Using Counter-propagation Neural Network for Digital Audio Watermarking

Chuan-Yu Chang and Wen-Chih Shen
Graduate School of Computer Science and Information Engineering
National Yunlin University of Science & Technology
Douliou, Yunlin, Taiwan
chuanyu@yuntech.edu.tw

Abstract

Recently, the watermarking is an important technique to protect copyright, which allows authentic watermark to be hidden in multimedia such as digital image, video and audio. Watermarking has been developed to protect digital media illegal reproductions and modifications. Traditionally, watermarking require complex procedures to embed and to extract watermark, such as randomizing the watermark, choose positions to embed and extract the watermark, embed the randomized watermark into the original audio and extracted the watermark from the specific positions. Therefore, in this paper, we propose a scheme called Counter-propagation Neural Network (CNN) for digital audio watermarking. Different from the traditional methods, the watermark is embedded in the synapses of CNN instead of the original audio signal. The experimental results show that the proposed method has capabilities of robustness, imperceptibility and authenticity.

Keywords: Digital watermark, Counter-propagation neural network, Audio information hiding.

1. Introduction

The rapid growing of computer network and multimedia technologies makes it easier to access digital media. Therefore, it is urgent to protect copyright piracy. To tackle the copyright protects problem, many digital watermarking schemes have been proposed[1-4]. The watermarking technique embeds owner's information into digital media and provides the corresponding authentication mechanism. Therefore, the digital watermarking should has capabilities of robustness, imperceptibility and authenticity. Similarly, the audio watermarking should be inaudible or statistical invisible to prevent unauthorized detection and removal.

In general, the audio watermarking techniques can be classified into two groups, embedding watermark in time domain and embedding watermark in frequency domain, depending on the processing domain of the host audio in which the watermark is embedded. In time domain, the pre-masking and post-masking effect of time domain is used to implement audio watermarking. Lie *et al.* [8] proposed a method, which based on the relative energy relations between adjacent sample sections, to embed digital watermarks into a host audio signal in the time domain. However, their method cannot resist four types of attacks, such as the AAC compression, WMA compression, Amplitude compression and resampling. In frequency domain, Jong *et al.* [6] proposed a watermark embedding scheme accomplishes the perceptual transparency after watermark embedding by exploiting the masking effect of the human auditory system (HAS). Lee *et al.* [7] proposed a digital audio watermarking technique in the cepstrum domain. They insert a digital watermark into the cepstral components of the audio signal using a technique analogous to spread spectrum communications, hiding a narrow hand signal in a wideband channel.

Recently, neural networks, with their features of fault tolerance and potential for adaptive training, have been proposed as alternative approaches. Yang *et al.* [4] proposed an artificial neural network-based scheme for watermarking of audio signals. They use a multi-layer perceptron (MLP) to estimate the watermark scaling factor (WSF) intelligently from the knowledge of host audio signal. Despite that, all the mentioned methods [4,6,7,8] were lack of robustness in audio watermarking.

In addition, the traditional methods need complex embedding and corresponding extraction procedures. Therefore, in this paper we proposed a specific counter-propagation neural network (CNN) for audio watermarking. Different from the traditional methods, the watermark is embedded in the synapses of CNN instead of the original audio signal. Therefore, the quality of the watermarked audio is almost same as the original audio signal. In addition, because of the watermark is stored in the synapses, most of the attacks are not degrade the quality of the extracted watermark image. The CNN substitutes the complex embedding and corresponding extraction procedure. The experimental results show that the proposed CNN does not need complex embedding and corresponding extraction procedure. Furthermore, the proposed method has capabilities of robustness, imperceptibility and authenticity.

The remainder of this paper is organized as follows. Section 2 shows the preprocessing of

original audio signal. The architecture of the CNN and the algorithm of embedding, and extracting method are shown in Section 3. Section 4 summarizes the experimental results. Finally, conclusions are given in the Section 5.

2. Preprocessing of original audio signal

In order to resist the maliciously attacks, the original audio signal is firstly segmented into several non-overlapping frames with frame size of *N*. The synchronization code sequence is initialized the starting point position [8]. This assumption will be violated when an attacker tries to crop the signal or insert redundancy in the front end. The audio energy is using to evaluate the adequate frame to embed watermark, the energy of each frame is calculated as:

$$E(k) = \frac{1}{n}\sum_{i=1}^{n} |s(i)|, k = 1,2,..., N \quad (1)$$

where $E(k)$ denotes the energy of frame *k*, and the $s(i)$ represents the magnitude of audio signal and *n* is the number of frame size. The candidate frames set $X(k)$ is obtained by threshold of energy. The threshold is to search the position of the non-quiet sound segment, and suitable embedding the watermark.

