PAPER

Noise Reduction in CMOS Image Sensor Using Cellular Neural Networks with a Genetic Algorithm

Jegoon RYU[†a)], ***Student Member and*** **Toshihiro NISHIMURA**[†], ***Member***

SUMMARY In this paper, Cellular Neural Networks using genetic algorithm (GA-CNNs) are designed for CMOS image noise reduction. Cellular Neural Networks (CNNs) could be an efficient way to apply to the image processing technique, since CNNs have high-speed parallel signal processing characteristics. Adaptive CNNs structure is designed for the reduction of Photon Shot Noise (PSN) changed according to the average number of photons, and the design of templates for adaptive CNNs is based on the genetic algorithm using real numbers. These templates are optimized to suppress PSN in corrupted images. The simulation results show that the adaptive GA-CNNs more efficiently reduce PSN than do the other noise reduction methods and can be used as a high-quality and low-cost noise reduction filter for PSN. The proposed method is designed for real-time implementation. Therefore, it can be used as a noise reduction filter for many commercial applications. The simulation results also show the feasibility to design the CNNs template for a variety of problems based on the statistical image model.

key words: *cellular neural network, genetic algorithm, CMOS image sensor, photon shot noise*

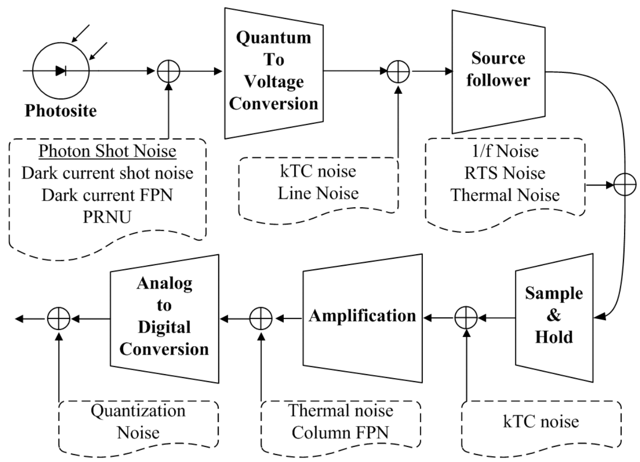

Fig. 1 Noise sources in the CMOS image sensor.

1. Introduction

The noise properties of digital cameras are under intensive research worldwide in an effort to advance image quality. Recently, CMOS image sensor has attracted public attention because of the various benefits such as lower power consumption, random access of each pixel, and simple manufacturing process, although it also has several drawbacks including low sensitivity and high noise level. Understanding the relationship between the image output of a CMOS image sensor and noise sources has also required a high-quality solution design with CMOS technology. Figure 1 shows the main noise sources in the CMOS image sensor [1]–[3].

Some of the noise characteristics in digital images can be reduced by using the extended exposure times and high ISO (International Standards Organization) sensitivity settings. High-speed settings in digital camera produce more noise than low ISO settings. In a CMOS image sensor, the charge-to-voltage conversion occurs in each pixel, and most of the functions are integrated on the chip. This technique has significant implications in sensor architecture. There are limitations such as smaller pixel size which will lead to lower dynamic range, higher dark voltage and lower fill factors. As a result, the number of noise sources in a CMOS image sensor, including the PSN, dark current noise, and Photo-Response Non-Uniformity (PRNU) noise, which together create the main problems [4], and the overall image quality will be reduced.

In this study, we consider the problem of restoring images degraded by the region-dependent PSN. The PSN results from the uncertainty in the number of photons collected in the photosite which is unavoidable and always exists in all imaging applications. The property of PSN is that when the signal increases, so does the amount of the noise. However, the PSN increases at a rate slower than the signal thus, as the signal increases, the SNR of PSN improves. This implies that the PSN will be a bigger problem in the shadows of underexposed images [5]–[7].

Recent noise reduction methods for the digital image are based on median filtering [8], [9], Lee filtering [10] or Wiener filtering [11]. Median filter is a simple and very effective noise reduction filtering process, but the performance is just effective for the noise induced by strong spike-like isolated values. Lee's filter can smooth noise and preserve edges efficiently, and it is efficient in smoothing Gaussian white noise, but it is not good at smoothing PSN. In Pavlovic and Tekalp [11], Wiener filter is applied for the restoration of image recorded by photographic film. But, the filtering is inadequate for removing PSN since it is designed mainly for Gaussian noise suppression.

In this paper, we propose a PSN reduction algorithm based on adaptive GA-CNNs. The genetic algorithm (GA) is used to determine the elements of CNNs in a template

Manuscript received June 10, 2009.
Manuscript revised October 11, 2009.
[†]The authors are with the Graduate School of Information, Production, and Systems, Waseda University, Kitakyushu-shi, 808–0135 Japan.
a) E-mail: jkryu@ruri.waseda.jp
DOI: 10.1587/transinf.E93.D.359

and is preferred to overcome the problems of stability and adaptation in the CNN [12]–[14]. The effectiveness of this learning algorithm is also confirmed by our computer simulated result.

