

# Dot based Printout Watermarking Suitable for Practical Environment

Seung-Jin Ryu<sup>a,\*</sup>, Min-Sik Kim<sup>a</sup>, Dong-Ho Jeon<sup>a</sup>, Jong-Kyu Kim<sup>a</sup>

<sup>a</sup>The Affiliated Institute of ETRI, Daejeon, 34188, Republic of Korea

\*Corresponding Author: sjryu@nsr.re.kr

## Abstract

Information leakage by printouts is common in ordinary office environment. This paper proposes a practical watermark scheme applicable with actual imaging devices to trace printouts comprising watermark patterns. We generate a watermark pattern composed of tiny yellow dots whose coordinates are irregularly located onto the intersections of virtual lattice. We exploit the relationship among the dots' coordinates by Hough transform and discrete Fourier transform to detect the embedded watermark in the printouts. Empirical evidence from a large database of watermark images indicates the superior performance of the proposed method.

**Keywords:** Printout Watermark, Geometric Distortion, Hough Transform, Detection by Smartphone.

## 1. Introduction

With the development of easy-to-use printing devices, printouts are commonly used in our office environment. At the same time, the very nature of printouts, which can be easily leaked, brings into questions many of the positive aspects associated with the printouts. In fact, most of confidential information, such as CAD drawings or organization's internal documents, is disclosed with printouts, and a constantly growing number of uncovered leakages is certainly only the tip of the iceberg.

To prevent or at least trace the leakage of printout, mostly two approaches, i.e., source device forensics<sup>(1)</sup> and digital watermarking<sup>(2)</sup>, were suggested in the literature. Source device forensics, which *passively* investigates a given material without any prior-knowledge, has been researched for the last decade<sup>(3)</sup>. The techniques are based on the assumption that the resulting material includes inherent artifacts to designate a specific source device. Such artifacts due to the printing process, e.g., banding

frequency<sup>(4)</sup>, character features<sup>(5)</sup>, or halftoning effect<sup>(6)</sup>, were revealed by numerous previous research. Unfortunately, the forensic methods mainly focused on identifying printing sources so that the techniques were not adequate to trace specific information of leakage behaviors.

Aside from the *passive* approaches, digital watermarking for printout, which *actively* inserts auxiliary information into a given document, has also been researched intensively. The earliest watermarking method for printout explicitly printed visible logos onto the documents<sup>(7)</sup>. However, scholars rarely consider it as digital watermarking since modern watermarking schemes require imperceptibility. One of the first imperceptible watermarking schemes robust to printing and scanning was proposed by Low *et al.*<sup>(8)</sup>. They slightly shifted text lines up or down, or words left or right from their original positions. Similarly, Borges *et al.* suggested adjusting the brightness of characters according to the insertion information<sup>(9)</sup>. However, these schemes were not able to detect the embedded watermark when the printout with the watermark was geometrically distorted. Recently, printer steganography, which hides unrecognizable tiny yellow dots by the naked eyes to designate printing information, was applied to numerous color laser printers<sup>(10)</sup>. It gave rise to the violation of privacy rights because various color laser printers, made by HP, Canon, or Xerox, occasionally inserted those dots without owner's permission. Inspired by the printer steganography, Briffa *et al.* proposed to embed a plurality of imperceptible yellow dots for documents during the printing process, yet it was weak against geometric distortions<sup>(11)</sup>. Finally, several natural language process (NLP) based watermarking schemes were proposed for printout<sup>(12,13)</sup>. They replaced chosen words with respect to a given codebook unless it did not change the meaning of the sentences. However, the nuances of the text were slightly changed although the transitions were rarely noticeable. Moreover, all of the methods mentioned above only

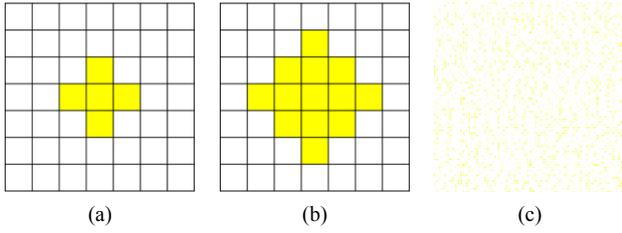


Fig. 1. Dot patterns and a watermark pattern: (a) a dot pattern denoting bit zero, (b) a dot pattern denoting bit one, and (c) a watermark pattern generated by the combination of (a) and (b).

operated with flatbed scanners with which distortions were merely occurred.

