# The Structure of Reversible DTCNN (Discrete-Time Celluar Neural Networks) for Digital Image Copyright Labeling

Gye-Ho Lee* · Seung-jo Han*

# 디지털영상의 저작권보호 라벨링을 위한 Reversible DTCNN(Discrete-Time Cellular Neural Network) 구조

이계호* · 한승조*

This study was supported by research funds from Chosun University, 2000

ABSTRACT

In this paper, we proposed structure of a reversible discrete-time cellular neural network (DTCNN) for labeling digital images to protect copylight. First, we present the concept and the structure of reversible DTCNN, which can be used to generate 2D binary pseudo-random images sequences. We presented some, output examples of different kinds of reversible DTCNNs to show their complex behaviors. Then both the original image and the copyright label, which is often another binary image, are used to generate a binary random key image.

The key image is then used to scramble the original image. Since the reversibility of a reversible DTCNN, the same reversible DTCNN can recover the copyright label from a labeled image. Due to the high speed of a DTCNN chip, our method can be used to label image sequences, e.g., video sequences, in real time. Computer simulation results are presented.

요 약

본 논문은 저작권보호를 위해 디지털영상의 라벨링을 위한 reversible DTCNN(discrete-time cellular neural network) 구조를 제안한다. 이러한 저작권보호 라벨링을 위해서 2차원 이진 pseudo 랜덤 영상열에 사용할 수 있는 새로운 reversible DTCNN의 구조와 개념을 설명하고 이에 대한 복잡행위를 보여주기 위해 reversible DTCNN의 서로 다른 방법들의 예시를 들어 설명한다. 또한 서로 다른 2진영상인 원영상과 복사된 영상은 서로 다른 2진 랜덤 영상키를 사용한다.

이 영상키는 원영상을 스크램블하는데 사용된다. 따라서 reversible DTCNN를 다시 역변환시켜서 저작권보호가 라벨링된 영상으로부터 복사된 영상임을 찾아낼 수 있다. 그러나 이러한 동영상을 처리하는 데는 S/W에서는 많은 시간이 소요되므로 고속 DTCNN 칩을 사용하여 실시간에서 동영상이나 비디오영상을 저작권보호를 위한 라벨링에 사용할 수 있으며, 이러한 결과를 컴퓨터에서 시뮬레이션됨을 보인다.

키워드

discrete-time cellular neural network (DTCNN), copyright label, key image

*Korea Telecom, **Chosun University, Divison of Electronics&Communication Information Eng.

접수일자 : 2003. 5. 24

## I. Introduction

The distribution and use of digital video and electronic images is rapidly increasing with the development of multimedia, common database of worldwide computer networks, electronic publication systems and electronic mail systems[1]. This development provides many advantages such as fast and easy access and inexpensive duplication of products. But it also increases the potential for misuse and theft of these information[6, 7]. It is because 1) making copies of electronic image is easy and inexpensive; 2) each copy is exactly identical to the original; and 3) distribution of the copies via network or floppy is easy, fast and uncontrollable. For this reason, creators of multimedia information are seeking technical solutions to the problems associated to copyright protection of image data in a multimedia environment. In addition to copyright labeling of broadcast images, applications to image labeling techniques include copyright and/or secure labeling of electronic publishing, facsimiles, scientific imaging and medical imaging.

A promising solution to this problem is to add to each image a signature, or "water mark", which identifies its copyright holder[8-11] and has the following properties:

1. Signature should declare the copyright holder
2. Signature must contain a user code, which verifies the user is in legal possession of the data;
3. The image data is labeled in a manner which allows its distribution to be tracked.

A signature should satisfy the following conditions[8-11]:

1. Not inducing change of the visual perception of the image.
2. Not perceptually or statistically detectable by any one except those who have precise knowledge of the label generation system.
3. Big key space of the label generation system.
4. Resistant to image transformation method that do not impair the quality of the image considerably, such as standard compression method, or other data processing such as smoothing and pirates attacks.

Different kinds of methods had proposed to satisfy the above conditions. The method proposed in [8, 9, 10, 12] basically consist of two stages. In the first stage, locations where the signature bits will be embedded are chosen secretly. In the second stage, a scheme is used to change the value at every location. The authors of [11] proposed a method based on the synchronization of inverse nonlinear dynamical systems, which was developed for the transmission of information hidden in a chaotic carrier signal[13]. This method does not need location choice because the signature is embedded in the whole image.

All the above methods are based on the serial mechanics, in case of real time image sequences transmission, such as digital video image sequences, there should be a tradeoff between the speed and size of key-space. In this paper, we proposed a method based on reversible discrete-time cellular neural networks(RDTCNN) to provide a very hight speed and very large key space for invisible signature in images. First, a RDTCNN generates a random binary key image using both original image and the copyright label, which is often a binary image. The key image is then used to scramble the original image using an invertible mapping. Since the reversibility of a RDTCNN, the same RDTCNN can restored the copyright label from a labeled image.