Our motivation for the use of CNNs is based on their potential ability to provide reduced noise in a digital computational system and high-speed parallel signal processing that supports practical applications. The complete CNNs algorithm could also be easily implemented by using a Very Large Scale Integrated circuit (VLSI) where the connections between the processors are determined by a cloning template. Although the size of the CNNs cell array is small, it is also capable to handle large image sizes. The adaptive CNNs proposed in this study consists of five control and five feedback templates and five biases for reducing noise flexibly, since PSN image is changed according to the number of photons.

This paper is organized as follows. Section 2 shows the PSN modeling on CMOS image sensor. The CNNs model and the design of CNNs templates based on GA are discussed in Sect. 3. Experimental results are summarized in Sect. 4. Finally, the conclusion is presented in Sect. 5.

2. PSN Modeling on CMOS Image Sensor

The statistical property of photon shot noise is clearly elucidated in Goodman's works [15]. The PSN originates from the particle property of photon. Each photon emission and absorption is considered as an independent occurrence. The income number of photons fluctuate, and this fluctuation follows Poisson distribution [16], [17]. This fluctuation is observed as PSN.

If x denotes the number of photoevents occurring in a given time interval, or the number of photocounts, the probability $Prob(x)$ in Fig. 2 that x impulses fall within the time interval can be represented as:

$$Prob(x, \lambda) = \frac{\lambda^x}{x!} e^{-\lambda} \tag{1}$$

where λ is the average number of photons detected at the photosite, and it is given by:

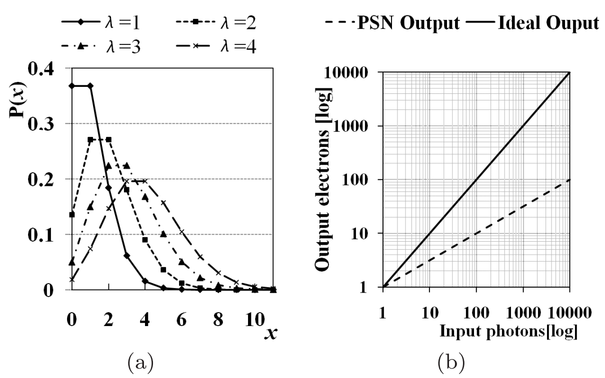

Fig. 2 Photon shot nose. (a) represents Poisson distribution of photon shot noise and (b) is the relation between income photons and absorption photons.

$$\lambda = \alpha I A \tau \tag{2}$$

where α is a proportionality constant, I is the intensity of light incident on the photosite of illuminated area A during the measurement time of interest τ. An important property of Poisson distribution is that its variance is equal to its average:

$$\sigma_x^2 = \lambda \tag{3}$$

The signal to noise ratio associated with this distribution, as defined by the ratio of mean to standard deviation, is given by:

$$SNR = \frac{\lambda}{\sigma_x} = \sqrt{\lambda} \tag{4}$$

The uncertainty in the number of photons collected during the given period of time is simply the photon shot noise given by:

$$\sigma_{ph} = \sigma_x = \sqrt{\lambda} \tag{5}$$

As a CMOS image sensor is operated under PSN-dominated region, the absorption of N photons on average

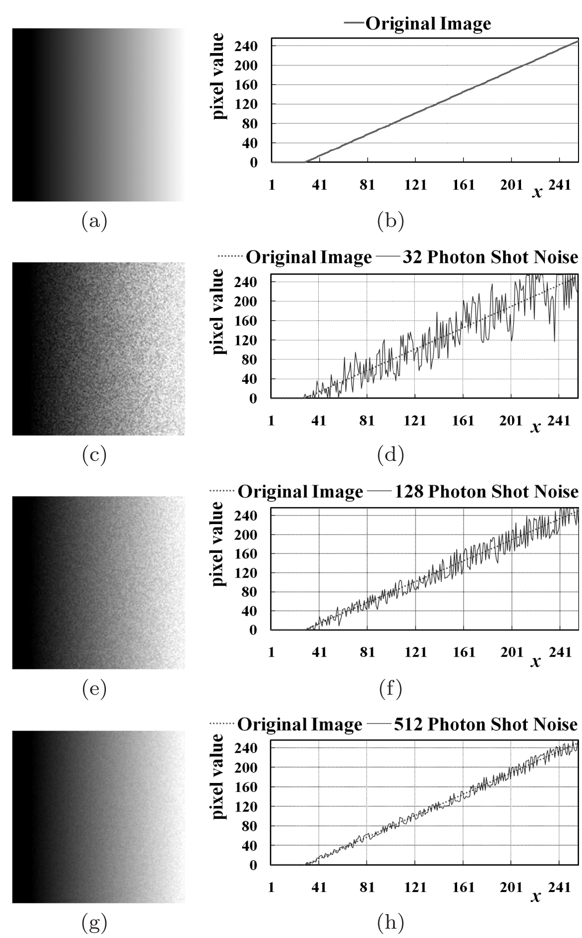

Fig. 3 PSN image and one-dimensional signal by Poisson distribution according to the nubmer of photons: (a) Noise-free image, (c) is PSN image of 32 photons, (e) 128 photons, and (g) 512 photons.

for a photo-sensor is accompanied with a standard deviation of $\sqrt{N}$ photons as in Fig. 2 (b). The magnitude of PSN depends on the signal level and exposure time, as predicted by Poisson Statistic. Figure 3 illustrates the characteristics of PSN according to the average number of photons.