To the best of our knowledge, existing watermarking schemes for printout have technical limitations to use practically. Therefore, in this paper, we propose a novel printout watermarking scheme that overcomes the above drawbacks so that it is eventually applicable with smartphone. Specifically, we introduce watermark pattern barely perceptible by the human visual system (HVS). In addition, various kinds of attacks such as digital-analog/analog-digital (D-A/A-D) conversion, scaling, rotation, perspective projection, and cropping are considered which unintentionally occur during the detecting process with webcam or smartphone. Furthermore, the robustness to partial distortions, i.e., some of the watermark in the printout is damaged by text, is considered as well.

The rest of the paper is structured as follows. We first introduce a proposed dot based watermark generation scheme considering both imperceptibility and robustness to distortions in Section 2. Based on the watermark generated in this section, the proposed watermark detection scheme suitable for practical environment is described in Section 3. Subsequently, Section 4 reports the experimental results from a massive test setup before Section 5 concludes the paper.

## 2. Watermark Generation for Printout

As mentioned above, the practical printout watermark should satisfy both imperceptibility by the HVS model and robustness to various distortions. To satisfy the requirements, we propose a dot based watermarking scheme which highly improves the printer steganography<sup>(10)</sup> and Briffa's method<sup>(11)</sup>. To achieve the imperceptibility, the proposed method outputs irregularly spaced yellow dots of different sizes, which are tiny enough with respect to the

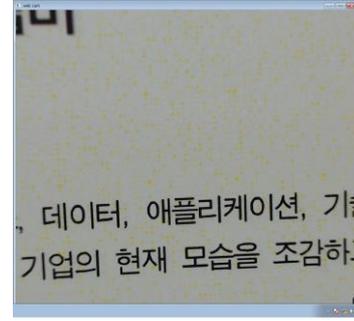


Fig. 2. An acquired image frame in practical environment (by Logitech C920).

HVS model, according to the type of information to be inserted. Since yellow is least noticeable among four colors of CMYK toners, especially accompanied by white colored blank area occupying most of the printout, using the yellow dots are adequate for printout watermark. Fig. 1 shows proposed dot symbols and the example of generated watermark which denote bit zero, bit one, and the combination of two symbols, respectively. The following subsection describes the details of the proposed generation method.

### 2.1 Details of Watermark Generation

As depicted in Fig. 1, we convert each bit of embedding data to a different sized yellow dot within the predefined size of white background area. We normally defines the background size as  $7 \times 7$  pixels because its actual size is about  $0.06 \times 0.06$  cm<sup>2</sup> under 300 DPI printing environment, which is default setting for most of color laser printers. It is tiny enough to satisfy the requirement of imperceptibility. To minimize the perceptibility, we additionally adjust the distances among the dots which are randomly varied by the average distance threshold  $d$ . Eventually, the watermark pattern is comprised of the dot symbols sequentially located with random distances within the appropriate size of square region. Since our ultimate goal is designing a watermark detectable with smartphone, the size of watermark pattern should be restricted in terms of mobile camera's facility. In this perspective, the size of the watermark pattern is empirically chosen around 1 inch<sup>2</sup>, maximally capable with about  $40 \times 40$  symbols. Furthermore, watermark suitable for practical environment should robust to various distortions caused by illumination, toner powder, or characters. For this purpose, we attach Reed-Solomon error correction code to the embedding data<sup>(14)</sup>. The amount of the correction code is varied with circumstances.

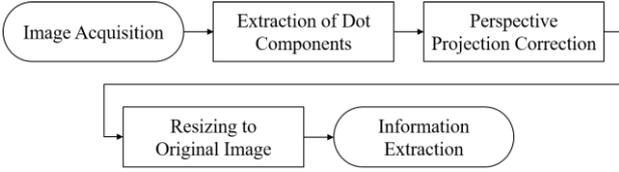


Fig. 3. The procedure of watermark detection method.