In section 2, we present the concept and the structure of RDTCNN. In section 3 we present some output examples of different kinds of RDTCNNs to show their complex behaviors. In section 4, we present the structure and the scheme for labeling images using RDTCNN. In section 5, the simulation results are given. In section 6, conclusions are contained.

## II. Generate 2D random sequences using Discrete-Time Cellular Neural Networks

Consider an $M \times N$ DTCNN[4], having $M \times N$ cells arranged in $M$ rows and $N$ columns. We denote the cell on the *ith* row and the *jth* column as cell $C_{ij}$. The r-neighborhood of $C_{ij}$ is defined by [5]

Definition 1: r-neighborhood

The r-neighborhood of a cell $C_{ij}$, in an $M \times N$ DTCNN is defined by

$$N_r(i,j) = \{C_{kl} | max(|k-i|, |l-j|) \leqq r, 1 \leqq k \leqq M; 1 \leqq l \leqq N\} \quad (1)$$

where $r$ is a positive integer number.

We use the following set of equations to define the cell $C_{ij}$ :

1. State equation

$$x_{ij}(t) = \sum_{C_{kl} \in N_r(i,j)} A(i,j;k,l) y_{kl}(t) + \sum_{C_{kl} \in N_r(i,j)} B(i,j;k,l) u_{kl} + I$$

(2)

$$, 1 \leqq i \leqq M, 1 \leqq j \leqq N$$

2. Output equation

$$y_{ij}(t) = f(x_{ij}(t-1)), 1 \leqq i \leqq M, 1 \leqq j \leqq N \quad (3)$$

3. Initial state

$$x_{ij}(0), 1 \leqq i \leqq M, 1 \leqq j \leqq N \quad (4)$$

$x_{ij}(t) \in R, y_{ij}(t) \in and u_{ij} \in \{0,1\}$ are state, output and input of the cell $C_{ij}$, respectively. $I \in R$ is the bias. $t \in Z$ is the discrete time. $A(i,j;k,l) and B(i,j;k,l)$ are feedback and feed-forward synaptic weights, respectively. F() is the nonlinear output property of a cell, it should be easy to be implemented by using VLSI techniques.

For simplicity, we can rewrite Eq.(2) into the following form[5]

$$x_{ij}(t) = A * y_{ij}(t) + B * u_{ij} + I \quad (5)$$

where "*" denotes a 2D convolution operator. $A$ and $B$ are feedback and feed-forward templates, respectively.

Then we consider the following DTCNN defined by

1. State equation

$$x_{ij}(t) = A^* y_{ij}(t), 1 \leq i \leq M, 1 \leq j \leq N \quad (6)$$

2. Output equation

$$y_{ij}(t+1) = f(A^* y_{ij}(t)) \oplus y_{ij}(t-1), 1 \leq i \leq M, 1 \leq j \leq N \quad (7)$$

where $\oplus$ is the exclusive OR.

3. Cell Nonlinearity

$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (8)$$

4. Initial conditions

$$y_{ij}(-1), y_{ij}(0), \ 1 \leq i \leq M, 1 \leq j \leq N \quad (9)$$

We can see that the nonlinearity of a cell is different from the original one as in [4], we used way to guarantee that the outputs of DTCNNs can be represented by 0-1 logic. The output equation is also different from that in the original paper[4] and needs a 2-bits local digital memory for storing the previous outputs. Since XOR is reversible, from Eq. (7) we have

$$y_{ij}(t-1) = f(A^* y_{ij}(t)) \oplus y_{ij}(t+1),$$
$$1 \leq i \leq M, 1 \leq j \leq N \quad (10)$$

Which means that if we know the output $y_{ij}(t+1)$ at iteration $t+1$ and the output $y_{ij}(t)$ at iteration $t$ then we can find output $y_{ij}(t-1)$ at iteration $t-1$. Thus, this kind of DTCNN is reversible. We call the DTCNN defined by Eqs. (6)-(9) as a *reversible DTCNN (RDTCNN).*

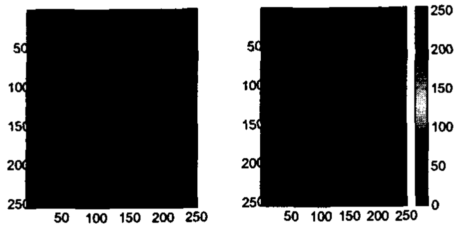
## III. Output of RDTCNNS

Many 2D sequences generated by the RDTCNN as in Eqs. (6)-(9) are effectively random. This is verified by using the statistical test provided in [1, 2, 3]. In this section we will present some typical examples of the outputs of different RDTCNNs.

