A ROTATION, SCALING AND TRANSLATION RESILIENT IMAGE WATERMARKING ALGORITHM USING CIRCULAR GAUSSIAN FILTERS

Yiwei Wang and John F. Doherty

Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802, USA
{yxw131,jfdoherty}@psu.edu

Robert E. Van Dyck

National Institute of Standards and Technology
Gaithersburg, MD 20899, USA
vandyck@antd.nist.gov

ABSTRACT

The proliferation of data exchange on the Internet presents a challenge in the field of copyright protection, as the unauthorized duplication and distribution of multimedia data become easier. Digital watermarking has been proposed to solve the problem. However, most image watermarking algorithms lose synchronization under rotation, scaling and translation operations. Several papers have been published to address the problem. Most of them use methods in the Fourier transform domain. In this paper, we propose another approach utilizing circular Gaussian filters. Experimental results are given to demonstrate the effectiveness of our algorithm.

1. INTRODUCTION

In the last ten years, the Internet has grown rapidly. This fact, combined with the success of personal computers, makes the piracy of multimedia data easier, and therefore presents a challenge in the field of copyright protection. As a possible solution to this problem, digital watermarking, which is embedding copyright information into the data, has drawn a lot of attention. However, most image watermarking algorithms fail when the image goes through geometrical transforms, such as rotation, scaling and translation (RST). The reason is that geometrical transforms destroy the image synchronization which is vital to most watermarking algorithms [1, 2]. Several methods have been proposed to address this problem [3, 4, 5, 6]. Most of them use the translation invariant property of the Fourier transform magnitude coefficients. In this paper, we present another approach based on circular Gaussian filters.

The rest of the paper is organized as follows: In Section 2, we review the properties of circular Gaussian filters. The algorithm is described in Section 3. Experimental results are demonstrated in Section 4. At the end, we present conclusions and comments in Section 5.

2. CIRCULAR GAUSSIAN FILTERS

A 2-D filter is called a circular filter if its impulse response, $h(x,y)$, satisfies the following requirement: if $x_1^2+y_1^2 = x_2^2+y_2^2$, then $h(x_1,y_1) = h(x_2,y_2)$. If we apply a circular filter to an image and sort the filtered image into an ordered vector, assuming that we disregard the boundary situations, it is obvious that the ordered vector is the same regardless of the rotation and translation of the original image.

The 2-D Fourier transform of function $f(x,y)$ is defined by [7]

$$\begin{aligned} F(u,v) &= \mathbf{F}(f(x,y)) \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\exp(-j2\pi(ux+vy))dxdy. \end{aligned} \quad (1)$$

The inverse Fourier transform is then [7]

$$\begin{aligned} f(x,y) &= \mathbf{F}^{-1}(F(u,v)) \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v)\exp(j2\pi(ux+vy))dudv. \end{aligned} \quad (2)$$

One useful property of the Fourier transform is that scaling in the original spatial/time domain causes an inverse scaling in the frequency domain. If $\mathbf{F}(f(x,y)) = F(u,v)$, then

$$\mathbf{F}(f(kx,ky)) = \frac{F(u/k,v/k)}{k^2}. \quad (3)$$

The circular Gaussian filter is a specific class of circular filter. Its impulse response is

$$g(x,y) = (\exp(-\frac{x^2+y^2}{2\sigma^2}))/(2\pi\sigma^2). \quad (4)$$

It is well known that the Fourier transform of a Gaussian function is still Gaussian. The Fourier transform for the function in equation (4) is

$$G(u,v) = \exp(-2\pi^2\sigma^2(u^2+v^2)). \quad (5)$$

Furthermore, assuming $g(x, y)$ is a Gaussian filter as described by equation (4) and $G(u, v)$ is its Fourier transform, if

$$g^*(x, y) = (\exp(-\frac{x^2 + y^2}{2\sigma^{*2}}))/(2\pi\sigma^{*2}), \tag{6}$$

where $\sigma^* = k\sigma$, then its Fourier transform

$$G^*(u, v) = G(ku, kv). \tag{7}$$

Therefore, scaling the parameter of σ of a Gaussian filter is equivalent of scaling its frequency response.

3. ALGORITHM

A circular Gaussian filter

$$g_0(x, y) = (\exp(-\frac{x^2 + y^2}{2\sigma_0^2}))/(2\pi\sigma_0^2) \tag{8}$$

is applied on an image $I_0(x, y)$, and the resulting image is $R_0(x, y)$. Assuming E_{I_0} is the energy of $I_0(x, y)$ and E_{R_0} is the energy of $R_0(x, y)$, we can define the energy factor $\alpha_{I_0}^{R_0} = E_{R_0}/E_{I_0}$. If the image is spatially scaled by a factor k, i.e.,

$$I_1(kx, ky) = I_0(x, y), \tag{9}$$

we know from Section 2 that by using a circular Gaussian filter $g_1(x, y) = (\exp(-\frac{x^2+y^2}{2\sigma_1^2}))/(2\pi\sigma_1^2)$, where $\sigma_1 = k\sigma_0$, we can obtain a result image $R_1(x, y)$ that produces the same energy factor (i.e., $\alpha_{I_1}^{R_1} = E_{R_1}/E_{I_1} = \alpha_{I_0}^{R_0}$). Although the above derivation is for analog images and filters, their digital counterparts follow the same relationship closely, given that reasonable interpolation is used.