3. Genetic Learning for Templates of CNNs

3.1 Architecture of CNNs

CNNs are characterized by the parallel computing of simple processing elements locally interconnected. They were initially proposed as continuous time architecture by Chua and Yang [18] and later a discrete-time extension (DTCNN) was introduced by Harrer and Nossek [19]. In this paper, two dimensional DTCNN is considered. The DTCNN is a clocked system, whose dynamical behavior is described by a set of discrete state equations. At a discrete time k, the state x^c of a cell $c = (c_1, c_2)$ depends on the time-independent input u^d applied to its neighboring cells $d = (d_1, d_2)$ and the time-varying output $y^d(k)$. The cell state equation is given as follows [19]:

$$x^c(k) = \sum_{d \in N_r(c)} a_d^c y^d(k) + \sum_{d \in N_r(c)} b_d^c u^d + i^c \tag{6}$$

The real-valued coefficients a_d^c are called the feedback coefficients and determine how the state of a cell c depends on the output of its neighboring cells d. Likewise, the control coefficients b_d^c means how the state of a cell c depends on the input of its neighbors d. For each cell, a real-valued cell bias i^c is added to adjust its threshold. A significant feature of CNNs is that it has two independent input capabilities, i.e., the generic input and the initial state of the cells. Normally they are bounded by:

$$|u^d| \le 1 \ \ and \ \ |x^c(0)| \le 1 \tag{7}$$

Each coefficient can be represented by the matrices A and B, known as cloning templates.

$$a_d^c = A_{d_1-c_1, d_2-c_2}, b_d^c = B_{d_1-c_1, d_2-c_2}, \ i^c = I \tag{8}$$

where A and B are called the feedback and control template, respectively. The matrices A and B are $(2r+1)\times(2r+1)$ real matrices and I is a scalar number in two dimensional CNNs. To avoid clutter, the Θ operator is introduced.

$$P\Theta g^c(k) = \sum_{d \in N_r(c)} P_{d_1-c_1, d_2-c_2} g^d(k) \tag{9}$$

Using Eq. (8) and (9), the Eq. (6) can be rewritten as:

$$x^c(k) = A\Theta y^c(k) + B\Theta u^c + I \tag{10}$$

Finally, the output of cell c at time step $k > 0$ is obtained by thresholding the state of c at time step $k-1$:

$$\begin{aligned} y^c(k) &= f(x^c(k-1)) \\ &= \begin{cases} +1 & for \ x^c(k-1) \ge 1 \\ x^c(k-1) & for \ -1 \le x^c(k-1) < 1 \\ -1 & for \ x^c(k-1) < -1 \end{cases} \end{aligned} \tag{11}$$

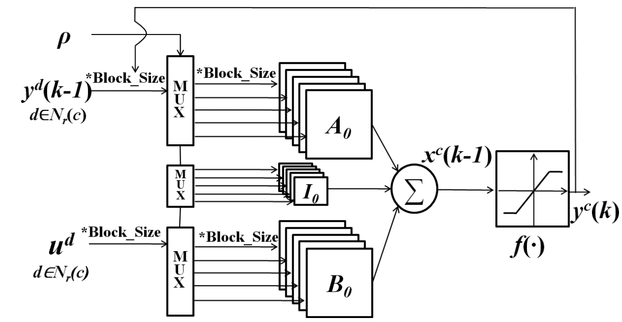

Fig. 4 Block diagram of the proposed DTCNN.

The real valued range of $y(k)$ is limited to $[-1, +1]$, where -1 represents a white pixel, and $+1$ represents a black pixel. Values between the range $[-1, +1]$ represent gray levels. The block diagram of the proposed CNNs is shown in Fig. 4. Images corrupted by PSN are influenced by the average number of photons. So, according to the range of the average number of photons, feedback template, control template, and bias are designed as $\{A_0, A_1, A_2, A_3, A_4\}$, $\{B_0, B_1, B_2, B_3, B_4\}$, and $\{I_0, I_1, I_2, I_3, I_4\}$, respectively, and computed.

3.2 Separable Templates

Assuming that the control and feedback templates are any symmetric two dimensional templates, large templates can be decomposed into a set of small-size templates. It plays a key role to be easily implemented in VLSI. For having the simplicity of design, symmetric templates are used in this system. If the given control and feedback templates A and B are two dimensional discrete convolution of two vectors, the template A and B can be separated into outer product of two 1-D templates: $A = A_c * A_r$, $B = B_c * B_r$: (A_c and B_c which are column vectors act along the vertical directions of the image plane, A_r and B_r which are row vectors act along the horizontal directions of the image plane). Here the symbol $*$ stands for both outer product of two vectors and 2-D convolution of two matrices.