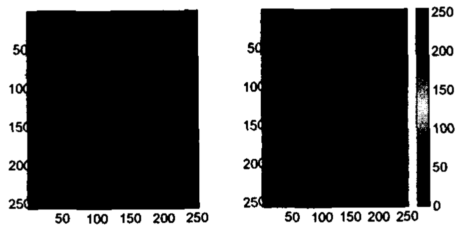
We first consider the RDTCNN with $A$ template

$$A = \begin{pmatrix} -1 & -1 & 1 \\ 1 & 1 & 0 \\ 1 & -1 & -1 \end{pmatrix} \quad (11)$$

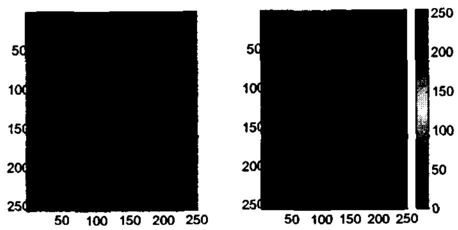
Observe that this template has zero average value. The simulation results are shown in Fig. 1.Figure 1(a) shows two initial outputs of the RDTCNN. Since there is a one-bit binary memory for each cell, it is convenient to represent the memorized output and the current output in the same figure to give the reader a comprehensive impression of the output of this RDTCNN, In each figure, the left and the right subfigures represent the memorized and current outputs of the RDTCNN, respectively. form Figure 1(b)-(j) we can see that the outputs of the RDTCNN becomes very random-like rapidly conditions, we can simply recover the two images shown in Fig. 1(a) by running the same RDTCNN along the reversible direction 100 iterations.



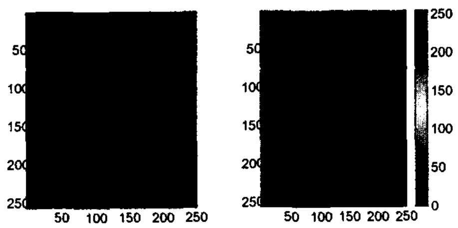
(a)



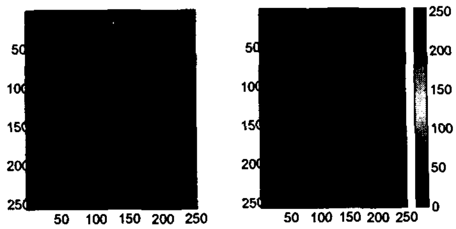
(b)



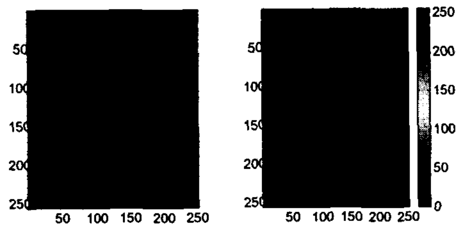
(c)



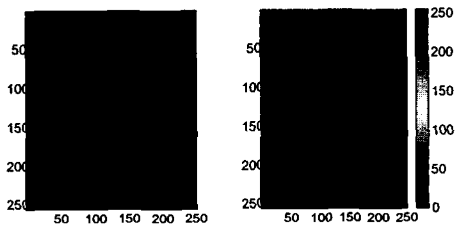
(d)



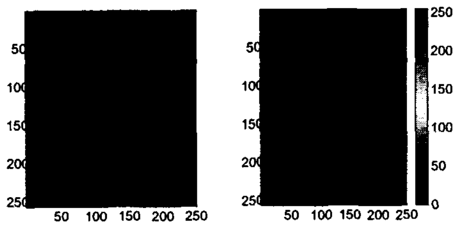
(e)



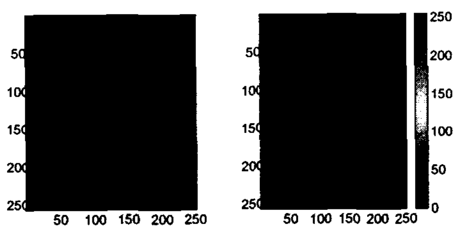
(f)



(g)



(h)



(i)

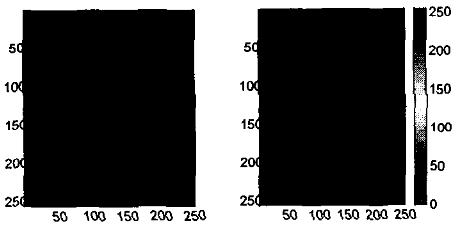
Fig. 1 (Continued)

Figure 1: Typical output of the RDTCNN with parameter given by (11). (a) Initial conditions $y_{ij}(-1)$ and $y_{ij}(0)$. (b) Iterations 3 and 4. Iterations 5 and 6. (d) Iterations 11 and 12. (e) Iterations 19 and 20. (f) Iterations 29 and 30. (g) Iterations 39 and 40. (h) Iterations 49 and 50. (i) Iterations 99 and 100.

