A Method for Detecting Watermarks in Print using Smart Phone – Finding No Mark

Takaaki Yamada
Yokohama Research Laboratory, Hitachi., Ltd.
Yokohama, Japan
takaaki.yamada.tr@hitachi.com

Motoki Kamitani
Contents Solutions Division, Hitachi Solutions, Ltd.
Tokyo, Japan

ABSTRACT
A method has been developed for detecting watermarks using a smartphone. A smartphone with a reader application based on this method and held over a watermarked printed image can detect a watermark in the image and use it to obtain corresponding product information. To reduce the computational complexity of a full search for each possible geometrical correction of the image, the locations where watermarks are not embedded are determined in advance. Image quality is maintained by not embedding watermarks for calibration purposes. System testing demonstrated that the developed method is practical to some degree under the constraint that the camera must be positioned within a restricted search range with respect to the printed image.

Categories and Subject Descriptors
H.5.1 [Multimedia Information Systems] Artificial, augmented, and virtual realities

General Terms
Performance

Keywords
Digital watermark, smart phone, sales promotion

1. INTRODUCTION
Continuing advances in mobile broadband networks and improvements in electronic devices have led to widespread use of smartphones and tablets [1]. These devices now typically include a compact camera, a high-resolution display with touch screen, and multiple functions including Internet access with a browser that can display standard web pages. The launching of improved devices in rapid succession is accelerating the development of various information services and applications, such as reader applications.

If a smartphone with a reader application is held over a printed image with an optical mark, the application can read the mark and process the information it conveys. This provides a link between the physical and cyber worlds [2]. Such applications can be used, for example, to speed up the inventory process and to promote sales. In addition to visible barcode and augmented reality (AR) marks, invisible digital watermarks also can be used.

Digital watermarking is used to embed information in content by modifying the content slightly and to detect it. The changes are almost imperceptible to the human eye but can be detected using various means for verification purposes. Watermark detection ideally does not depend on the format in which the content is distributed because watermarks are hidden in the content itself. Many algorithms have been developed for general-purpose image watermarking in both the pixel and frequency domains [3, 4]. Several conventional methods create watermarks with robustness against image processing during printing and scanning [5, 6]. That is, if an image watermarked using such methods is printed, the watermarks can be detected in a scanned image of the print.

Although an application for detecting watermarks can be installed in a camera phone, the computational complexity in watermark detection must be reduced due to the limited computing resources [7] even if with highly efficient smartphones.

We have developed a method that reduces the computational complexity of watermark detection by quickly identifying in advance the locations where watermarks are not embedded. The actual watermark detection process then focuses on the other locations. Because watermarks for calibration are not embedded, the image quality of watermarked images is maintained unlike with most conventional methods.

2. TARGET APPLICATION AND CONVENTIONAL METHODS
2.1 Target application
We are developing a watermark application for sales promotion that works as shown in Fig. 1. An advertiser prepares an original image for a product advertisement and the network address information for the corresponding service site such as an online shopping site. The advertiser generates an ID number unique to the network-address information and stores both the number and information as a record in a database. The advertiser embeds the ID number as digital watermarks in the original image. The watermarked image is then published in print media such as a magazine.

A person viewing the media who is interested in the advertisement can hold a smartphone equipped with our reader application over the advertisement image and continuously capture images of it. The application detects the watermarks in each image and extracts the ID number, i.e., the key to the database. The reader application then accesses the database through the Internet and retrieves the corresponding network-address information. It then invokes the web browser and connects to the service site. The user can then view and/or buy the advertised product. In this way, the person is prompted to visit the shopping site directly from the advertisement.