The quality of the extracted watermark is highly depending on the preprocessing procedure. Figure 1 shows the schematic diagram of the preprocessing. After the preprocessing, each candidate frame data *X* can be represented as a vector form:

$$X = \{x_1, x_2, ..., x_n\} \quad (2)$$

where *n* is the length of candidate frames.

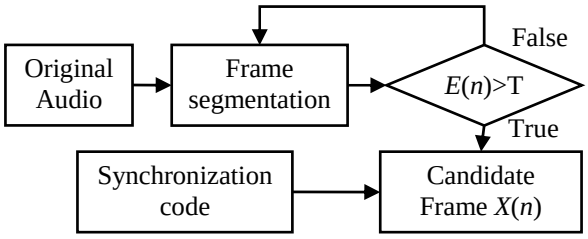


Figure 1.The schematic diagram of the preprocessing.

3. Counter-propagation neural network for audio watermarking

The traditional watermarking methods require complex embedding and corresponding extraction procedures. In this paper, a counter-propagation neural network-based (CNN) audio watermarking method is proposed. The proposed watermarking method integrated the embedding and extraction procedure into a counter-propagation neural network. Figure 2 shows the architecture of the CNN. The CNN is consists of *n* input neurons to represent the length of candidate frame data, *N* hidden neurons to represent the total pixels of the watermark image and *m* output neurons to represent the gray value of the watermark image. In order to ensure that the proposed CNN has capability of embedding and extracting watermarks, the candidate frame data and the corresponding watermark are used to train the CNN. After the network's evolution, the watermark is embedded into the synapses between the hidden layer and output layer.

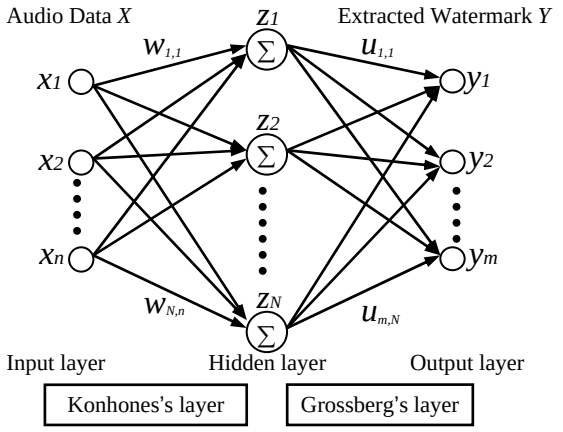


Figure 2. The architecture of the counter-propagation neural network for watermarking.

There are two sets of weights that are adjusted with two different learning algorithms, the Konhones's self-organizing learning and the Grossberg's supervised learning. In the first stage, weights *W* connecting the candidate frame data *X* and the hidden layer are trained using Kohonen's self-organizing learning rule, and achieve the goal to classify training pattern to hidden layer. In the second stage, the weights between the hidden layer and the output layer are trained using Grossberg's learning algorithm.

The input candidate frame vector *X* is fully connected to the neurons in hidden layer with weights *W*.

$$W = \{w_{1,1}, w_{1,2}, ..., w_{N,n}\} \quad (3)$$

where $w_{j,i}$ denotes the weight between *j*-th neuron and input x_i, $i = 1,2,...,n$. Accordingly, the total input of the *j*-th neuron is defined as:

$$Z_j = \left[\sum_{i=1}^{n} (x_i - w_{j,i})^2\right]^{1/2}, j = 1,2,..., N \quad (4)$$

which represents the total distance between the input *X* and the weights of the *j*-th hidden neuron. After the distances are computed, a winner-takes-all mechanism is applied to the neurons in the hidden layer to find the winner neuron unit to update weights in the konhones's layer [5]. The winner of the hidden neuron H^* is defined as:

$$H^* = \min Z_j, j = 1,2,..., N \quad (5)$$

accordingly, the weights between the input layer and winner neuron in hidden layer is update by:

$$w_{H^*,i}(k+1) = w_{H^*,i}(k) + \eta(k) \times \left(x_i - w_{H^*,i}(k)\right) \quad (6)$$

where $\eta(k)$ is learning rate. In addition, the learning rate $\eta(k)$ is decreased gradually during the training epoch *k*. The learning function $\eta(k)$ can be specified as follows:

$$\eta(k) = \eta(0) \exp\left(-\frac{k}{k_0}\right) \quad (7)$$

where $\eta(0)$ is initial learning rate, k_0 is a positive constant.