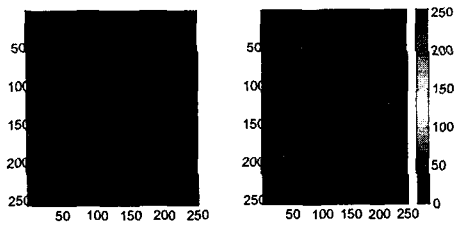
We then consider the RDTCNN with $A$ template

$$A=\begin{pmatrix} 1 & 1 & 0 \\ -1 & -1 & 1 \\ 1 & -1 & -1 \end{pmatrix} \tag{12}$$

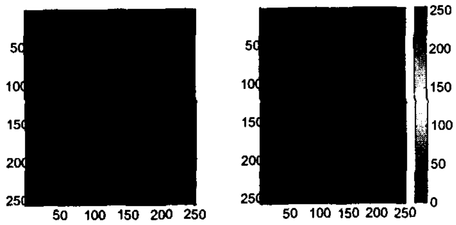
Observe that this template has zero average value. the simulation results are shown in Fig. 2.



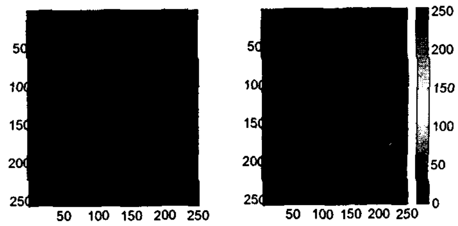

(a)



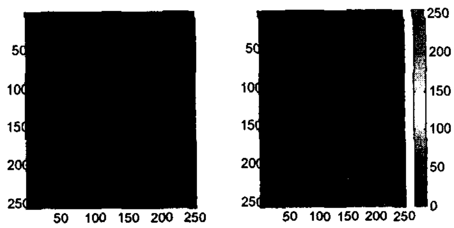

(b)



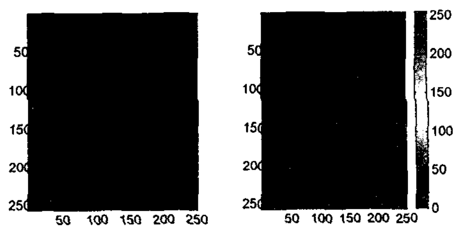

(c)



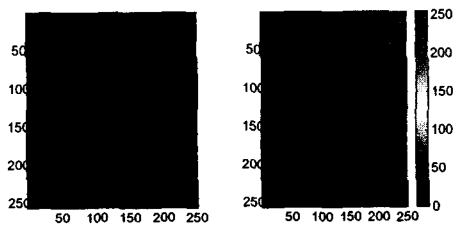

(d)




(e)

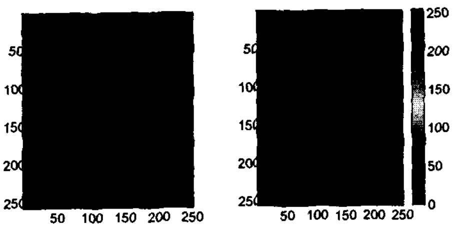



(f)

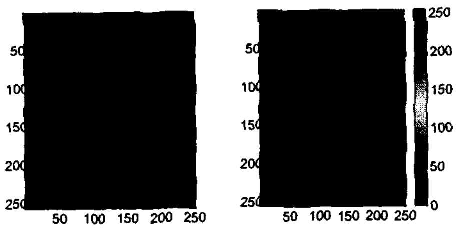



(g)

Fig. 2(*Continued*)




(h)




(i)

Fig. 2

Figure 2: Typical output of the RDTCNN with parameter given by (12). (a) Iterations 3 and 4. (b) Iterations 5 and 6. (c) Iterations 11 and 12. (d) Iterations 19 and 20. (e) Iterations 29 and 30. (f) Iterations 39 and 40. (h) Iterations 69 and 70. *(i) Iterations 99 and 100.*

We then consider the RDTCNN with $A$ template

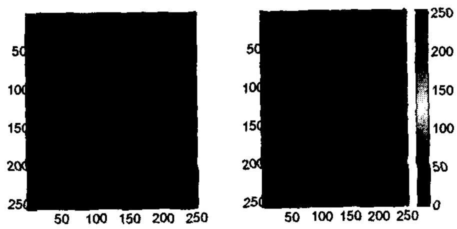
$$A=\begin{pmatrix} -2 & 1 & 1 \\ 1 & -1.7 & 1 \\ 1 & 1 & -2.3 \end{pmatrix} \tag{13}$$

Observe that this template has zero average value and the diagonal elements are negative and asymmetric. The simulation results are shown in Fig. 3.