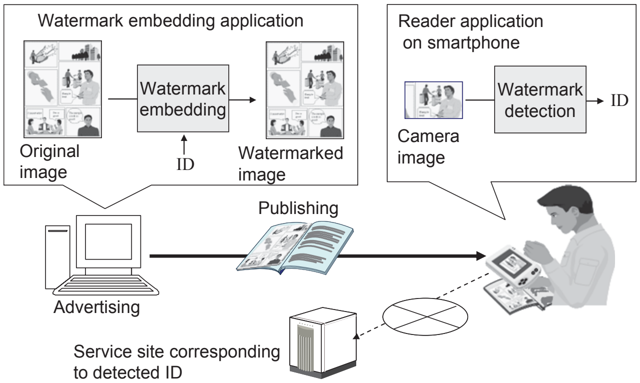

Figure 1. Illustration of target application

2.2 Technical requirements

The target application requires creating watermarks that are almost imperceptible to the human eye and robust against image processing. For practical operation, the original and watermarked images are often stored in compressed format such as JPEG, meaning that the watermarks should be robust against encoding. Moreover, once the image is printed, watermarks should survive analog-digital conversion and still be readable even if the printed image becomes degraded. For instance, the image may become stained or be viewed under poor illumination conditions. When the image is captured by a smartphone camera, it is affected by the camera parameters (focus, sensitivity setting, lens aperture, etc.).

Although watermarks should be readable under such circumstances, we focus on projected images to simplify our discussion. The camera image must be geometrically corrected to enable watermark detection, as shown in Fig. 2.

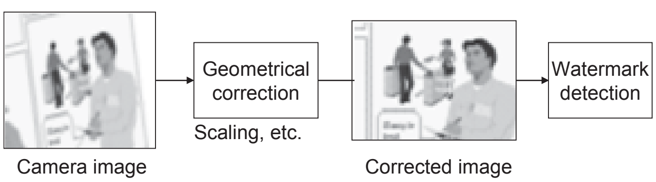

Figure 2. Geometrical correction

2.3 Conventional methods

Methods for efficient geometrical correction in mobile devices with limited computing power include ones that use [7, 8] and do not use [9, 10] visible marks. Visible marks such as frame borders in a watermarked image can help with geometrical correction of the camera image. That is, the camera image can be corrected by searching for the locations of visible marks in the image, assuming that a transformation is given to marks in the printed image, and then applying induced reverse transformation to the camera image. Invisible watermarks can then be detected in the corrected image [7]. Because visible marks have stronger signals than invisible watermarks, they are stably detectable and thus useful for geometrical correction. However, visible marks are often unnecessary and create image noise. We thus do not use visible marks in our application.

If visible marks are not used, geometrical correction is computationally complex. The transformation to be made to the print image can be estimated using a method like image recognition [9]. Otherwise, geometrical correction can be accomplished by using a full-search method. That is, a search is made for the watermarks in the camera image for each possible geometrical correction. For instance, a set of candidate corrected image can be generated by estimating practically-possible set of correction methods and applying each one to the camera image in turn. Correction methods such as rotation, scale, translation, and random distortion are possibly useful [10]. However, reducing computational complexity is critical for the full-search method to be used on mobile devices. Moreover, watermark detection must be done within a few seconds for the method to be used on a smartphone.

We use a research approach to reduce the computational complexity of the full-search method for geometrical correction. We identify in advance the locations where watermarks are not embedded and then detect watermarks in the other locations in the corrected image. This approach differs from both types of conventional methods for geometrical correction. The image quality of watermarked image is maintained with our approach because marks for calibration are not embedded. Our approach also differs from that of the conventional method for three-valued watermarking {-1, 0, 1} [11]. While the conventional method requires full-search, our watermark detection method does not because it uses three steps: 1) identify regions in camera image where watermarks are not embedded, 2) geometrically correct the camera image on the basis of the regions identified, and 3) detect binary-valued watermarks {0,1} in the other regions in the corrected image.

2.4 Algorithm

The algorithm assumed to be used for image watermarking is robust against printing and image capture if parameters such as watermark pattern are adequate.