The output of the CNN can be represented as vector form:

$$Y = \{y_1, y_2, ..., y_m\} \quad (8)$$

where *m* is the bit length of watermark image in binary representation. Each output neuron receives all the output of hidden neurons with weights *U*.

$$U = \{u_{1,1}, u_{1,2}, ..., u_{m,N}\} \quad (9)$$

where $u_{l,j}$ denotes the weight between the *j*-th neuron in hidden layer and the *l*-th output neuron. Thus, the output of the *i*-th neuron in the output layer is obtained as

$$y_i = \sum_{j=1}^{N} u_{i,j} Z_j \quad (10)$$

Since the WTA, only the wining neuron H^* in the hidden layer has output value one. Thus, the output of the *i*-th output neuron can be simplied as:

$$y_i = u_{i,H^*} \quad (11)$$

Therefore, only the weights connecting the winning neuron in the hidden layer with the neurons in the output layer will be updated. Accordingly, the weights in the output layer of the CNN are updated as:

$$u_{l,H^*}(k+1) = u_{l,H^*} + \eta(k) \times \left(desire_l - y_l(k)\right) \quad (12)$$

where $desire_l$ is pixel value of the watermark image. The learning function $\eta(k)$ is calculated by Eq (7). In addition, the instantaneous output error ξ_q is defined as:

$$\xi_q = \sum_{l=1}^{m} \left| y_l - desire_l \right| \quad (13)$$

Thus, the total error of the CNN is defined as

$$\xi_{total} = \sum_{q=1}^{N} \xi_q \quad (14)$$

where *N* denotes the total pixels of the watermark image.

3.1 Embedding algorithm

The watermark embedding approach is summarized as follows:

Input: The candidate frame data *X*.

Output: The watermark image will be embedded into the synapse (*W*, *U*) of the converged CNN.

Step 1.Arbitrarily assigns the initial weights to *W* and *U*.

Step 2.Use Eq. (4) to calculate the output of each hidden neuron.

Step 3.Use Eq. (5) to find the winner neuron.

Step 4.Update the weights *W* according to Eq. (6).

Step 5.Use Eq. (12) to update weight *U*.

Step 6.Use Eq. (14) to calculate the output error of CNN. If ξ_{total} is less than a predefined threshold value, stop training. Otherwise, go to Step 2.

3.2 Extracting algorithm

To extract watermark from the trained CNN described in Section 3.1, the watermark extracting approach is summarized as follows:

Input: The candidate frame data *X*.

Output: watermarked image *Y*.

Step 1.Use Eq. (4), Eq. (5) to calculate the output of the winner neuron in hidden layer.

Step 2.Use Eq.(8) and Eq.(11) to obtain the gray-level (in binary) of the watermark image *Y*.

4. Experimental results

To show the proposed CNN has good capability for audio watermarking, four experiments are presented to demonstrate robustness, imperceptibility and authenticity. In our experiments, 16bits mono-track audio music with sampling rate 44.1 kHz was used for simulation. We selected three different types of song from the database randomly. Figures 3(a-c) show the original audio wave of female singer, male singer and chorus, respectively. Figures 4 (a-b) show the 32×32 binary rabbit image and the 32×32 grayscale YUNTECH watermark image, respectively. In order to evaluate the robustness of the Lie's method [8] and the proposed CNN method, two public domain software: GoldWave and ImTOO are used to attack the watermarked audios. Eight types of attack are performed to attack the watermarked audios:

- Type 1: MP3 compression, compress audio signal to 128k bit/s (wav->mp3->wav).
- Type 2: AAC compression, compress audio signal to 128k bit/s (wav->AAC->wav), MPEG-2.
- Type 3: WMA compression, compress audio signal to 128k bit/s (wav->wma->wav).
- Type 4: Amplitude compression, multiplier 6 dB
- Type 5: Smoother filter.
- Type 6: Resampling (44kHz->22kHz ->44kHz).
- Type 7: Remove noise (clearly audible).
- Type 8: Silence reduction, reduction -48 dB.