For the symmetrical $1 \times N$ templates $A_c = [a_{ck} \ ... \ a_{c1} \ a_{c0} \ a_{c1} \ a_{c2} \ ... \ a_{ck}]^T$ and $A_r = [a_{rk} \ ... \ a_{r1} \ a_{r0} \ a_{r1} \ a_{r2} \ ... \ a_{rk}]$, ($k$ is $(N-1)/2$), the following $N \times N$ two dimensional feedback template and control template result through outer product:

$$A = \begin{pmatrix} a_{ck}a_{rk} \dots a_{ck}a_{r1} & a_{ck}a_{r0} & a_{ck}a_{r1} \dots a_{ck}a_{rk} \\ \vdots \ \ \ddots \ \ \vdots & \vdots & \vdots \ \ \ddots \ \ \vdots \\ a_{c1}a_{rk} \dots a_{c1}a_{r1} & a_{c1}a_{r0} & a_{c1}a_{r1} \dots a_{c1}a_{rk} \\ a_{c0}a_{rk} \dots a_{c0}a_{r1} & a_{c0}a_{r0} & a_{c0}a_{r1} \dots a_{c0}a_{rk} \\ a_{c1}a_{rk} \dots a_{c1}a_{r1} & a_{c1}a_{r0} & a_{c1}a_{r1} \dots a_{c1}a_{rk} \\ \vdots \ \ \ddots \ \ \vdots & \vdots & \vdots \ \ \ddots \ \ \vdots \\ a_{ck}a_{rk} \dots a_{ck}a_{r1} & a_{ck}a_{r0} & a_{ck}a_{r1} \dots a_{ck}a_{rk} \end{pmatrix} \tag{12}$$

$$B = \begin{pmatrix} b_{ck}b_{rk} \dots b_{ck}b_{r1} \; b_{ck}b_{r0} \; b_{ck}b_{r1} \dots b_{ck}b_{rk} \\ \vdots \quad \ddots \quad \vdots \qquad \vdots \qquad \vdots \quad \ddots \quad \vdots \\ b_{c1}b_{rk} \dots b_{c1}b_{r1} \; b_{c1}b_{r0} \; b_{c1}b_{r1} \dots b_{c1}b_{rk} \\ a_{c0}b_{rk} \dots b_{c0}b_{r1} \; b_{c0}b_{r0} \; b_{c0}b_{r1} \dots b_{c0}b_{rk} \\ b_{c1}b_{rk} \dots b_{c1}b_{r1} \; b_{c1}b_{r0} \; b_{c1}b_{r1} \dots b_{c1}b_{rk} \\ \vdots \quad \ddots \quad \vdots \qquad \vdots \qquad \vdots \quad \ddots \quad \vdots \\ b_{ck}b_{rk} \dots b_{ck}b_{r1} \; b_{ck}b_{r0} \; b_{ck}b_{r1} \dots b_{ck}b_{rk} \end{pmatrix} \tag{13}$$

3.3 Genetic Learning

Genetic Algorithm is a kind of adaptive search methods that are modeled after the genetic evolutionary process. One of the most important part of GA is the evaluation of the so called "fitness function", objective or cost function, to allocate a quality value to every solution produced. Another important feature of GA is intrinsically parallel. In place of using a single point and local gradient information, they evolve a generation of candidate solutions where each individual represents a specific solution not related to other solutions. Therefore, many real-world applications including the DTCNN template learning problem can be easily implemented on parallel machines resulting in a significant reduction of the required computation time [12]. GA has been widely used for DTCNN templates learning tasks, minimization of the objective function, called optimization [20]. It has also been shown the influence and robustness in theory and practice [21]–[23].

A. Real-Coded Chromosome

The implementation of GA begins with the design of the decision variables and the encoding scheme of the chromosome. Encoding method must guarantee the effective transfer of information between chromosome strings. In this paper, we confine ourselves to floating point (real-coded) chromosomes [24]. The original GA uses a binary-coded format of chromosomes. The choice of the real-coded chromosomes improve the flexibility and accuracy of chromosomes representation. Moreover, there is no loss in precision when discretize to binary or other values. There is also greater opportunity to use different genetic operators.

The symmetrical $1{\times}N$ templates $A_c = [a_{ck} \dots a_{c1}\ a_{c0}\ a_{c1}\ a_{c2} \dots a_{ck}]$ and $A_r = [a_{rk} \dots a_{r1}\ a_{r0}\ a_{r1}\ a_{r2} \dots a_{rk}]$ are treated as a starting point in the representation problem that gives rise to a floating point chromosome. To represent the DTCNN IIR filter, the chromosome contains the symmetrical elements of the templates A and B. For $N \times N$ DTCNN templates learning, the floating point chromosome of length $(2N + 3)$ is given by:

$$\left\{ \begin{array}{l} a_{ck},\ \dots, a_{c1}, a_{c0}, a_{rk},\ \dots, a_{r1}, a_{r0}, \\ b_{ck},\ \dots, b_{c1}, b_{c0}, b_{rk},\ \dots, b_{r1}, b_{r0}, i \end{array} \right\} \tag{14}$$

where k is $(N - 1)/2$ and the chromosome is illustrated in Fig. 5. Each component has the floating point value with the precision up to 0.0001. The GA generates the starting population of length $(2N + 3)$ containing random real numbers in $[-2, +2]$ except for the a_{c0}, a_{r0}, b_{c0} and b_{r0} elements

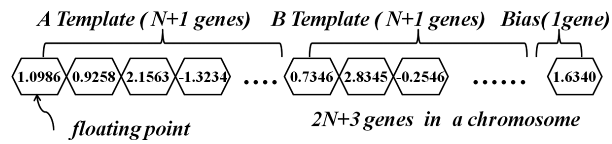

Fig. 5 Structure of the chromosome.

which are chosen in $[-2.235, +2.235]$ because of the stability constraint for CNNs circuits, the $[-5, +5]$ range that corresponds to the expected VLSI technological limits, and adaptive searching for DTCNN IIR filter.

