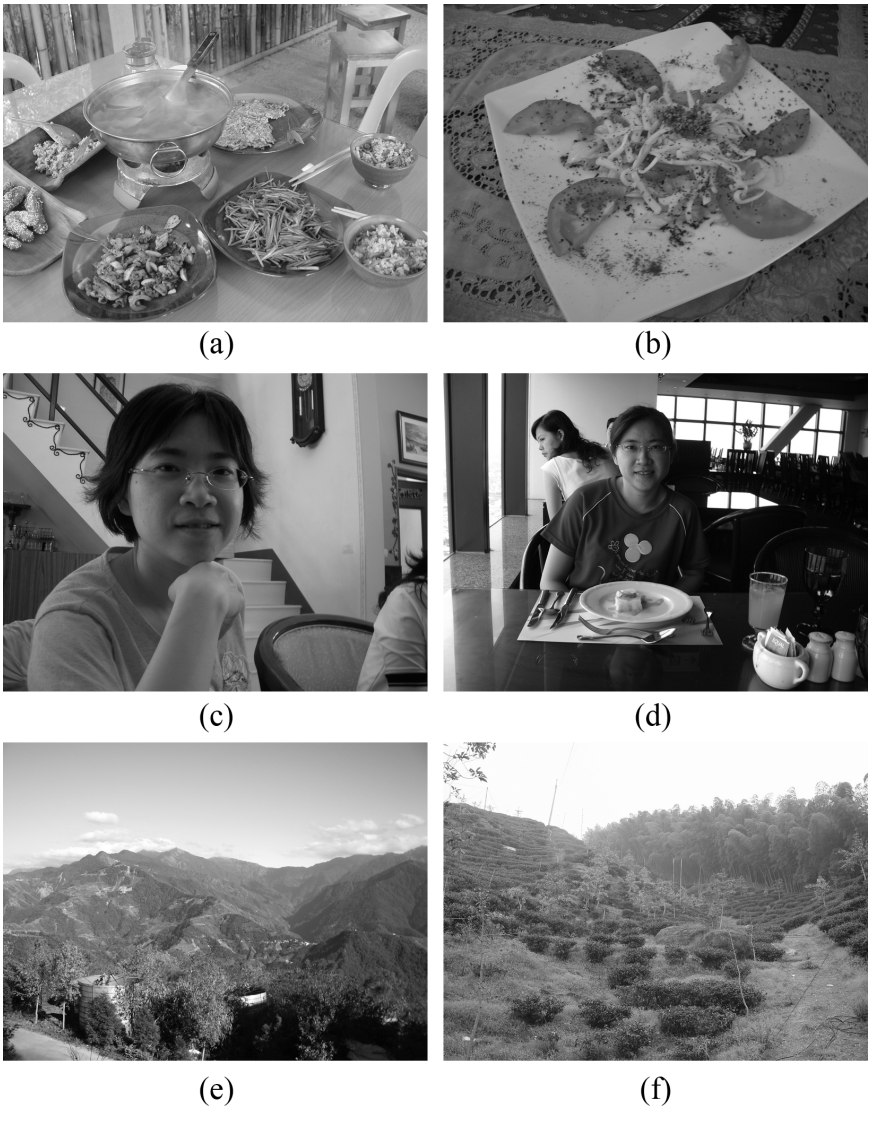

Fig. 2.8. Images watermarked by the proposed system

Table 2.1. The number and the percentage of correct bits of each watermark embedded in the test image.

Images used in the experiment	Quality factor (Q)			
	Q=30		Q=50	
	#	%	#	%
Fig. 2.8(a)	423	82.6%	473	92.4%
Fig. 2.8(b)	427	83.4%	475	92.8%
Fig. 2.8(c)	415	81.1%	466	91.0%
Fig. 2.8(d)	413	80.7%	465	90.8%
Fig. 2.8(e)	437	85.4%	486	94,9%
Fig. 2.8(f)	434	84.8%	481	93.9%

As shown in Table 2.1, though JPEG compression was applied, about 85 percent of the watermark bits are still correct. If a larger quality factor was used, an image with better quality would be obtained, and the percentage of the correct watermark bits would increase to 95%. Since the image resizing was applied while inverse Haar wavelet transform was performed, these two operations can be completed in a few seconds. Thus, the developed system is an practical solution to protect the copyright of the photograph taken by Android smartphone.

2.5 Conclusion

In this chapter, we propose a copyright embedding system for Android platform. Using this system, pre-specified copyright information is automatically embedded into pictures with digital watermark technology when these pictures are taken. In addition, original images (i.e., images without modification) can be preserved selectively. This system has following features:

1. the watermarking approach based on Haar wavelet transform is adopted, and the watermark embedding process itself can also be performed without floating-point computation, so computational complexity of watermark embedding process is effectively reduced,
2. the watermark can be extracted without the use of the original image,
3. the watermark embedded into an image would not be removed by commonly used image processing operations, and
4. embedding copyright information, resizing the images, and uploading the images to Internet are automatically performed without manual intervention.

As shown in Section 2.4, this system is very suitable for the protection of the photographs taken by Android phones to prevent piratical behaviors. Therefore, our approach points out a practical application for smartphones.

References

1. Cox, I.L., Miller, M.L., Bloom, J.A.: Digital Watermarking. Morgan Kaufmann, San Francisco (2001)
2. Cox, I.L., Miller, M.L., Bloom, J.A., Fridrich, J., Kalker, T.: Digital Watermarking and Steganography. Elsevier Science & Technology, London (2007)
3. Huang, H.C., Chen, Y.H., Abraham, A.: Optimized watermarking using swarm-based bacterial foraging. Journal of Information Hiding and Multimedia Signal Processing 1, 51–58 (2010)
4. Huang, H.C., Chen, Y.H.: Genetic fingerprinting for copyright protection of multicast media. Soft Computing 13, 383–391 (2009)
5. Huang, H.C., Fang, W.C.: Metadata-based image watermarking for copyright protection. Simulation Modelling Practice and Theory 18, 436–445 (2010)
6. Pan, J.S., Huang, H.C., Jain, L.C.: Intelligent Watermarking Techniques. World Scientific Publishing Company, Singapore (2004)
7. Pan, J.-S., Huang, H.-C., Jain, L.C. (eds.): Information Hiding and Applications. SCI, vol. 227. Springer, Heidelberg (2009)
8. Hsu, C.T., Wu, J.L.: Hidden digital watermarks in images. IEEE Trans. on Image Processing 8, 58–68 (1999)
9. Duan, F.Y., King, I., Chan, L.W., Xu, L.: Intra-block max-min algorithm for embedding robust digital watermark into images. In: Proc. Multimedia Information Analysis and Retrieval, pp. 255–264 (1998)
10. Bender, W., Gruhl, D., Morimoto, N., Lu, A.: Techniques for data hiding. IBM Systems Journal 35, 313–336 (1996)
11. Chen, Y.H., Su, J.M., Fu, H.C., Huang, H.C., Pao, H.T.: Adaptive watermarking using relationships between wavelet coefficients. In: Proc. IEEE International Symposium on Circuits and Systems, pp. 4979–4982 (2005)
12. Asamwar, R.S., Bhurchandi, K.M., Gandhi, A.S.: Interpolation of images using discrete wavelet transform to simulate image resizing as in human vision. International Journal of Automation and Computing 7, 9–16 (2010)