The Normalized Correlation (NC) is used to evaluate the similarity measurement of extracted

binary watermark, which can be represented as:

$$NC = \frac{\sum_i \sum_j P(i,j)P'(i,j)}{\sum_i \sum_j [P(i,j)]^2} \quad (15)$$

where $P(i,j)$ and $P'(i,j)$ represent the bit value of (i,j)-th pixel of original watermark and extracted watermark image, respectively.

The Peak Signal to Noise Ratio (PSNR) is used to evaluate the quality of watermarked audio, which can be represented as:

$$PSNR\ (db) = 10 \log_{10} \frac{X^2_{peak}}{\sigma_e^2} \quad (16)$$

where σ_e^2 is defined as:

$$\sigma_e^2 = \left(\frac{1}{M}\right)\sum_{i=1}^{M}\left(X(i) - Z(i)\right)^2 \quad (17)$$

where M is the length of the host audio, $X(i)$ is the magnitude of host audio at time *i*. Similarly, $Z(i)$ denotes the magnitude of watermarked audio at time *i*. X^2_{peak} denotes the squared peak value of host audio. The higher PSNR means that the watermarked audio is more similar to the original audio.

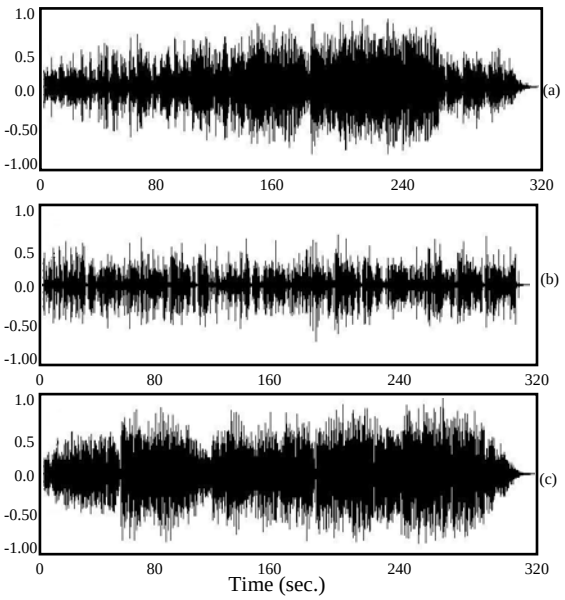


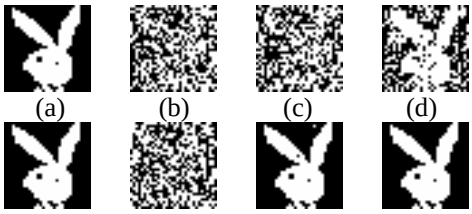
Figure 3. Original Audio Signals, (a) female singer, (b) male singer, (a) chorus singer.

(a) (b)

Figure 4. Watermark images, (a) binary Rabbit image (b) gray scale YUNTECH logo image.

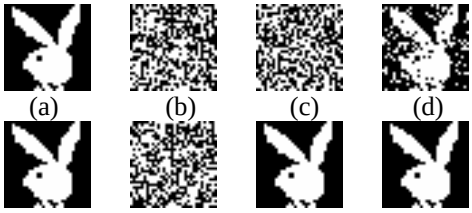
4.1 Experiment 1: Robust testing for Lie's method

In this experiment, the watermark image is a 32×32 binary rabbit image and the size of audio frame *N* is 300. The initial d' of the Lie's method [8] is set to 0.05. The binary rabbit watermark is embedded into three original audios. In order to evaluate the robustness of Lie's method, the eight kinds of attacks were applied to the watermarked audios. Figures 5-7 show the extracted watermarks of the female singer, male singer, and chorus singer that suffered from eight different type attacks. Obviously, Lie's method cannot extract the complete watermark from various attacks, such as AAC, WMA, Amplitude compression, and resampling result in messy pattern. In other words, the hided watermark has been destroyed by various attacks.