(1) Definition:

Notation	Definition
V	Set of possible values of element such as image pixel (e.g. from 0 to 255)
N	Number of selected elements in the original image
m	Number of regions
$n = \lfloor N/m \rfloor$	Number of elements in a region
$\mathbf{Y} \in V^N$	Array of element values in the original image
$\mathbf{Y}_j \in V^n$	Array of element values in the j-th region of the original image
$\mathbf{Y'}_j \in V^n$	Array of element values in the j-th region of the watermarked image
$\mathbf{s} = \{s_k \mid 1 \le k \le n\}$	Array of watermark strengths
$\mathbf{M} \in \{-1,+1\}^n$	Array of pseudo random numbers
$P = \{p_k \mid 1 \le k \le n\}$	Array indicating to a watermark pattern
$\boldsymbol{f} : V^N \to V^N$	Image processing function

$\mathbf{c}=\{c_j \mid 1\le j\le m\}$	Estimation value determining whether or not watermark is detected

(2)Embedding

The original image $\mathbf{Y}$ is divided into regions so that

$$\mathbf{Y}=\bigcup_{j=1}^{m}\mathbf{Y}_j,\quad \mathbf{Y}_p\bigcap_{p\neq q}\mathbf{Y}_q=\phi \tag{1}$$

is satisfied.

Each element of a watermark pattern is calculated by multiplying the corresponding watermark strength and pseudo random number: $p_i = s_i \cdot M_i$. The j-th region of watermarked image $\mathbf{Y'}_j$ is obtained by adding watermark pattern P to the corresponding region of the original image.

$$\mathbf{Y'}_j=\mathbf{Y}_j+P \tag{2}$$

The watermark strength can be determined by using a previously reported method [12]. The original image is analyzed by using a human visual model before watermark embedding, and visually important areas, such as edge shapes, where changes are more perceptible than changes in other areas are extracted. The range of element-value changes is estimated on the basis of this analysis. These changes cannot be recognized by the human eye. Maximizing the watermark strength within this range results in watermarks that are almost imperceptible but survive image processing.

(3) Watermark detection

The watermark pattern is found by calculating a correlation value on the basis of Equation 3 and determining whether c_j is greater than a positive threshold value T_{high}, which is defined beforehand on the basis of Equation 4.

$$\begin{aligned} c_j &= \frac{1}{n}\sum_k M_k \cdot Y'_{j,k} = \frac{1}{n}\sum_k M_k (Y_{j,k}+s_k \cdot M_k) \\ &= \frac{1}{n}\sum_k M_k \cdot Y_{j,k} + \frac{1}{n}\sum_k s_k \end{aligned} \tag{3}$$

Note that the first term of Equation 3 should be near zero due to the randomness of $\mathbf{M}$. The second term should determine the correlation value independent of the image content. Therefore, c_j follows a normal distribution with its mean being the mean of the watermark-strengths s_k.

$$result=\begin{cases} b_j & if\ c_j \ge T_{high} \\ not\ detected & if\ else \end{cases} \tag{4}$$

A one-bit schema can be implemented by applying one watermark pattern if the embedded bit value is 1 and applying another watermark pattern if the value is 0. In a multiple-bit schema, the image is divided into regions and the one-bit process is applied to each region.

In the assumed application, watermarked image $\mathbf{Y'}$ undergoes with various image processing. Therefore, the estimation value should be calculated using $f(\mathbf{Y'})$ rather than the $\mathbf{Y'}$ in Equation 3.

3. FINDING LOCATIONS WHERE WATERMARKS ARE NOT EMBEDDED

3.1 Native watermark

If the number of elements in a region of the image is small, the content-dependant value, the first term of Equation 3, $\frac{1}{n}\sum_k M_k \cdot Y_{j,k}$, does not approach zero. Moreover, if the watermark strength is low, the content-dependant value may exceed the second term of Equation 3. This would disturb watermark-detections with the conventional methods as well as our proposed method for finding locations where watermarks are not embedded.