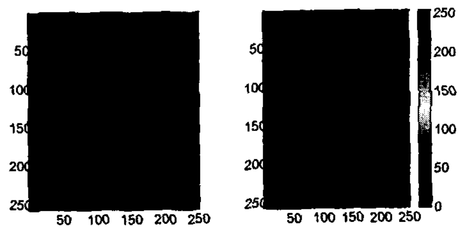
We then consider the RDTCNN with $A$ template

$$A=\begin{pmatrix} 1.5 & -1.3 & 1.7 \\ -1.7 & 0 & -1.5 \\ 1.9 & -1.9 & 1.3 \end{pmatrix} \tag{14}$$

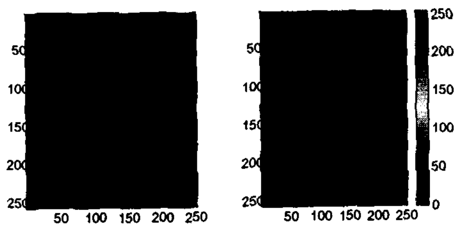
Observe that this template has zero average value and there are 4 mutually negative pairs of elements. The simulation results are shown in Fig. 4.



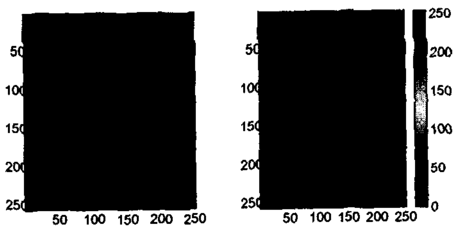

(a)




(b)



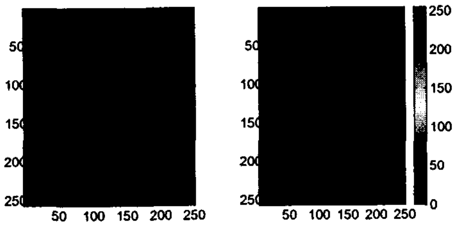

(c)



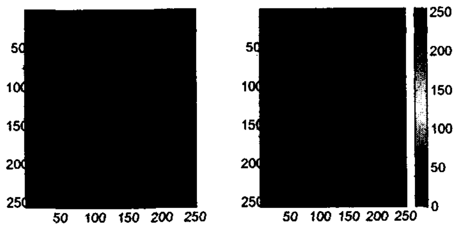

(d)

Fig. 3(*Continued*)

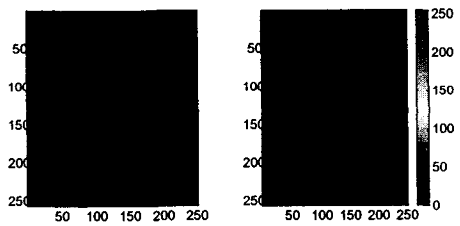
Figure 3: Typical output of the RDTCNN with parameter given by (13). (a) Iterations 3 and 4. (b) Iterations 5 and 6. (c) Iterations 11 and 12. (d) Iterations 19 and 20. (e) Iterations 29 and 30. (f) Iterations 39 and 40. (g) Iterations 49 and 50. (h) Iterations 69 and 70. (i) Iterations 99 and 100.



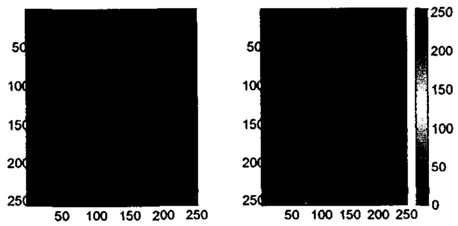

(e)



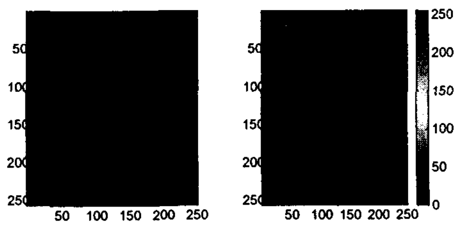

(f)




(g)

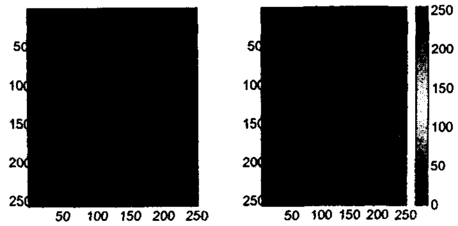



(h)

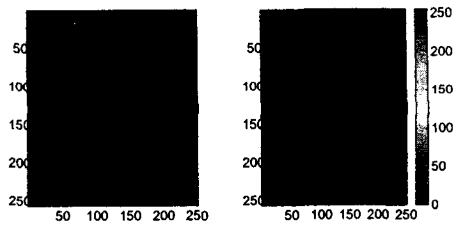



(i)