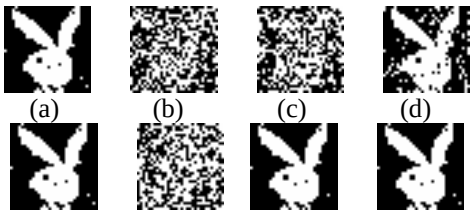
(a) (b) (c) (d)
(e) (f) (g) (h)

Figure 5. The extracted watermarks of the watermarked audio by female singer.



(a) (b) (c) (d)
(e) (f) (g) (h)

Figure 6. The extracted watermarks of the watermarked audio by male singer.



(a) (b) (c) (d)
(e) (f) (g) (h)

Figure 7. The extracted watermarks of watermarked audio chorus singer.

Table 1. NC and bit error of extracted watermark from different type of attack with watermarked audio by Lie's method

Type	NC of extracted watermark			bit error(32x32 bit)		
	female	male	chorus	female	male	chorus
1	0.994	1	0.989	2	0	9
2	0.61	0.592	0.623	447	455	388
3	0.626	0.542	0.631	440	482	398
4	0.939	0.934	0.957	260	136	104
5	0.994	1	0.989	2	0	9
6	0.56	0.56	0.55	479	485	513
7	0.992	1	0.989	4	0	8
8	0.994	1	0.989	2	0	9

Table 1 shows the NC values and the bit error of

the extracted watermarks. From Table 1, the NC values are between 0.54 to 0.93. The low NC values indicate that the Lie's method was unable to resist type (2,3,4,6) attacks.

4.2 Experiment 2: Robust testing for proposed method

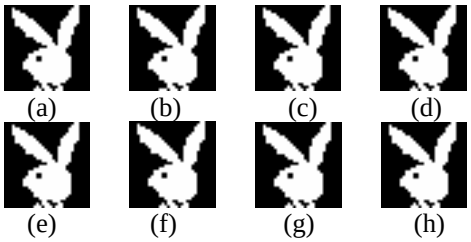


Figure 8. The extracted watermarks of watermarked audio female singer.

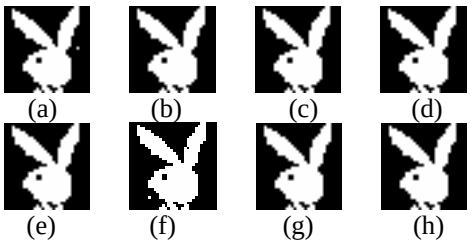


Figure 9. The extracted watermarks of watermarked audio male singer.

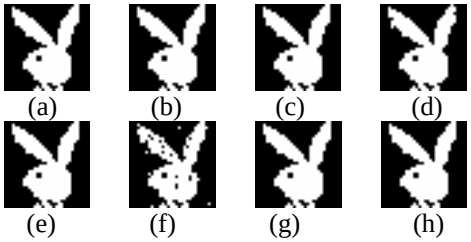


Figure 10. The extracted watermarks of watermarked audio chorus singer.

Table 2. NC and bit error of extracted watermark from different type of attack with watermarked audio by proposed CNN method

Type	NC of extracted watermark			bit error(32x32 bit)		
	female	male	chorus	female	male	chorus
1	1	1	1	0	1	0
2	0.997	1	1	1	0	0
3	1	1	1	0	0	0
4	1	1	0.992	0	0	4
5	1	1	1	0	0	0
6	1	0.997	0.95	0	2	27
7	1	1	1	0	0	0
8	1	1	1	0	0	0

In this evaluation, the initial learning rate η is set to 1.35. The number of neuron is 32×32 (size of watermark image). The threshold value of 0.008 was established to terminate training. In order to evaluate the robustness of proposed method, eight kinds of attacks are applied to degrade the watermarked audio. Figures 8-10 show the extracted watermarks of watermarked audio that suffered from the same attacks as the previous experiment. Obviously, all the watermarks are extracted completely by using CNN. Table 2 shows the NC value and the bit error of extracted watermarks. It indicates the proposed method is robust and being able to resist eight attacks.