### 3. Watermark Detection for Printout

This section presents a watermark detection method from watermarked printout. Fig. 2 depicts an acquired image frame from a watermarked printout by Logitech C920 webcam. As shown in the Fig. 2, we can notice that the digitalized image frame undergoes several attacks such as D/A-A/D conversion, geometrical transformations, gradual lighting changes, cropping, and partial distortions by characters. Therefore, through the following subsections, we explain each step of the proposed detection scheme depicted in Fig. 3. Specifically, we detail the proposed scheme to overcome the attacks mentioned above, so that we prove the practicality of the proposed scheme.

#### 3.1 Extraction of Dot Components

The proposed method first extracts a set of dots from a captured image frame. As described in Section 2, the proposed watermark consists of a purity of yellow dots irregularly printed onto a paper. Since the dots are marked in yellow, we can easily separate the dots by extracting the yellow objects from the given image. Hence, the image is first converted from RGB to hue-saturation-value (HSV) color space because color information is mainly included in the Hue channel<sup>(15)</sup>. Especially, the yellow component is generally depicted around 60° of Hue angle. Therefore, threshold values for Hue, Saturation, and Value are set as follows.

$$\begin{aligned}
 0^\circ &\leq H \leq 120^\circ \\
 0.4 &\leq S \leq 1.0 \\
 0.0 &\leq V \leq 1.0
 \end{aligned}$$

The other benefit of thresholding in the HSV space is shadow removal. Since the shadow, due to the illumination change in practical environment, barely has color information, it is effectively separated in the HSV space. After extracting the yellow objects, mostly assumed as watermark dots, the center coordinate  $(\bar{x}, \bar{y})$  of each object is computed by dividing the x- and y-related first

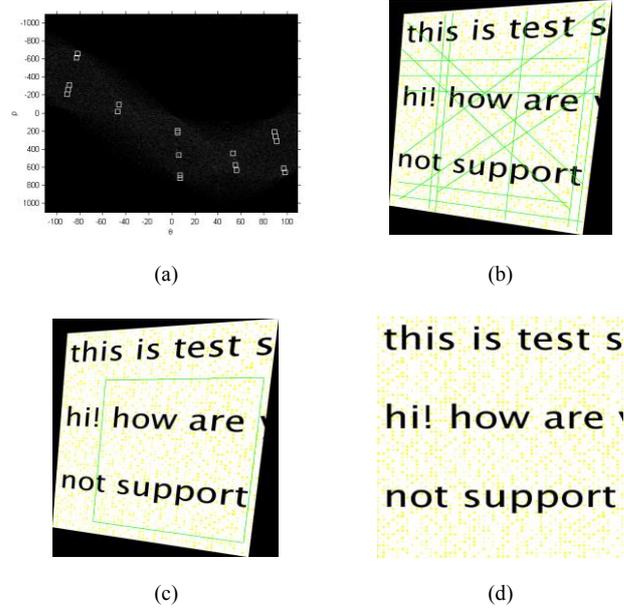


Fig. 4. Snapshots during the correction of perspective projection: (a) extracted lines in Hough domain, (b) the corresponding lines seen in spatial domain, (c) four selected lines for the correction, and (d) the corrected image.

moments of each object by the zeroth moment values<sup>(15)</sup>. This is shown in the following equation.

$$\begin{aligned}
 m_{ij} &= \sum_{x,y} (object(x,y) \cdot x^i \cdot y^j) \text{ where } object(x,y) = 1 \\
 \bar{x} &= \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}}
 \end{aligned} \tag{1}$$

#### 3.2 Correction of Perspective Projection by Hough Transform

Once the candidate coordinates for the dots are decided, the given image is restored to the original shape. The important assumption supporting to detect the watermark in the practical environment is that the dots are irregularly placed onto the intersections of virtual lattice. By revealing the relationship among the dot's coordinates, the reconstruction of the image is performed in two stages. In this section, we explain how to inversely transform perspective projection, which always happens during the image acquisition process.