If the array of selected element values in the original image $\mathbf{Y}$ is natively correlated with watermark pattern $\mathbf{M}$, false positive errors can easily occur. We call such content-specific signals as "native watermarks." It is generally possible to define a watermark pattern so that the probability of false positives can be mathematically calculated as low one. By using such a pattern, the probability of false positives can be reduced. These native watermarks should be weakened during the embedding process to reduce detection failures.

3.2 Embedding procedure

Step 1 (input): Determine input parameters n, N, m, and $\mathbf{M}$ and specific regions where watermarks are not embedded, indicated by the array of the specific region numbers, $\mathbf{h}=\{h_i \mid 1\le i\le h_{\max}, h_{\max}<m\}$. Read original image, divide it into regions, and repeat following steps for each region.

Step 2 (primary embedding): Encode multiple bits of embedding information into watermarks, and embed them in regions on basis of (2) if $j\neq h_i$. If the number of elements in a region is great enough, skip steps 3 and 4 because the effect of native marks is less important.

Step 3 (native watermark reduction): Weaken native watermarks in specific regions of watermarked image. Calculate adjustment value D_j given by Equation 5 by averaging effect of correlation value between an array of element values for each specific region in the original image and the watermark pattern,. Each element value for the specific regions is subtracted from the adjustment value under the constraint that the adjustment value is less than the watermark strength.

$$\begin{aligned} &D_j=\frac{1}{n^2}\sum_k M_k \cdot Y'_{j,k},\quad Y'_{j,k}\leftarrow Y_{j,k}-D_j \\ &\quad if\ \ j=h_i\ and\ D_j<s_k \end{aligned} \tag{5}$$

Step 4 (error correction): If adjustment value D_j is so small near the minimum accuracy of the element value (e.g. 1), subtract the error-correction value so that native watermarks disappear and rounding errors are reduced.

Step 5 (output): Output watermarked image $\mathbf{Y}'$. Note that watermarks are not embedded in specific regions of the image.

3.3 Watermark detection procedure

Step 1 (input): Use the same parameters (n, N, m, $\mathbf{M}$, and $\mathbf{h}$) as used in the embedding procedure. Define two positive threshold values T_{low} and T_{high}. Read an input image $\tilde{Y}$. Assume that it was given by a geometrical transformation function $r(Y')$. Since the detection process is unaware of what transformation was given, the preset search range is a set of practically possible parameters for simulating transformation. The following steps are repeated for the search range until a successful result is output.

Step 2 (candidate image generation): Generate a candidate image by applying reverse transformation to the input image, that is $Y''=r^{-1}(\tilde{Y})$. For instance, when a parameter in the search range for scaling is 50%, we assume that the input image was output from a function for scaling half. Corresponding reverse transformation is a function for scaling an image by 200%. That is, corresponding candidate image can be generated by scaling the input image by 200%. If the function is correctly assumed, the candidate image Y'' has the same aspect ratio as the watermarked image Y'.

Step 3 (correlation in each specific region calculation): Divide the candidate image into regions and calculate correlation value $\tilde{c}_j$ for each one (if $j = h_i$) on the basis of Equation 3 (note that candidate image Y'' is used instead of Y'). If the correlation value is less than a threshold T_{low} on the basis of Equation 6, watermarks are not embedded in the region. If all specific regions are found, the candidate image can be the corrected image. See Fig. 3. If they are not satisfied, generate another candidate image (i.e., return to step 2).

$$\begin{aligned} &result = no\ watermark \\ &\quad if\ \tilde{c}_j < T_{low}, 0 < T_{low} < T_{high} and\ j = h_i \end{aligned} \tag{6}$$

Step 4 (information decoding): Determine on basis Equation 4 whether watermarks are embedded in other regions of the corrected image.

Step 5 (output): Verify decoded bits of information by using methods such as one for error correction. Output verified data as successful watermark detection result.