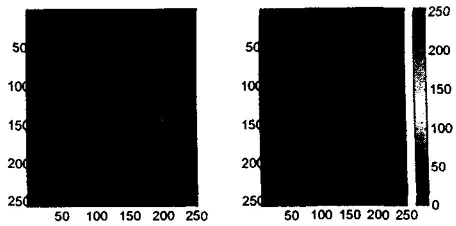
Fig. 3



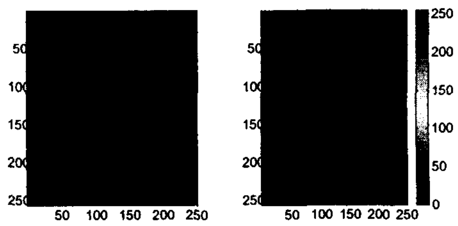

(a)




(b)




(c)




(d)

Fig. 4(*Continued*)

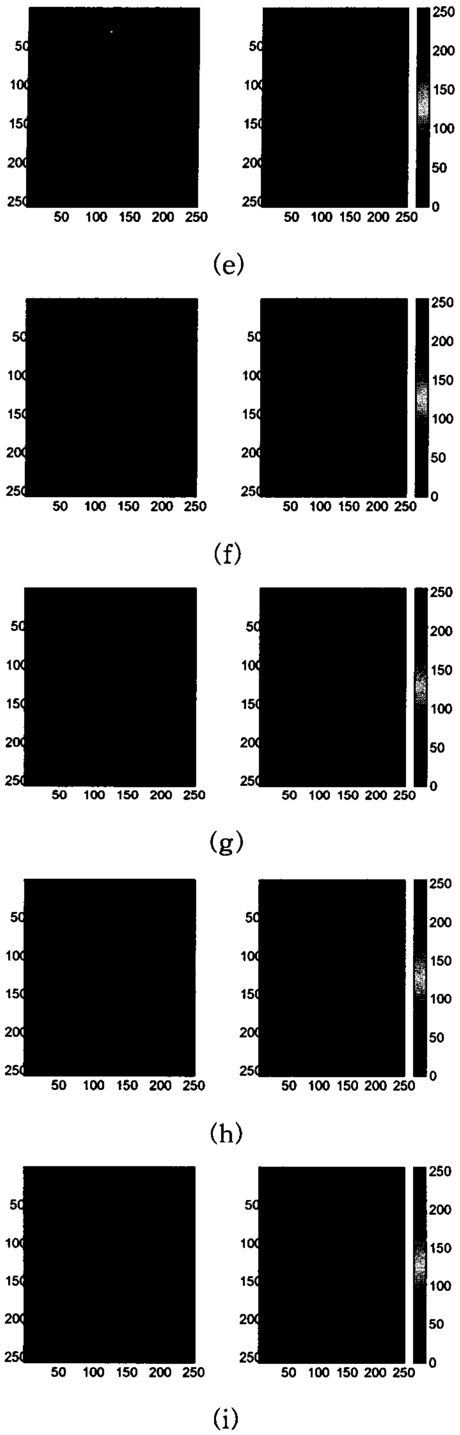



Fig. 4

Figure 4: Typical output of the RDTCNN with parameter ginven by (14). (a) Iterations 3 and 4. (c) Iterations 11 and 12. (d) Iterations 19 and 20. (e) Iterations 29 and 30. (f) Iterations 39 and 40. (g) Iterations 49 and 50. (h) Iterations 69 and 70. (i) Iterations 99 and 100.

## IV. Image Copyright Labeling Using Reversible Discrete-Time Cellular Neural Networks

The block diagram of image copyright labeling is shown in Fig. 5. In Fig. 5(a), the copyright label $X$ is a binary image of the same size of the original image $Y$. In the case when $X$ is smaller than $Y$, we can choose a part of $Y$ of the same size of $X$ secretly as the original image. In this paper, without loss of generality, we only consider the case when $X$ and $Y$ have the same size. Suppose every pixel of $Y$ is represented by $n(\geqq 8)$ bits. We choose the $[\frac{n}{2}]$th (where $[x]$ denotes the biggest integer less than $x$) bit of every pixel of $Y$ as an initial condition of RDTCNN, $\{y_{ij}(0)\}$. And the other initial condition $\{y_{ij}(-1)\}$ is chosen as the copyright label $X$. Using this two initial conditions, the RDTCNN in Eqs. (6)-(9) runs an iteration to generate the key image $K=\{y_{ij}(1)\}$. Then the labeled image $Z$ can be given by

$$Z_{ij} = F(Y_{ij}, K_{ij}) \tag{15}$$

Since $|Z_{ij} - Y_{ij}| \leqq 1$ should be satisfied for purpose of invisible image copyright labeling, we can simply $F(\cdot)$ as shown in Fig. 5. The $F(\cdot)$ in Fig. 6 can be implemented by replacing the least significant bit(LSB) of every pixel in image $Z$ with the corresponding pixel value in image $K$. This operation can be implemented parallel using CNN universal machine[14]. The key space of this scheme is all the parameters of the RDTCNN. Since the templates are real number, the key space is very big.