We first try to find straight lines that pass through as many dots as possible. Since the coordinates for the dots are irregularly located on the intersection of the lattice, many of the straight lines, regardless of geometrical distortions,

belong to either horizontal lines or vertical lines with high probability. In order to find the candidate straight lines passing through the dots, Hough transform is applied to the given coordinates<sup>(15)</sup>. The Hough transform is as follows.

```

for  $\theta = 0$  to  $\pi$ 
   $\rho = x \cos\theta + y \sin\theta$ 
  accumulator[ $\rho, \theta$ ] = accumulator[ $\rho, \theta$ ] + 1
end
select peaks from accumulator

```

After the Hough transform, selected straight lines are represented by  $\rho$  and  $\theta$ , which refer to the distance from the origin and the angle value of the line, respectively. Fig. 4(a) and Fig. 4(b) depict the resulting lines in terms of the Hough and the spatial space, respectively. It is noticeable that the result reveals clusters for both horizontal and vertical lines.

Perspective projected image can be restored to the corresponding original image if four matched pairs of points exist. Therefore, with the extracted lines, especially belonging to either the horizontal or vertical line group, we attempt to estimate four corner points. Specifically, we select two appropriate angles, each of them indicating either the horizontal or vertical line group, followed by two lines in each group are extracted having the minimum and the maximum distance from the origin, respectively. Consequently, four intersections by the selected two horizontal and two vertical lines are assumed the matching points for the perspective projected image, which are then used to transform the perspective projected image into the square of predefined size. Fig. 4(c) and Fig. 4(d) visualize the four intersections with the selected lines and the corrected image, respectively.

### 3.3 Resizing to Original Image

As described in the previous section, the distorted image is mapped into the predefined size of square. Since the intervals between the selected horizontal or vertical lines are not predictable, the aspect ratio of the restored image is not equal to one with high probability. Therefore, we inspect the periodicity between the restored dots through both x-axis and y-axis to resize the image into proper aspect ratio. Recalling watermark generation section, each dot pattern is inserted with a size of  $7 \times 7$  pixels. Therefore, to correct the size of the image, the periodicity of the dots in the x- and y-axis directions is obtained followed by the interval is corrected to 7 pixels. Specifically,

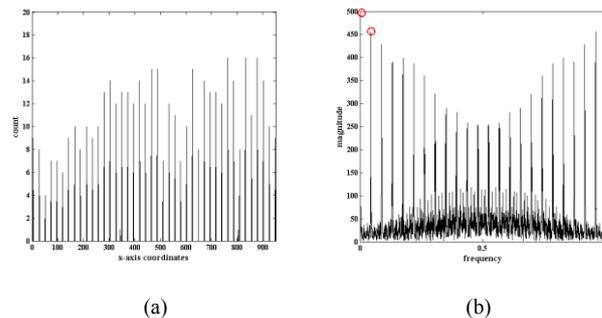


Fig. 5. A x-axis periodicity revealed by resizing procedure: (a) a periodicity of accumulated local maxima through y-axis, (b) and the corresponding impulse train in frequency domain.

the periodicity of the x-axis direction is revealed by accumulating local maxima among dots through the y-axis direction, which is given by

$$\hat{x}(i) = \sum_{j=1}^N map(i, j) \quad (2)$$

Fig 5(a) visualizes the periodicity of accumulated values in spatial domain. Afterwards, Discrete Fourier Transform (DFT) of  $\hat{x}$  is applied to explicitly uncover the periodicity<sup>(16)</sup>.

$$\hat{X} = DFT(\hat{x}) \quad (3)$$

Since the magnitude of  $\hat{X}$  is represented by the impulse train, as shown in Fig. 4(b), the reciprocal of the difference between DC and the first peak is determined as the current periodicity of the x-axis direction. The y-axis periodicity is computed similarly. Finally, the original image is estimated by resizing the given image with respect to the ratio of the computed periodicity to the predefined periodicity.

### 3.4 Decision of Embedded Information

This section presents a data extraction method from the watermark pattern of the reconstructed image. First of all, we determine dots' coordinates by accumulating the image followed by selecting local maxima as described in the previous section. Subsequently, the size of dot's area is computed. Dots, whose areas are less than threshold  $t$ , are considered as empty space because of noisy circumstance (e.g., illumination, shadow, toner powder, and etc.). Thereafter, we decide each dot, whose area is larger than