B. Fitness Function

GA searches for the optimal solution by maximizing a fitness function that assigns a quality value to each individual of the population. This quality value is used as a comparative evaluation criteria of each individual against other members of the population and is a key factor for the real optimization problem.

The fitness value of a member indicates how suitable it is, by representing the desired solution in $M \times N$ image. To compute the fitness value, evaluation procedure is executed with the encoded template, and the results are compared to the desired image using the following equation:

$$\varepsilon_k = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (\varphi_{ij} - \eta_{ij}^{k})^2}{M \times N} \tag{15}$$

and Eq. (15) can also be represented as follows:

$$\varepsilon_k = \frac{\|\Phi - H_k\|^2}{M \times N} \tag{16}$$

where k denotes the number of the chromosome, φ_{ij} is the value of the (i, j) pixel of the desired image and η_{ij} stands for the pixel of the denoising image. Φ denotes the set of the desired output image, and H_k is the set of the denoising image by k chromosome.

To compute the fitness value, the following equation is designed for a better solution.

$$f_k = 10 \log_{10} \left(\frac{Peakvalue^2}{\varepsilon_k} \right) \tag{17}$$

where *Peakvalue* is the maximum pixel value in $M \times N$ image. Individuals that represent better solutions are awarded with higher fitness values, thus enabling them to survive more generations.

C. Genetic Algorithm Operations

The selection of individuals to reproduce a superior generation is quite an important problem, since it influences the existence or decline of competitive individuals. The higher the fitness value, the more likely it is that the individual will be chosen for the next generation.

Here we use a tournament selection method. In the tournament selection, n individuals are randomly selected from the population and the best of the n individuals will be selected as the parent individual for further genetic processing. This process is repeated until the mating pool is filled. Though this strategy selects good individuals for the next generation, it cannot ensure that the best individual in any population survives throughout the optimization process. To overcome this problem, the GA performs the descending sort of all individuals using the fitness value, and elitist strategy is used in the GA such that once some highest individuals among the current generation are found, these will be kept unchanged into the next generation.

Crossover is mainly accountable for the global-search property of GA. It basically combines understructures of two parent chromosomes to produce offspring chromosomes. Several crossover schemes have been suggested in the literature, such as single-point, multi-point, random crossover or uniform crossover. In our approach, the BLX [25] crossover is applied to the floating point numbers. Figure 6 represents the BLX crossover operation for the one-dimensional case. A gene y for the offspring chromosome is generated from the space $[\lambda_1,\ \lambda_2]$ as follows:

$$y = \begin{cases} \lambda_1 + r(\lambda_2 - \lambda_1) : if\ u_{min} \le y \le u_{max} \\ repeat\ generating\ :\ otherwise \end{cases} \tag{18}$$

where r is $uniform\ random\ number \in [0, 1]$.

$$\begin{aligned} \lambda_1 &= \alpha - \phi(\beta - \alpha) \\ \lambda_2 &= \beta + \phi(\beta - \alpha) \end{aligned} \tag{19}$$

Note that λ_1 and λ_2 will lie between u_{min} and u_{max}, the variables' lower and upper bounds, respectively. It also was observed that $\phi = 0.5$ provides good results.

Mutation is the operator responsible for the injection of new genetic material into the population. A uniform mutation operator is used to the real coded numbers. It randomly selects one of the variables from a parent chromosome which is chosen by tournament selection, and sets a uniform random number which do not exceed the bounds of the region allowed for the parameters.

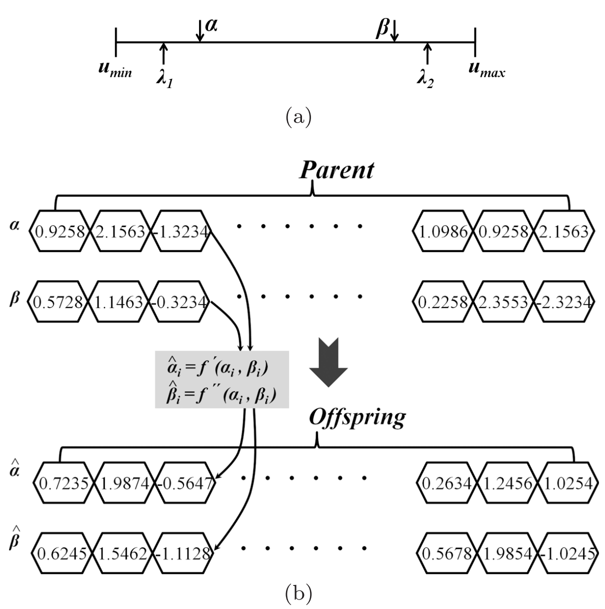

Fig. 6 Scheme of BLX crossover and the procedure of crossover.

4. Experimental Results and Discussion

For demonstrating the effectiveness of the proposed method, we have performed experiments on the 256×256 gray scale images that are noisy images polluted by PSN. Test images are taken from USC-SIPI and CVG-UGR image database [26], [27]: Barbara, Cameraman, Lenna, Boat, Airplane, Baboon, and Peppers.