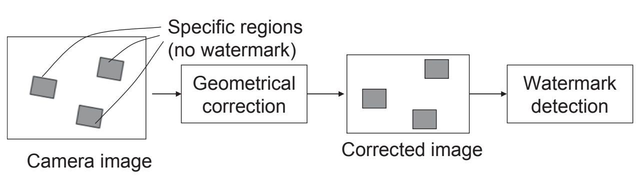

Figure 3. Finding specific regions where watermarks are not embedded.

In the multiple-bit schema, embedded information can be extracted by applying these steps repeatedly. A process checking for absence if watermark is in would face with both watermark patterns which represent '0' and '1' bits. To avoid computing correlation value twice, both watermark patterns must be similar in the correlation calculation.

4. PROTOTYPE

4.1 Description

The information payload was 64 bits, which is enough to identify content as shown in Fig. 1. We implemented our proposed method for embedding and detecting as a PC software prototype. Moreover, the algorithm used for watermark detection was implemented as a reader application in a current smart phone. It was used for system testing. Its specifications are listed in Table 1.

Once the reader application is invoked, it starts to capture images and tries to detect watermarks in each consecutive image, as shown in Fig. 4. Each image is output to the display one by one. If watermarks are detected, a popup message with the detected information is output on the display. If watermarks are not detected, the application processes the next image. The frame rate is 10 frames per second (fps), and this is the speed of watermark detection.

Along with capturing an image in the input video stream when it is held over a printed image, the prototype can capture a transformed image of the printed image. Our geometrical correction method searches for the various possible affine transformations including (1) shifting, (2) scaling, and (3) rotation. From the viewpoint of user behavior, these transformations are equivalent to; (1) shifting the camera horizontally in the range from -10 to 10 cm, (2) holding the camera from 9 to 11 cm (10 ± 1 cm) above the printed image, and (3) rotating the camera horizontally within the range ± 3°.

Figure 4. Reader application prototype.

Table 1. Smart phone used.

Component	Specifications
Memory	RAM 512MB, ROM 1GB
CPU	1GHz
Camera	Backside illumination, CMOS sensor, 8 M-pixels
Display	4.2 inch, 854×480 pixels

5. UNIT TESTING

5.1 Watermark robustness

Watermark robustness was evaluated by detecting watermarks in original uncompressed image content that were encoded twice. In the first encoding, the watermarked uncompressed image was encoded so that the file size of the compressed file would be no more than 10% of the uncompressed one (meaning 1/10 JPEG compression). The compressed image was decoded once into uncompressed format. In the second encoding, the decoded image was re-compressed so that the file size was no more than half that of the first compressed file (almost 1/20 JPEG compression). Watermark detection was done using decoded images from the re-encoded 1/20 JPEG file.

The peak-signal-noise-ratio (PSNR) for an unwatermarked image with 1/10 JPEG compression was 41.6 dB. Therefore, we set the watermark strength so that the image quality of the watermarked image would be nearly the same level as that of the unwatermarked image with 1/10 JPEG compression. The watermark strength was calculated in the same way for both the conventional and proposed embedding methods [12]. The PSNR values for a sample image watermarked using the conventional and proposed method were respectively 41.3 dB and 41.4 dB.

The double-encoded watermarks were successfully detected without any bit errors. A base method such as a watermark pattern with both the conventional and proposed method can provide practical watermark robustness against encoding.

5.2 Image quality of uncompressed image

Step 2 of the embedding procedure in the proposed method does not embed watermarks in specific regions, as described in Section 3. Moreover, even if the image is slightly modified in Step 3 by reducing native marks, the noises created by subtracting the adjustment value given by Equation 5 is theoretically smaller than that of the watermark strength. Therefore, the quality of images watermarked using the proposed method is theoretically maintained much better than with the conventional method given by Equation 2.

The PSNR values were calculated under the same conditions used for the robustness test. They are plotted in Fig. 5 against the ratio of specific regions. The ratio of the number of specific regions to that of all regions, h_{max} / m, affects image quality with the proposed method. The greater, the ratio is, the better, image quality is maintained. The PSNR value with the conventional method was constant and independent of the ratio.