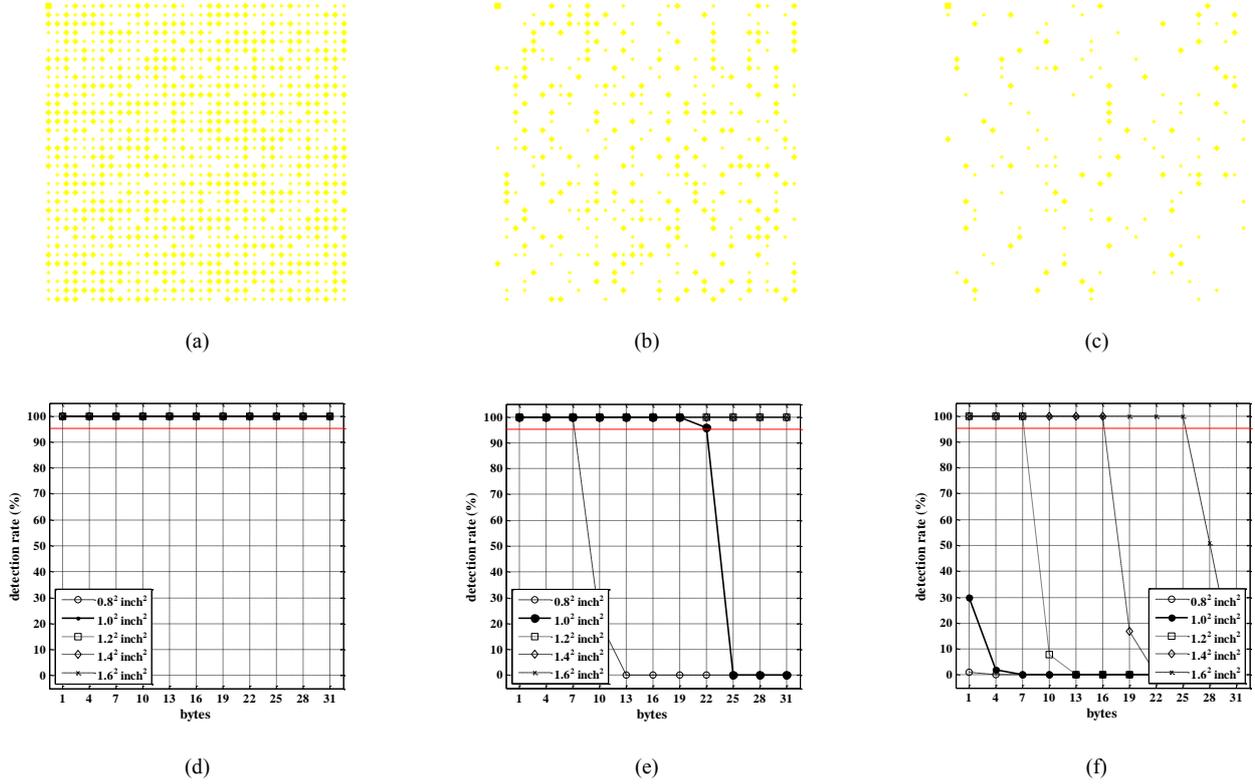


Fig. 6. The shapes of watermark patterns and the corresponding detection results; (a, b, c) visualize the shapes of watermark patterns with the fixed distance threshold  $d=1$ ,  $d=3$ , and  $d=7$ , respectively; (d, e, f) depict the corresponding detection rates varying the size of watermark pattern in the range of  $0.8^2$  to  $1.6^2$  inch<sup>2</sup> in steps of  $0.2^2$  inch<sup>2</sup> and the amount of embedded data from 3 bytes to 31 bytes in steps of 3 bytes.

threshold  $t$ , as one of the two symbols shown in Fig. 2 by 2-means clustering<sup>(15)</sup>. Threshold  $t$  is set as 0.4 empirically. Finally, Reed-Solomon code corrects errors during the detection process.

#### 4. Experimental Results

This section reports results from an extensive series of watermark detection experiments. The setup includes many variations of watermark patterns generated by the proposed method and the corresponding detection performance. Specifically, we evaluated the detection performance along two main directions as follows.

- The *baseline experiments* digitally simulate the performance of the proposed method under various conditions. We measured the detectability of the watermark varying the parameters mentioned in Section 2 and Section 3. Partially distorted watermark patterns with the specific parameter setting were analyzed as well. Besides, geometrical distortions, which inevitably happen during the practical acquisition process, were also simulated before the

following experiments.