There are several parameters in a GA which have to be specified for the template learning. The following values in Table 1 were used in the simulation. For control template and feedback template, 5×5 template is adopted and the length of chromosome is 13. Figure 7 represents the change of the fitness as the procedure of GA increases. Though the fitness is an important parameter which can determine the improved quality of population, its absolute value is not very important. Our simulation should be efficient to many kinds of images, so we took an average with several images for fitness of one generation. The figure shows the convergence of our GA, and the quality of population is improved when the run time increases.

The quality of denoised images is evaluated by normalized mean square error (NMSE), Laplace mean square error (LMSE) and peak signal to noise ratio (PSNR).

Table 1 Parameter values in genetic algorithm.

Item	Value
The population size	200
The maximum generation	3000
The length of chromosome	13(5*5CNN), real number
Crossover probability Pc	0.65, BLX Crossover
Mutation probability Pm	0.02 per gene
Elite preservation size Pe	0.2
Selection criteria	Tournament selection

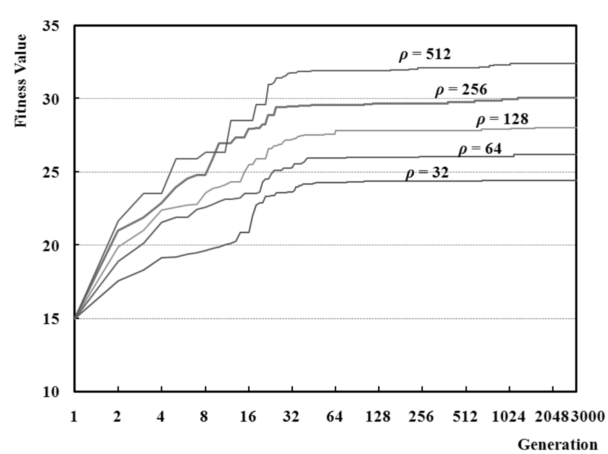

Fig. 7 Change of fitness according to PSN.

Fig. 8 Noise reduction experiment on Barbara image: (a) Noise-free image, (b) PSN image with 256 photons, (c) result by mean filter, (f) α-trim mean filter, (d) median filter, (e) adaptive median filter, (g) Wiener filter, and (h) the proposed filter.

$$NMSE = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(O_{i,j} - E_{i,j})^2}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(O_{i,j})^2} \tag{20}$$

$$LMSE = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(\Psi(O_{i,j}) - \Psi(E_{i,j}))^2}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(\Psi(O_{i,j}))^2} \tag{21}$$

$$\Psi(O_{i,j}) = O_{i+1,j} + O_{i-1,j} + O_{i,j+1} + O_{i,j-1} - 4O_{i,j} \tag{22}$$

where M and N denote the original image size, and (i, j) is the pixel coordinate in an image. $O_{i,j}$ and $E_{i,j}$ denote the pixel values of original image and denoised image, respectively.

To calculate the PSNR in decibels (dB), the Mean Square Error (MSE) which requires two $M \times N$ gray scale images $O_{i,j}$ and $E_{i,j}$ is calculated using the following equation:

$$MSE = \frac{1}{M \times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(O_{i,j} - E_{i,j})^2 \tag{23}$$

$$PSNR = 10\log_{10}\left(\frac{(PeakSingal)^2}{MSE}\right) \tag{24}$$

where $PeakSignal$ is the maximum pixel value of the image. LMSE is one of quality measure based on the properties of the human visual system. This measure can yield a good performance for the signification of object boundary and edge in images for the human visualization and for imaged which are severely degraded with low spatial frequency noise. With given noise variances, the smaller the values of NMSE and LMSE, the better the quality of the denoised image. Contrarily, the larger the values of PSNR, the better the quality of the denoised image.

We compared the results of the proposed method with other denoising techniques including mean filtering, α-trim mean filtering, median filtering, adaptive median filtering, Lee filtering, and Wiener filtering. The images denoised by other methods were shown in Fig. 8 with Barbara image, and the corresponding values of NMSE, LMSE, and PSNR calculated were also shown in Table 2. Denoising results obtained by the proposed method are shown in the bottommost row in Fig. 8 and Table 2. From the simulation results, it is easy to see that the noise has been effectively reduced, the visual effect has been highly enhanced compared with the corresponding noise versions, and the proposed method has received the best numerical performance. Although the denoised image by the proposed method still appears a little bit grainy, the shape and content of the original image are retained very well. Under the same noisy level, the values of NMSE and LMSE of the denoised images are much smaller than those of the corresponding noisy images, while the PSNR values of the denoised images are much larger than those of the noisy images. It can also be observed that the smaller the photon level ρ is, the smaller the PSNR values are; whereas the larger the NMSE and LMSE values are.

The mean filter is the worst denoiser in this simula-

Table 2 Evaluation obtained by different denoising algorithm. Image: Barbara.