To embed a watermark in a grayscale image, first we normalize the size of the image to remove possible scaling effects. Given image I and a preset energy factor range $[\alpha - \epsilon, \alpha + \epsilon]$, we adjust σ of a circular Gaussian filter to produce a resulting image R that gives an energy factor in that range. Using σ and a preset variable σ^*, we decide the scaling factor $k^* = \sigma^*/\sigma$ and then adjust the size of the image $I^*(k^*x, k^*y) = I(x, y)$.

Second, we apply the Gaussian filter

$$g^*(x, y) = (\exp(-\frac{x^2 + y^2}{2\sigma^{*2}}))/(2\pi\sigma^{*2}) \tag{10}$$

on I^* for n times. The difference image D of the nth and the $(n-1)th$ time is computed. The locations of pixels with large amplitudes in D are usually the locations of dominant edges. These locations are good places to embed the watermark since human visual systems are less sensitive around the edges. We apply the Gaussian filter several times so that the locations of dominant edges are selected while the effects of short edges and noise are suppressed. A threshold T is calculated using a predetermined function. The elements

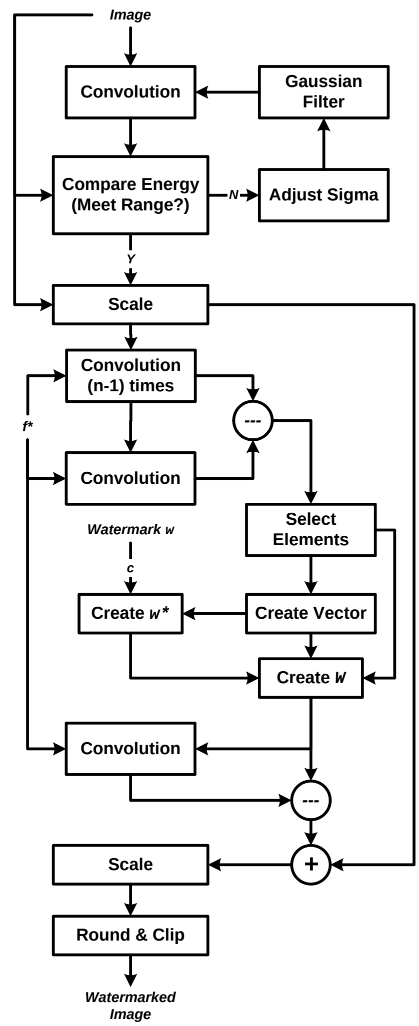

Figure 1: Block diagram of the watermarking process

in D, whose amplitudes are greater than T, are selected and sorted into an ordered vector $\overrightarrow{v_d}$. The watermark $\overrightarrow{w}$ that we use is a random vector of dimension l. The values of $\overrightarrow{w}$ are ± 1. First, $\overrightarrow{w}$ is multiplied by a factor c to increase the watermark energy. Because l is very small compared to the dimension of $\overrightarrow{v_d}$, denoted by L, we then stretch it to $\overrightarrow{w^*}$ to match the dimension of $\overrightarrow{v_d}$ using the following equation:

$$\overrightarrow{w^*(m)} = \overrightarrow{w(n)}, \frac{(n-1)L}{l} < m \leq \frac{nL}{l}. \tag{11}$$

An image W is created from $\overrightarrow{w^*}$ by matching the relationship of D and $\overrightarrow{v_d}$ (i.e., if $D(i, j) = \overrightarrow{v_d}(m)$, then $W(i, j) = \overrightarrow{w^*}(m)$). The watermark image W_i is defined by $W_i = W - W * g^*$, where $*$ denotes 2-D convolution. W_i is added to I^* and the result is scaled to the original size. The grayscale values are then rounded and clipped to create the watermarked image. A block diagram is given in Figure 1 to illustrate the watermarking process. The mapping between D and $\overrightarrow{v_d}$ for the watermarked image will be different from the mapping of the original image. The reason of adding W_i instead of W is to suppress the effect of the watermarking process so that the mappings for the watermarked image is similar to that of the original.

The detection process is very similar to the watermarking process. First, the suspect image is scaled to achieve a certain energy factor. Following steps similar to the ones in the previous paragragh, we can create a reference watermark image $W_i^\dagger$ from a suspect image. Then another image S is created by $S = I^\dagger - I^\dagger * g^*$, where $I^\dagger$ is the scaled suspect image. A test variable β is defined as

$$\beta = \frac{(W_i^\dagger \cdot S)}{\sqrt{(W_i^\dagger \cdot W_i^\dagger)(S \cdot S)}}, \qquad (12)$$

where $\cdot$ denotes inner product. By thresholding β, we can determine if the watermark presents in the image. The block diagram is given in Figure 2.