- The *practical experiments* investigate the performance of the proposed scheme under actual D-A/A-D conversion, i.e., printing process followed by image acquisition process with scanner and mobile. For this series of experiments, the parameter setting for watermark generation was kept constant, whereas detecting devices and the corresponding circumstances were altered diversely.

If not stated otherwise, we used the following settings. A Reed-Solomon code whose length is same with the corresponding watermark was attached to the watermark so that we guaranteed 25% of error correction ability. A set of experiment, which shares same parameters, was constructed with 100 watermarked images.

##### 4.1 Baseline Experiments

(a) Various Parameter Settings without Distortions

First of all, we investigated the relationship among imperceptibility, capacity, and detectability. For this purpose, we generated numerous watermark patterns with different settings by keeping the distance parameter  $d$ ,

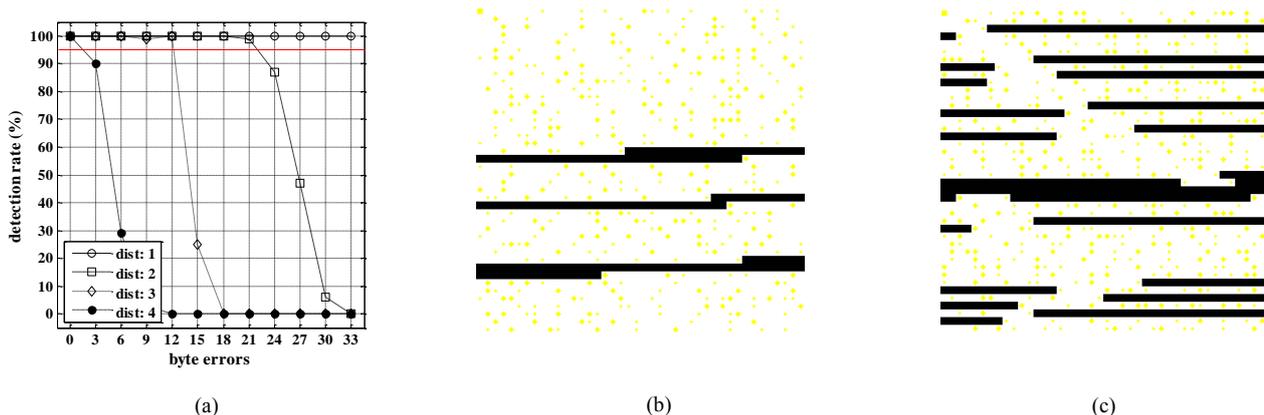


Fig. 7. Watermark detection with partial distortion; (a) depicts the detection rates with respect to the distance threshold  $d$  and the amount of byte errors; (b, c) visualize distorted watermark patterns by 3 byte errors with  $d=4$  and 15 byte errors with  $d=3$ , respectively.

while systematically varying the amount of embedding data and the size of embedding region, resulting in altogether 16,500 runs. The imperceptibility is determined by the distance parameter  $d$  of the proposed method. The closer the gap between the dots, which means smaller  $d$  value, the stronger visual artifact is caused. On the contrary, the embedding capacity is influenced by the distance and the size of the embedding region. The size of the region should be small enough by considering image acquisition process with actual devices such as mobile and webcam. Therefore, we investigated the trade-off among imperceptibility, capacity, and detectability varying the sparseness, the amount of embedding data, and the size of the regions. Fig. 6 depicts the results of the proposed method. The results report up to 19 bytes of embedding capacity with the distance parameter  $d=3$  and the size of 1.0 inch<sup>2</sup>, which satisfy requirements for capacity, imperceptibility, and detectability. Even though the imperceptibility is hard to evaluate, we empirically conclude that the distance of higher than three is sufficient for the criterion by four expert observers participated in the experiments.

#### (b) Robustness to Partial Distortions

Subsequently, the watermark should be robust to the distortions by characters. To verify the robustness, we randomly manipulated embedded bytes in the watermark patterns fixing the size of the region as 1.0 inch<sup>2</sup> whereas varying the distance and the number of erroneous bytes. This series of experiments comprised in total 4800 runs. Fig. 7 shows the detection rate against partial manipulation, 3 bytes manipulation with  $d=4$ , and 15 bytes manipulation with  $d=3$ , respectively. The results revealed the robustness

of proposed method against partial manipulations.