Method	$\rho = 32$			$\rho = 64$			$\rho = 128$			$\rho = 256$			$\rho = 512$		
	NMSE	LMSE	PSNR	NMSE	LMSE	PSNR	NMSE	LMSE	PSNR	NMSE	LMSE	PSNR	NMSE	LMSE	PSNR
Mean	0.0305	10.520	21.66	0.0259	14.817	22.36	0.0238	18.654	22.76	0.0228	21.271	22.93	0.0228	21.304	23.01
α-t. mean	0.0374	10.366	20.77	0.0299	14.667	21.73	0.0265	18.861	22.26	0.0250	21.354	22.53	0.0243	21.951	22.63
Median	0.0443	3.8847	20.01	0.0337	5.3503	21.22	0.0285	6.6743	21.94	0.0260	7.5952	22.35	0.0247	7.512	22.59
A-median	0.0454	1.5629	19.93	0.0296	1.9122	21.84	0.0214	2.2208	23.19	0.0176	2.5172	24.05	0.0154	2.4974	24.63
Lee	0.0305	11.456	21.67	0.0259	17.209	22.71	0.0237	22.617	22.97	0.0228	26.694	23.19	0.0121	29.348	23.58
Wiener	0.0659	0.7660	18.32	0.0327	0.6089	21.40	0.0162	0.4277	24.43	0.0081	0.2642	27.46	0.0040	0.1480	30.43
Prop. filter	0.0395	0.6459	22.17	0.0174	0.4602	24.46	0.0123	0.3748	26.62	0.0068	0.2183	29.56	0.0038	0.1003	33.01

Table 3 The PSNR value by the proposed method.

Image	The number of input photons				
	$\rho = 32$	$\rho = 64$	$\rho = 128$	$\rho = 256$	$\rho = 512$
Barb.	22.17	24.46	26.62	29.56	33.01
Came.	22.15	24.04	26.01	29.83	31.89
Lenna	22.23	24.81	26.19	29.01	32.09
Boat	22.56	23.62	26.32	29.58	32.16
Airp.	22.17	24.52	26.95	29.01	32.48
Babo.	22.18	24.36	26.88	29.00	32.62
Pepp.	21.95	24.05	26.08	29.34	32.95

tion. Wiener filter can also hardly reduce the noise, but it is worse than the proposed method. The result confirmed that the proposed method is about 3 db better than Wiener filter. Compared with LNSE of Wiener filter and the proposed method, the proposed method shows better performance, which is to preserve the fine details and boundaries of objects, than Wiener filter. The PSNR values of the proposed method using several test images are also listed in Table 3. The main advantage of this filter is that PSN is suppressed very successfully while maintaining the sharpness of the fine details and edges. The simulation results demonstrate that the adaptive GA-CNNs have good overall performance.

5. Conclusion

Template learning is a crucial step in CNNs technology. In this paper, a genetic approach for CNNs template learning and optimization to reduce the PSN noise was introduced. The adaptive CNNs were also designed to reduce PSN according to the number of photons. We noted that noise reduction of CMOS image for the proposed method is equivalent to seeking a maximum of a fitness function. Simulation results showed that the proposed method could reduce PSN noise more effectively than mean, median, and Wiener filter. According to the metrics of PSNR, NMSE, and LMSE, which are the image-quality measures, the proposed method produced good results in reducing PSN and preserving edges and fine details. Finally, since this work does not address noise reduction for color images, this omission will be rectified in future work. Moreover, the satisfactory performance of the algorithm encourages us to utilize this method in different applications.

References

[1] J.E. Farrell, F. Xiao, P. Catrysse, and B. Wandell, "A simulation tool for evaluating digital camera image quality," Proc. SPIE Electron. Imaging 2004 Conf., vol.5294, pp.124–131, 2004.
[2] H. Wach and E. Dowski, "Noise modeling for design and simulation of computational imaging systems," Proc. IS&T/SID Color Imag. Conf.: Color Sci., Syst., and Appl., pp.211–216, 2004.
[3] T.T. Kolehmainen, J. Aikio, M. Karppinen, A. Mattila, J. Makinen, K. Kataja, K. Tukkiniemi, and P. Karioja, "Simulation of imaging system's performance," Proc. SPIE Optic. Sci. and Tech., vol.5178, pp.204–212, 2004.
[4] Y. Reibel, M. Jung, M. Bouhifd, B. Cunin, and C. Draman, "CCD or CMOS camera noise characteristics," Proc. Euro. Physical J. of Applied Physics, pp.75–80, 2003.
[5] C.W.J. Beenakker and M. Patra, "Photon shot noise," Modern Physics Letters B, World Scientific Publishing Company, vol.13, no.11, pp.337–347, 1999.
[6] N. Kawai and S. Kawahito, "Noise analysis of high-gain, low-noise column readout circuits for CMOS image sensor," IEEE Trans. Electron Devices, vol.51, no.2, pp.185–194, 2004.
[7] S. Kawahito and S. Itoh, "A photon count imaging using an extremely small capacitor and a high-precision low-noise quantizer," Proc. SPIE IS&T, vol.5017, pp.68–75, 2003.
[8] A. Ioannidis, D. Kazakos, and D.D. Watson, "Application of median filtering on nuclear medicine scintigram images," Proc. 7th Int. Conf. Pattern Recognition, pp.33–36, 1984.
[9] E.R. Ritenour, T.R. Nelson, and U. Raff, "Application of the median filter to digital radiographic images," Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, vol.9, pp.23.1.1–23.1.4, 1984.
[10] J.S. Lee, "Digital image enhancement and noise filtering by use of local statistics," IEEE Trans. Pattern Anal. Mach. Intell., vol.2, no.2, pp.165–168, 1980.
[11] G. Pavlovic and A.M. Tekalp, "Restoration in the presence of multiplicative noise with application to scanned photographic images," Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, vol.4, pp.1913–1916, 1990.
[12] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[13] T. Sziranyi and M. Csapodi, "Texture classification and segmentation by cellular neural networks using genetic learning," Comput. Vis. Image Understand., vol.71, no.3, pp.255–270, 1998.
[14] P. Lopez, D.L. Vilarino, and D. Cabello, "Design of multilayer discrete time cellular neural networks for image processing tasks based on genetic algorithm," IEEE Int. Symp. Circuits Syst. Proc., vol.4, pp.133–136, 2000.
[15] J.W. Goodman, Statistical Optics, John Wiley & Sons, 1985.