4.3 Experiment 3: Imperceptibility for proposed method

To show the imperceptibility of the proposed method, a binary rabbit watermark and a gray scale Yuntech logo image are embedded into the three songs, respectively. However, only a binary rabbit watermark is embedded into the three songs in Lie's method. Table 3 shows the PSNR values of the watermarked audios by Lie's and the proposed method. The average PSNR value of the Lie's method is about 31dB. On the other hand, the PSNR values of the proposed method are all infinite. The infinite PSNR values illustrate that the watermarked audio signal is the same as the original audios. This is because of the watermark is embedded in the synapses of CNN instead of the original audio signal; therefore, the quality of the watermarked audio is almost same as the original audio signal. It indicates the proposed method is able to imperceptibility. Figure 11(a-c) shows the extracted corresponding watermarks. These figures show that the proposed method is capable of embedding/extracting gray level watermark into/from audio.

Table 3. PSNR of watermarked audio by Lie's method and the proposed method

	female	male	chorus
Watermarked binary image by Lie's method	32.88dB	30.19dB	33.14dB
Watermarked binary image by proposed method	∞	∞	∞
Watermarked graylevel image by proposed method	∞	∞	∞

Figure 11. The extracted watermarks from attack-free female, male and chorus singer, respectively.

4.4 Experiment 4: Authenticity testing

In this experiment, we use raw audio, audio without embedded watermark, to test authenticity of the proposed CNN and Lie's method. Figure 12(a-c) and Figure 13(a-c) show the extracted watermarks

from Lie's and the proposed method, respectively. Obviously, both Fig. 12 and Fig. 13 are not legal watermark images. These experimental results show that the proposed CNN can not extract legal watermark from the raw original audio. Therefore, the proposed method has ability to extract corresponding watermark from watermarked audios, but not from unmarked audios.

(a) (b) (c)

Figure 12. The extracted watermarks of Fig. 12(a-c) for Lie's method.

(a) (b) (c)

Figure 13. The extracted watermarks of Fig. 13(a-c) for proposed method.

5. Conclusions

In this paper, we proposed a counter-propagation neural network (CNN) for audio watermarking. Different from the traditional methods, the watermark is embedded in the synapses of CNN instead of the original audio. Therefore, the quality of the watermarked audio is almost same as the original audio. The proposed CNN does not need the original audio to extract the watermark. In addition, because of the watermark is stored in the synapses, most of the attacks could not degrade the quality of the extracted watermark image. Thus, the proposed CNN has capability to resist various attacks. The watermark embedding procedure and extracting procedure is integrated into the proposed CNN. Therefore, the proposed method is simple than traditional methods. The experimental results show that the proposed method has good capabilities of robustness, imperceptibility and authenticity.

Acknowledgement

This work was supported by the National Science Council, Taiwan, R.O.C. under Grants NSC 92-2213-E-224-041.

References

[1] Arttameeyanant, P., Kumhom, P., Chamnongthai, K., "Audio watermarking for Internet," *Proc. of the IEEE Int. Conf. on Indust. Tech.*, Vol 2, pp. 976 - 979, 2002.

[2] Chuan-Yu Chang, Sheng-Jyun Su, "The application of a full counterpropagation neural network to image watermarking," *Proc. of the IEEE Int. Conf. on Networking, Sensing and Control*, pp.993-998, 2005.

[3] Cvejic, N, Seppanen, T., "Increasing robustness of LSB audio steganography using a novel embedding method," *Pro. of the IEEE Int. Conf. on Infor. Tech.: Coding and Computing*, Vol 2, pp.533 - 537, .2004.

[4] Huijuan Yang, Patra, J.C., Chan, C.W., "An artificial neural network-based scheme for robust watermarking of audio signals, "*Proc. of the IEEE Int.l Conf. on Acou., Spe., and Sig.*, Vol 1, pp. 1029 -1032, 2002.

[5] Fredric M. Ham and Ivica Kostanic, *Principles of Neurocomputing for Science & Engineering*, McGraw-Hill, Singapore, 2001.

[6] Jong Won Seok and Jin Woo Hong, "Audio watermarking for copyright protection of digital audio data," *Elec. Let.*, Vol 37, pp. 60-61, 2001.

[7] Sang-Kwang Lee and Yo-Sung Ho, "Digital Audio Watermarking in the Cepstrum Domain," *Proc. of the IEEE Trans. on Con. Elect.*, Vol 46, pp. 744 - 750, Aug. 2000.

[8] Wen-Nung Lie and Li-Chun Chang, "Robust and high-quality time-domain audio watermarking subject to psychoacoustic masking," *Proc.of the IEEE int. Symp. on Circ. and Sys.*, pp.45-48, 2001.