Then we consider how to extract the copyright image from a labeled image. In Fig. 6(b), we first use $F^{-1}(\cdot)$ to recover the key image $K$ from $Z$. $F^{-1}(\cdot)$ is given by

$$K_{ij} = F^{-1}(Z_{ij}) = LSB(Z_{ij}) \tag{16}$$

where $LSB(Z_{ij})$ denotes the LSB of $Z_{ij}$. The recovered key image $K$ is chosen as an initial condition $\{y_{ij}(-1)\}$ of RDTCNN. Also, we choose the $[\frac{n}{2}]$th bit of every pixel of $Z$ as the other initial condition $\{y_{ij}(0)\}$ of RDTCNN. From Fig. 5, we can see that $\{y_{ij}(0)\}$ is the same as that we used in Fig. 6(a). Since our RDTCNN is reversible, after running an iteration, it gives the copyright label.

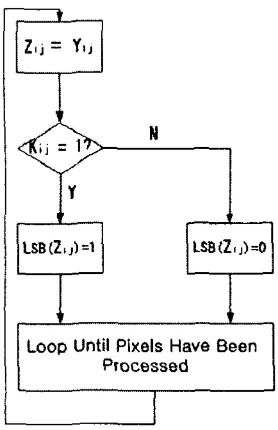
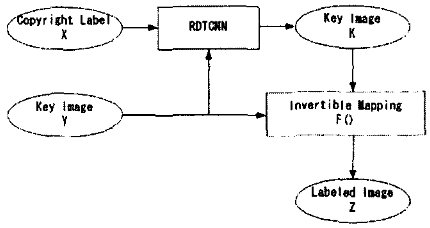
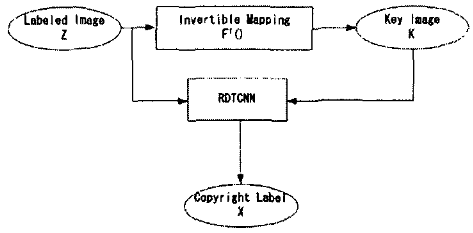



Fig. 5

Figure 5: The flow-chart of function $F(\cdot)$. $LSB(x)$ = least significant bit of $x$.




(a)




(b)

Fig 6.

Figure 6: (a) The image copyright labeling processes. (b) Extracting copyright label using an identical RDTCNN as in (a)

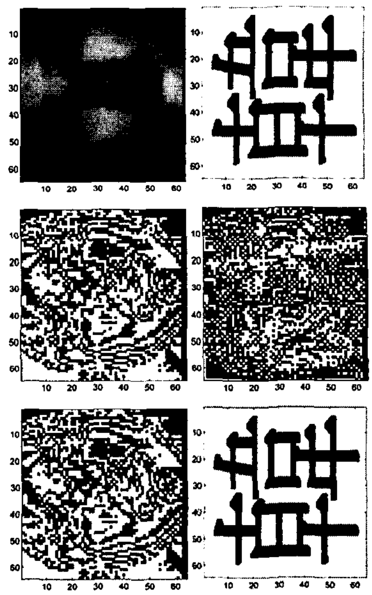


Fig. 7

Fig 7: Simulation results of labeling copyright label in a gray-scale image. (a) The original gray-scale image. (b) The binary image of a Chinese character as the copyright label. (c) The other initial condition getting from the original image. (d) The key image generated by the RDTCNN. (e) The labeled image. (f) The restored copyright label.

## V. Simulation Results

The simulation results are shown in Fig. 7. In Fig. 7(a), a gray-scale face image of size 64×64 and 256 gray levels is shown. Therefore every pixel is represented by 8 bits. Fig. 7(b) shows a binary image of a Chinese character, which is used to label the image in Fig. 7(a), of size 64×64. This image is used as the initial condition $\{y_{ij}(-1)\}$. Fig. 7(c) shows a binary image, in which every pixel has the same value as that of the 4th lowest bit of the corresponding pixel in Fig. 7(a). This image is used as the initial condition $\{yij(0)\}$. Fig. 7(d) shows the output of the RDTCNN after one iteration. This image is used as the key image $\{K_{ij}\}$. Then, the labeled image is shown in Fig. 7(e). One can see that the labeled image is indistinguishable from the original one. Fig. 7(f) shows the restored copyright label, which is identical to that shown in Fig. 7(b). In this simulation, the template $A$ is given by

$$A=\begin{pmatrix}0 & -1 & 1\\ 1 & -1 & -1\\ 1 & 1 & -1\end{pmatrix} \tag{17}$$

## Ⅳ. Conclusions

In this paper, we proposed a RDTCNN structure and use it to generate 2D random sequences for the purpose of digital image copyright labeling. Since the RDTCNN structure may have space-varying templates, and the templates are real numbers, the key space of our method is much larger than the method proposed in [11]. On the other hand, since a RDTCNN chip can work at a very high speed, our method has a great advantage over the other methods when lots of images are needed labeling in real time, e.g., digital video sequences of a commercial television channel.