As mentioned above, the PSNR values in the robustness test for the conventional and proposed methods were respectively 41.3 and 41.4 dB. Although the difference is imperceptive difference,

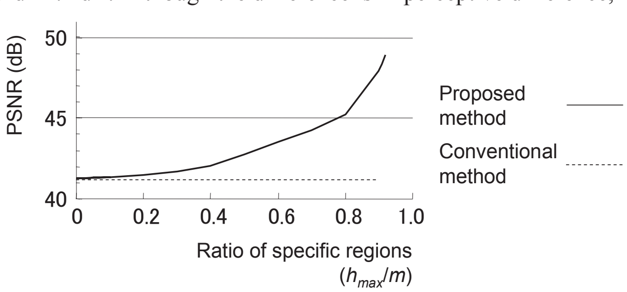

Figure 5. PSNR vs. ratio of specific regions.

those results demonstrates that total changes of element values by the proposed method was actually less than that with the conventional method. The results shown in Fig. 5 demonstrate that the PSNR can be improved by increasing the ratio of specific regions.

5.3 Computational complexity

The proposed method can quickly correct images geometrically because the geometric compensation uses only a few small regions of the image. The computational complexity of finding specific regions where watermarks are not embedded, given by Equation 6, is theoretically the same order as that of finding specific region in where calibration-purpose watermarks are embedded, given by Equation 4.

The proposed method by not embedding calibration-purpose watermarks has almost the same computation speed for detecting watermarks as the conventional method.

6. SYSTEM TESTING

6.1 Print image quality

The conventional criteria for evaluating watermarking technology do not include ones for measuring the quality of watermarked printed images. We used watermark strength in system testing of 1, 2, and 3, stronger than in the unit testing.

The sample image was 640×480 in RGB 24-bit color format. We embedded watermarks in the original image. The original and watermarked images were then printed in color. We used a common laser beam printer and the standard settings for the printer driver for photo printing with 600 dpi. The print size was 75×100 mm on an A4 sheet (210×297 mm) of plain paper.

The original and watermarked printed images were subjectively compared on a desk in an office under stable fluorescent illumination. Six participants evaluated the images and rated the watermark disturbance on a scale of 1 to 5 (Table 2). The average scores were respectively 5.0, 4.5, and 3.5 for the three watermark strengths.

These results indicate that watermark strength can be adjusted so as to obtain a particular print quality. The evaluated score of 4.5 for a watermark strength of 2 means that the image distortion caused by watermark embedding with the proposed method was almost imperceptible to the human eye. That is, the image quality of image watermarked using proposed method is practically useful.

Table 2. Level of disturbance rating scale

Disturbance	Score
Imperceptible	5
Perceptible but not annoying	4
Slightly annoying	3
Annoying	2
Very annoying	1

6.2 Usability

A smartphone with our reader application and held over a watermarked printed image using the proposed method, as shown in Fig. 4, could detect the watermarks in the images. It could also trigger the sending of the decoded information to the server through the Internet and in turn get corresponding network address information such as a URL, as shown in Fig. 1.

Detection of watermarks with a strength of 2 took about one second for an experienced user. Detection sometimes took several seconds for an inexperienced user despite the user having received instructions regarding the use of a search range, as described in Section 4.

These results demonstrate that the reader application is practical at some degree under the constraint that the camera must be positioned within a search range with respect to the printed image.

6.3 Information security

Common threats related to digital watermark technology include the creation and distribution of false content from which the original watermarks have been removed and/or false watermarks have been added. If a false watermark connect the user to a site that impersonates an original site, such a site could be a phishing site designed to steal a visitor's financial information, password, private information, and so on. However, this is not a problem with our reader application because only network addresses registered in the database are accepted (see Section 2.1). The users of our reader application are thus safe from this threat. In contrast, users of services using barcodes of network address face such security threats.