#### (c) Robustness to Geometrical Distortions

Eventually, we manipulated watermarked patterns geometrically, which always occurs during the image acquisition process. The patterns were systematically modified by representative Affine transformation, e.g., scaling and rotation. Separating from the Affine transformation, perspective projection was performed on the given watermark patterns. Since the projection matrix is hard to parameterize, we randomly modified both x-axis and y-axis aspect ratio in the range of 0.9 to 1.1 to generate modified patterns. Table 1 and Table 2 summarize the

Table 1. Watermark detection results with simulated rotations and scalings.

Rotation	Detection Rate [%]				
	Scaling				
	1.0	1.5	2.0	2.5	3.0
0	100	100	100	100	100
10	100	100	100	100	100
20	100	100	100	100	100
30	100	100	100	100	100

Table 2. Watermark detection result with simulated perspective projections.

y-axis aspect ratio	Detection Rate [%]	
	x-axis aspect ratio	
	0	0.9 ~ 1.1
0	100	100
0.9 ~ 1.1	100	100

Table 3. Devices used for the experiments.

Device	Model
Printer	Samsung Multi Xpress C9250ND
Scanner	Epson Perfection V37
Smartphone	Samsung Galaxy S7

detection rate under Affine transformation and perspective projection, respectively. Surprisingly, the proposed method demonstrated detection rate of 100% with various simulated geometrical distortions

#### 4.2 Practical experiments

Experimental results in the following sections depict the performance of the proposed method with actual printing and image acquisition devices. Table 3 lists the devices we used. Randomly generated watermark patterns with  $d=4$ , the size of 0.8 inch<sup>2</sup>, and embedding capacity of 7 bytes were firstly printed onto the papers by Samsung Multi Xpress C9250ND with 300 DPI. Thereafter, the watermarks on the papers were digitalized by Epson Perfection V37 and Samsung Galaxy S7, respectively. With the obtained image frames, we analyzed the performance of the proposed method.

##### (a) Detection with a Scanner

The watermarked image frames were obtained by scanning the papers in the range of 0° to 30°, applied in steps of 10°. Additionally, DPI of the scanner was also varied from 300 DPI to 900 DPI, in steps of 150 DPI. Table 4 completes the detection result of the proposed method under this scenario. The result showed similar result with the previous section because the scanned images were not affected by harsh attacks, e.g., perspective projection or illumination.

##### (b) Detection with a Smartphone

Our last series of experiments analyzed image frames acquired from an actual smartphone’s continuous video. Unlike the scanner, smartphone’s camera offers limited resolution. Because higher resolution provides better image details, we targeted HD and FHD videos, the resolution of 1280×780 and 1920×1080, respectively, which are normally the highest resolution supported by ordinary smartphones. Furthermore, we modified the distance between the watermarked papers and the smartphone from 7 cm to 16 cm in steps of 3 cm because the close distance

Table 4. Watermark Detection results from scanned watermark printouts.

Rotation	Detection Rate [%]				
	DPI				
	300	450	600	750	900
0	100	100	100	100	100
10	100	100	100	100	100
20	97	97	97	97	97
30	97	97	96	97	97

Table 5. Watermark Detection results with image frames acquired by smartphone.

Distance	Detection Rate [%]	
	Resolution	
	HD (1280×720)	FHD (1920×1080)
7 cm	45	53
10 cm	72	76
13 cm	67	71
16 cm	17	21

ensures the details of image frames but might cause lens distortions. Table 5 summarizes the results in this perspective. Compared to the case from a scanner, the results tended to give lower performance. The performance was strictly deteriorated due to lack of details when the distance was too far. The performance was also degraded with the distance of 7 cm. We suspect both lens distortions and the size exceeding of the watermarked region. In spite of the decreased performance, the results from midrange were acceptable applying to actual smartphone because the video frames were able to be processed continuously.