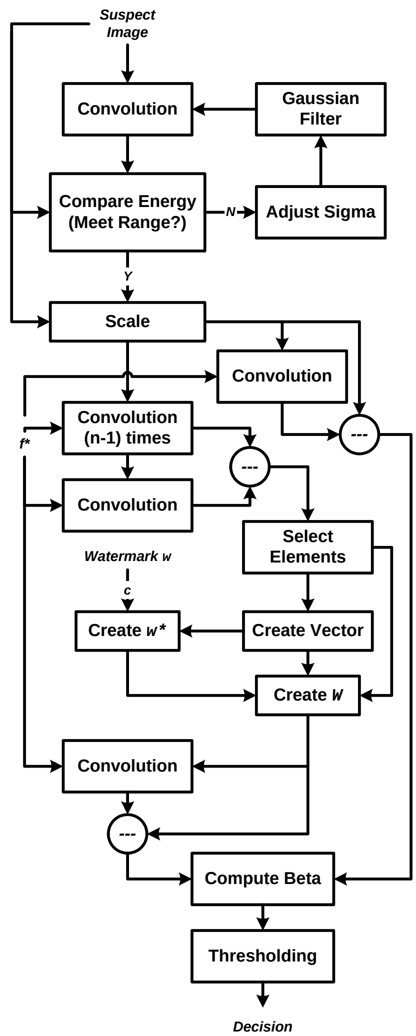

Figure 2: Block diagram of the detection process

4. EXPERIMENTAL RESULTS

The above algorithm is tested on the "Lena" image, which is a 8-bit grayscale image of size 512 by 512. The length of the watermark is 20. The energy range is [0.9499 0.9501]. Empirically, we pick σ^* to be 4, n to be 4 and c to be 2. To select the elements, we first compute the mean of the largest 2% of the values and set the threshold T to be 20% of the mean. The peak-to-peak signal-to-noise ratio (PSNR) is 47.1 dB. The original image and the watermarked images are given in Figure 3. β is -0.0013 for the original image and 0.0274 for the watermarked image. A geometrical transform that constitutes a scaling by a factor of 1.5, a rotation of 1.5^o counter-clockwise and an off-center cropping (which is equivalent to a translation and a cropping) is applied on the original image and the watermarked image. The results are given in Figure 4. In this case, β is -0.0023 for the original image and 0.0140 for the watermarked image. Tests on other images show similar results. Basically, watermarked images produce βs larger than 0.01, while original images produce both positive and negative βs with amplitudes on the order of 10^{-3}. Therefore, if we threshold β at 0.01, we can detect the presence of the watermark correctly.

5. CONCLUSIONS AND COMMENTS

In this paper, we present a RST resilient image watermarking algorithm using circular Gaussian filters. Instead of using spatial synchronization, we synchronize the watermark based on filtering results. The experimental results show the algorithm achieves robustness to rotation, scaling, translation and small amounts of cropping.

Unlike some existing algorithms [8, 9, 10], increasing the energy of the watermark does not always make it easier to detect. The detection process requires the mappings between D and $\overrightarrow{v_d}$ to be similar for the watermarked image and the original image. Such requirement will be violated when the watermark energy is too large.

The threshold of β is now determined based on observations. In our future study, we will investigate the false positive probability to set the threshold more properly. We will also test the robustness of our algorithm to compression, histogram modification, etc., and adjust the algorithm accordingly if necessary.

(a) original image

(b) watermarked image

Figure 3: Original and watermarked “Lena” image

(a) original image

(b) watermarked image

Figure 4: Original and watermarked “Lena” image after geometrical transform

6. REFERENCES

[1] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. of IEEE*, 87(7): 1079-1107, July 1999.

[2] G. Langelaar, I. Setyawan, and R. Lagendijk, "Watermarking digital image and video data," *IEEE Sig. Proc. Mag.*, 17(5): 20-46, September, 2000.

[3] J. *Ó* Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking," *Proc. ICIP 97*, pp. 536-539, October 1997.

[4] J. *Ó* Ruanaidh and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," *Signal Processing*, 66(3): 303-317, May 1998.

[5] C. Lin, *et al* "Rotation, scale and translation resilient public watermarking for images," *Proc. SPIE*, 3971(2): 90-98, January 2000.

[6] S. Pereira and T. Pun, "Robust template matching for affine resistant image watermarks," *IEEE Trans. Image Proc.*, 9(6): 1123-1129, June 2000.

[7] R. Gonzalez and R. Woods, *Digital Image Processing*, Addison-Wesley Inc., 1992.

[8] I. Cox, J. Kilian, F. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Proc.*, vol. 6, no. 12, pp. 1673-1687, Dec. 1997.

[9] W. Zeng and B. Liu, "A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images," *IEEE Trans. Image Proc.*, vol. 8, no. 11, pp. 1534-1548, Nov. 1999.

[10] Y. Wang, J. Doherty, and R. Van Dyck, "A novel wavelet-based algorithm for watermarking digital images," *Proc. CISS 2000*, vol. 1, pp. TA 5.7-12, 2000.