[16] M. Rabbani, "Bayesian filtering of poisson noise using local statistics," IEEE Trans. Acoust. Speech Signal Process., vol.36, no.6, pp.933–937, 1998.
[17] G. Cortellazzo, G.A. Mian, and R. Parolari, "Statistical characteristics of granula camera noise," IEEE Trans. Circuits Syst. Video Technol., vol.4, no.6, pp.536–543, 1994.
[18] L.O. Chua and L. Yang, "Cellular neural networks: Theory and application," IEEE Trans. Circuits Syst., vol.35, no.10, pp.1257–1290, 1988.
[19] H. Harrer and J.A. Nossek, "Discrete-time cellular neural networks," Int. J. Circuit Theory Appl., vol.20, no.5, pp.453–467, 1992.
[20] T. Kozek, T. Roska, and L.O. Chua, "Genetic algorithm for CNN template learning," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol.40, no.6, pp.392–402, 1993.
[21] H. Harrer, "Multiple layer discrete time cellular neural networks using time variant templates," IEEE Trans. Circuits Syst. II, vol.40, no.3, pp.191–199, 1993.
[22] D.L. Vilarino, D. Cabello, M. Balsi, and V.M. Brea, "Image segmentation based on active contours using discrete time cellular neurla networks," Proc. 5th IEEE Int. Workshop on Cellular Neural Networks and Their Applications, pp.331–336, 1998.
[23] P. Lopez, M. Balsi, D.L. Vilarino, and D. Cabello, "Design and training of multilayer discrete time cellular neural networks for antipersonnel mine detection using genetic algorithms," Proc. 6th IEEE Int. Workshop on Cellular Neural Networks and Their Applications, pp.363–368, 2000.
[24] F. Herrera, M. Lozano, and J.L. Verdegay, "Tackling real-coded genetic algorithms: Operators and tools for behavioral analysis," Artif. Intell., vol.12, pp.265–319, 1998.
[25] L.J. Eshelman and J.D. Schaffer, "Real-coded genetic algorithm and interval schemata," Foundations of Genetic Algorithms, vol.2, pp.187–202, 1993.
[26] USC-SIPI Image Database [Online], Available: http://sipi.usc.edu/services/database/Database.html
[27] CVG-UGR Image Database [Online], Available: http://decsai.ugr.es/cvg/dbimagenes/index.php
[28] R.D. Gow, D. Renshaw, K. Findlater, L. Grant, S.J. McLeod, J. Hart, and R.L. Nicol, "A comprehensive tool for modeling CMOS image-sensor-noise performance," IEEE Trans. Electron Devices, vol.54, no.6, pp.1321–1329, 2007.
[29] Y.H. Chung, Y. Choi, T.Y. Song, J.H. Jung, G.S. Cho, Y.S. Cheo, K.H. Lee, S.E. Kim, and B.T. Kim, "Evaluation of maximum-likelihood position estimation with poisson and Gaussian noise models in a small gamma camera," IEEE Trans. Nucl. Sci., vol.51, no.1, pp.101–104, 2004.
[30] X.H. Han, Z. Nakao, Y.W. Chen, and R. Kodama, "An ICA-domain shrinkage based poisson-noise reduction algorithm and its application to penumbral imaging," IEICE Trans. Inf. & Syst., vol.E88-D, no.4, pp.750–757, April 2005.

Jegoon Ryu is currently pursuing his PhD at Waseda University Graduate school of Information, Production, and Systems in Japan since 2006. He received his B.S. and the M.S. degree in Electronic Engineering from Inha University in Korea, in 1999 and in 2004, respectively. From 2004 to 2006, he was a lecturer at Korea Polytechnic University and a researcher at Intelligence Healthcare system Research Center. His main research interests are the CMOS image sensor, robot vision, and medical image & signal processing. He is currently a student member of IEEJ, IEEE and RSJ.

Toshihiro Nishimura is currently a professor of Information Architecture, Waseda University Graduate school of Information, Production and System since 2003. He received the Dr.-Eng. Degree from the University of Tokyo. His research interests are the artificial human retina, CMOS image sensor, medical ultrasonic imaging, nerve physiology, cardiovascular system, and all the power electronics on human affairs and space missions. He is currently Chairman of Committee on Medical & Biological Engineering IEE Japan, Trustee of Medical Ultrasonics Society, Councilor of IEEE power Electronics Society and a Member of JSMBE, IEEJ and IEEE, the general chair of ISICE 2007 and 2008. He is decorated many from the society of IEEJ-C and the IEEE. e-mail: tosi-hiro@waseda.jp