7. CONCLUSION

Aiming at developing a watermark application for sales promotion, we have developed a method for embedding watermarks and detecting them using a smartphone. Because watermarked printed image become strained, the camera image must be corrected geometrically before watermark detection. To reduce computational complexity of a full search for each possible geometrical correction, we set limits on the search range by assuming practically possible sets of transformations for the watermarked image. Geometrical correction is done on the basis of the specific regions found not to have embedded watermarks. Actual watermarks can be detected in other regions in the corrected image.

The results of unit testing demonstrate that proposed method maintains the quality of watermarked image because it does not embed watermarks in specific regions, although watermark detection using the proposed method has almost the same computational complexity as with the conventional method. System testing showed that a smartphone with our reader application and held over a watermarked printed image could detect a watermark in the image and use it to access corresponding information. Our reader application is thus practical at some degree.

Future work includes further development to improve usability. Usability could be improved by easing the constraint that the camera must be positioned within a particular range with respect to the printed image. It could also be improved by considering projection transformation and skewing.

Some parts of the developed method are used in commercial product.

8. REFERENCES

[1] Charland, A. and Leroux, B.: "Mobile Application Development: Web vs. Native", Communications of the ACM, 54, 5 (2011), 49-53.

[2] Kamijo, K. Kamijo, N. and Gang, Z.: "Invisible Barcode with Optimized Error Correction", in Proceedings of IEEE Int'l Conf. on Image Processing (ICIP), (2008), 2036 –2039.

[3] Wuand, M. and Liu, B.: "Multimedia Data Hiding", Springer-Verlag, NewYork (2003).

[4] Potdar, V. M., Han, S., and Chang, E.: "A Survey of Digital Image Watermarking Techniques", in Proceedings of IEEE Int'l Conf. on Industrial Informatics (INDIN05), (2005), 709-716.

[5] Yamada, T., Fujii, Y., Echizen, I., Tanimoto, K., and Tezuka, S.: "Print Traceability Systems Framework using Digital Watermarks for Binary Images", in Proceedings of IEEE Int'l Conf. on Systems, Man and Cybernetics (SMC2004), (2004), 3285-3290.

[6] Goto, Y., Uchida, O.: "Digital Watermarking Method for Printed Materials Embedding in Hue Component", in Proceedings of Int'l Conf. on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP09), (2009), 148 – 152.

[7] Nakamura, T., Katayama, A., Kitahara, R., and Nakazawa, K.: "A Fast and Robust Digital Watermark Detection Scheme for Cellular Phones" NTT Technical Review, 4, 3 (2006), 57-63.

[8] Perli, S. D., Ahmed, N., and Katabi, D.: "PixNet. Designing Interference-free Wireless Links using Lcd-camera Pairs",.In Proceedings of ACM Int'l Conf. on Mobile Computing and Networking (MOBICOM), (2010), 137-148.

[9] Yuan, W., Dana, K., Ashok, A., Varga, M., Gruteser, M., and Mandayam, N.: "Dynamic and Invisible Messaging for Visual MIMO", in Proceedings of the IEEE Workshop on the Applications of Computer Vision (WACV), (2012), 345-352.

[10] Atomori, Y., Echizen, I., Dainaka, M., Nakayama, S., Yoshiura, H.: "Robust Video Watermarking based on Dual-plane Correlation for Immunity to Rotation, Scale, Translation, and Random Distortion", Journal of Digital Information Management, 6, 2 (2008), 161-167.

[11] Louizis, G., Tefas, A., and Pitas, I.: "Copyright Protection of 3D Images using Watermarks of Specific Spatial Structure," in Proceedings of IEEE Int. Conf. on Multimedia and Expo (ICME2002), (2002), 557-560.

[12] Yamada, T., Ebisawa, R., and Takahashi, Y.: "Maintaining Image Quality when Watermarking Grayscale Comic Images for Electronic Books", in Proceedings of 9th IEEE Int'l Conf. on Industrial Informatics (INDIN2011), (2011), 322-327.