## 5. Conclusions

With the rapid progress of printing technology, we are threatened by a constantly increasing number of information leakages in forms of printouts. To trace the leakage of the printouts, in this paper we have focused on developing a practical watermark scheme applicable with actual imaging devices such as printers, scanners, and smartphones. Given that prior studies for information hiding, which print tiny yellow dots onto papers, we further invented a novel watermarking scheme even detectable with smartphones. By exploiting the relationship among the dots’ coordinates, which are irregularly placed onto the intersections of virtual lattice, we developed a watermark

detector appropriate to porting on smartphones. Experimental results confirmed the superior performance of the approach under a variety of settings. Apart from endeavors to extract printout watermark generated by color laser printers, watermark patterns printed by black and white printers, which are inherently affected by half-toning effect, should be considered in the literature. Several expecting issues caused by adopting black and white printers, e.g., reducing perceptibility, difficulty in separating dots, and increasing complexity, should be solved as well.

## References

- (1) P. J. Chiang, N. Khanna, A. K. Mikkilineni, M. V. O. Segovia, S. Suh, J. P. Allebach, G. T. C. Chiu, and E. J. Delp : “Printer and scanner forensics”, IEEE Signal Processing Magazine, Vol. 26, pp. 72-83, 2009
- (2) I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker : “Digital Watermarking and Steganography”, Morgan Kaufmann, 2008
- (3) H. T. Sencar and N. Memon, : “Digital Image Forensics: There is More to a Picture than Meets the Eye”, Springer, 2013
- (4) G. N. Ali, A. K. Mikkilineni, P. J. Chiang, J. P. Allebach, G. T. Chiu, and E. J. Delp : “Intrinsic and extrinsic signatures for Information hiding and secure printing with electrophotography devices”, in Proc. of the IS&T’s NIP19: International Conference on Digital Printing Technologies, pp. 511-515, 2003
- (5) A. Ferreira, L. Bondi, L. Baroffio, P. Bestagini, J. Huang, J. A. Santos, S. Tubaro, and A. Rocha : “Data-Driven Feature Characterization Techniques for Laser Printer Attribution”, IEEE Trans. On Information Forensics and Security, (will appear) 2017
- (6) S.-J. Ryu, H.-Y. Lee, D.-H. Im, J.-H. Choi, and H.-K. Lee : “Electrophotographic Printer Identification by Halftone Texture Analysis”, IEEE, Int. Conf. on Acoustics, Speech and Signal Processing, pp. 1846-1849, 2010
- (7) G. W. Braudaway, K. A. Magerlein, and F. Mintzer : “Protecting Publicly-Available Images with a Visible Image Watermark”, SPIE/IS&T Intl. Symp. on Electronic Imaging Science and Technology, Proc. Optical Security and Counterfeit Deterrence Techniques, pp. 126-133, 1996
- (8) S. H. Low, N. F. Maxemchuk, and A. M. Lapone : “Document Identification for Copyright Protection Using Centroid Detection”, IEEE Trans. Communications, Vol. 46, No. 3, pp. 372-381, 1998
- (9) P. V. K. Borges, and J. Mayer : “Text Luminance Modulation for Hardcopy Watermarking”, Signal Processing, Vol.87, pp. 1754-1771, 2007
- (10) Printer Steganography, [online] [http://en.wikipedia.org/wiki/Printer\\_steganography](http://en.wikipedia.org/wiki/Printer_steganography).
- (11) J. A. Briffa, C. Culnane, and H. Treharne : “Imperceptible Printer Dot Watermarking for Binary Documents”, Proc. of SPIE7723, Photonics, and Digital Technologies for Multimedia Applications, 77230M, 2010
- (12) M. J. Atallah, V. Raskin, M. Crogan, C. Hempelmann, F. Kerschbaum, D. Mohamed, and S. Naik : “Natural Language Watermarking: Design, Analysis, and a Proof-of-Concept Implementation”, Proc. of Information Hiding, Lecture Notes in Computer Science, Vol. 2137, pp. 185-200, 2001
- (13) M. Topkara, U. Topkara, and M. J. Atallah : “Information Hiding Through Errors: A Confusing Approach”, Proc. of SPIE, Security, Steganography, and Watermarking of Multimedia Contents IX, Vol. 6505, pp. 1-12, 2007
- (14) R. H. Morelos-Zaragoza : “The Art of Error Correcting Coding”, Wiley, 2006
- (15) W. K. Pratt : “Digital Image Processing”, Wiley, 1978
- (16) R. N. Bracewell : “The Fourier Transform and Its Applications”, McGraw-Hill, 1986