However, our scheme can only work under the condition that during copying or transmission, the LSB of th image should not be changed. Thus, this scheme is not resistant to filtering, JPEG compression. From this point of view, the present method is not readily applicable and cot suitable in most of practical cases. But the principle of our method should have a great potential and can be improved in many ways.

## References

[1] D.Z.Knuth, *The art of computer progra-mming(vol.2),* 2nd ed., Assison-Wesley,Canada:1981.

[2] J.Dagpunar, *Principles of random variate generation,* Oxford University Press,New York: 1988.

[3] D.Z.Knuth, *The art of computer program ming(vol.1), 2nd ed.,* Assison-Wes ley,Canada:1981.

[4] H.Harrer and J.A.Nossek, "Discrete-time cellular neural networks,"*Int. J.Circuit*

*Theory and Appls.*, vol.20, pp.453-467, Sept.-Oct. 1992.
[5] L.O.Chua and L.Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst.*, Vol.35, pp1257-1272, Oct. 1988.
[6] B.Kahin, "The strategic environment for protecting multimedia," *IMA Intellectual Property Project Proceedings*, vol.1, no1, pp.1-8, 1994.
[7] E.Koch, J.Rindfrey and J.Zhao, "Copyright protection for multimedia data,"*Proceedings of the International Conference on Digital Media and Electronic Publishing*, pp.54-60, Dec., 1994mLeeds, UK.
[8] E.Koch and S.Zhao, "Towards robust and hidden image copyright labeling," *in Proceedings of 1995 Workshop on Nonlinear Signal and Image Processing*, pp.452-455, Neos Marmaras, Greece, June 1995.
[9] O.Bruydonchx, J.J. Quisquater and B.Macq, "Spatial method for copyright labeling of digital image," in *Proceedings of 1995 Workshop in Nonlinear Signal and Image Processing*, pp.456-459, Neos Marmaras, Greecem June 1995.
[10] I.Pitas and T.H. Kashkalis, "Applying signa ture on digital images," in *Procee dings of 1995 Workshop in Nonlinear Signal and Image Processing, pp.460-463, Neos Marmaras, Greece, June 1995.*
[11] H.Dedieu, M.Hasler and M.Bouslami, "Invis ible signatures in images based on nonlinear dynamical systems," in *1995 Intern ational Symposium in Nonlinear Theory and Its Applications*(NOLTA'95), pp.163-166, Las Vegas, U.S.A., December, 1995.
[12] R.G. Van Schyndel, A.Z. Tirkel and C.F. Osborne, "A digital water-mark," *Proceedings of ICIP'94*, vol.II, pp.86-90, Nov. 1994.
[13] M.Hasler, "Synchronization principles and applocations," in *Circuits and systems Tutorials*, Eds: C.Toumazou, N.Battersby and S.Portam pp.314-327, IEEE Press, New York:1994.
[14] T.Roska and L.O.Chua, "The CNN universal machine: An analogic array computer," *IEEE Trans. Circuits Syst.II*, vol.40, pp.163-173, Mar.1993

## 저자소개

이계호(Gye Ho Lee)

1988년 3월 조선대학교 산업대학원 전자계산학과 (석사 졸업)
1989년 8월 KT 대전연수원 (교수)
1989년 3월 조선대학교 산업대학원 전자계산학과(석사졸업)
1999년 8월 -현재 조선대학교 일반대학원 전자공학과 박사과정
2002년 3월 KT 전북본부 남원지사 기술부 (기술부장)
2003. 2월~현재 KT전남본부 순천지사 고흥지점(지점장)
※관심분야: 통신보안시스템설계

한승조(Seung Jo Han)

1980년 조선대학교 전자공학과 졸업
1982년 조선대학교 전자공학과 석사
1994년 충북대학교 전자계산학과 박사
1986년 6월~1987년 3월 뉴올리언즈대학 객원교수
1995년 2월~1996년 1월 텍사스대학 객원 교수
2000년 12월~2002년 2월 버클리대학 객원 교수
1998년 3월~현재 조선대학교 전자정보통신공학부 교수
※관심분야: 통신보안시스템설계, S/W불법복제방지시스템, ASIC설계