# Blind Deconvolution by Self-Organization

Wen-Pin Tai, Ruei-Sung Lin, and Cheng-Yuan Liou[1]
[1] Department of Computer Science and Information Engineering,
National Taiwan University, 10764, Taipei, Taiwan ROC.
Tel: 8862-3625336 ext.515, Fax: 8862-3628167, E-mail: cyliou@csie.ntu.edu.tw

## Abstract

*In this work †, we devise a self-organizing network to solve both the unknown system and unknown input in blind deconvolution of blurred images. We utilize a criterion function which has a similar form as the Kullback-Leibler cross information formula to adapt the network's weights to approach the unknown system function. This adaptation gradually reduces the criterion value which is a distance measure between the system output and the output of the adapted system with a reconstructed input signal. The weight matrices of the neurons in the network will be shifted versions of the system function and will be aligned in the network according to their shifts during convergence. This is because that the convolution operation copes with this network scheme and the hidden topology of the shifted system functions can be aligned similarly in a 2D plane.*

Keywords: self-organizing network, blind deconvolution, blurred image, Kullback-Leibler information criterion.

## 1 Introduction

Blind deconvolution [1] is the problem of determining the components of a convolution, the input signal and the time-invariant system function, from their contaminated output. The operations involved in such a deconvolution process consist of the system identification and the input reconstruction. Because both the system function and its input are unknown, the problem is difficult to be solved based on the system output only. Many methods are devised to find these two components by incorporating properly specified knowledge (or constraints) [2].

The usages of various constraints for insufficient information in the deconvolution process also limit the applications of these methods. Parametric methods [3] [4] [5] model the input signal as an autoregressive moving average process by imposing some assumptions. The deconvolution involves the parameter estimation under these specific assumptions. There are several methods for blind deconvolution with few constraints. The nonparametric methods [6] [7] [8] [9] do not assume any model for system function or input signal. Non-negativity and finite support are the fewest deterministic constraints.

We propose a self-organizing approach to identify the system function and to reconstruct input signal for the blind deconvolution problem. We employ a 2D self-organizing network [10] with a criterion to solve the problem. This criterion is devised to minimize the distance between two function outputs based on the Kullback-Leibler [11] cross information expression. This expression fits and benefits much to our approach due to its many well-behaved analytic properties. By minimizing the distance function and evolving the network, we can solve both the input signal and the system function. In addition, the solutions are robust to the noise in the system output.

In the next section, the blind deconvolution problem is briefly reviewed. The self-organizing technique for blind deconvolution is discussed. The method for system identification and input reconstruction is derived in Section 2. Section 4 contains the simulation results carried out to display the performance of this approach.

## 2 The Self-Organizing Network for Blind Deconvolution

In discrete-time signal processing, the operation of convolution is defined by

$$g(m,n) = h(m,n) * f(m,n)$$

†This work is supported by National Science Council, Taiwan, Republic of China.

$$= \sum_{(p,q)\in H} h(p,q)f(m-p,n-q),$$
$$\text{for } (m,n) \in F, \qquad (1)$$

where $g(m,n)$, $f(m,n)$, and $h(m,n)$ are the system output, system input, and system function at the pixel point $(m,n)$, respectively. Both $f(m,n)$ and $h(m,n)$ are defined in proper support regions $F$ and $H$, which are usually rectangular regions in images. The problem to identify the system function $h(m,n)$ and to reconstruct the input $f(m,n)$ using the only information, the system output $g(m,n)$, is known as the blind deconvolution. In practical applications, the noise may appear in the output $g(m,n)$ of model (1).

In most image cases, the input $f(m,n)$, for $(m,n) \in F$, is finite and nonnegative, that is, $0 \leq f(m,n) < \infty$. For each $(m,n) \in H$, the value of system function $h(m,n)$ is also assumed nonnegative, that is, $h(m,n) \geq 0$. We require that the convolution in (1) does not change the energy of the input signal, the values of system function $h(m,n)$ must satisfy $\sum_{(m,n)\in H} h(m,n) = 1$. We also scale the values of $f$ in $F$, by the normalization $\sum_{(m,n)\in F} f(m,n) = 1$, such that $f$ can be processed as a 2D distribution function in our approach.

In [12], the blind deconvolution problem is formulated as a self-organizing learning process. Since this deconvolution is performed in absence of a training (or teaching) sequence, the problem is solved by achieving a matching in the learning process. We also design a matching (criterion or energy) function and an unsupervised learning algorithm to evolve a self-organization network with different contents.

We develop a new self-organizing network for the blind deconvolution problem. "The Magic TV" [10] application inspires us that the shifted blur (or system) functions can be aligned in a 2D rectangular plane according to their shifts when the input image is situated in a similar 2D rectangular plane and the optical aperture is in between the input image and the output image. A point source in the input plane will generate a shifted system function on the output image plane according to the shift of this point source from the center of the plane (as shown in Figure 1). This network scheme copes with the convolution operation where a shifted version of blur function in the output plane is generated from the shifted point source in the input plane. So we devise a 2D self-organizing network where neurons are regularly arranged in a 2D square plane to learn the system function.

This 2D network employs $N_1 \times N_2$ neurons. Each neuron $(i,j)$, $(i,j) \in N$, $i = 1 \ldots N_1$, $j = 1 \ldots N_2$, has its own synapse matrix $[w_{ij}(m,n)]$, which has the same

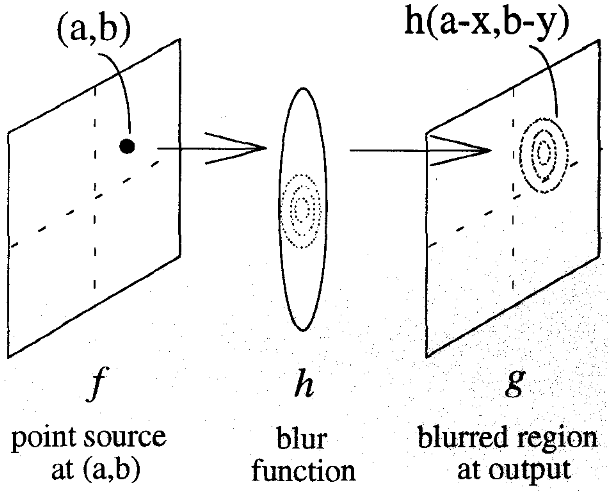


**Figure 1. In a real imaging system, the point source generates a shifted version of blur function in the output plane, which is equivalent to a 2D convolution operation.**

dimension as the size of $F$. The value $w_{ij}(m+i', n+j')$, where $(i',j')$ is the pixel position of the $(i,j)$-th neuron on the image, corresponds to the component $h(m,n)$ on this support. In our case, we set $N_1 = 32$, $N_2 = 32$, $i' = 8 \times (i-1) + 5$, and $j' = 8 \times (j-1) + 5$ for a $256 \times 256$ image. This means an image is partitioned into regular squares with the same size where these squares may overlap. Each neuron represents a square region. In our case for $256 \times 256$ image, each square has size $8 \times 8$ pixels. Each neuron is right on the center of the square, $(i',j')$. During the self-organization, the synapse matrices in the network are adapted as various candidates to approximate the system function $h(m,n)$. The one candidate which fits the network scheme will be further evolved. The synapses of the best-matching neuron $(c_1,c_2)$ will be a best estimate of the shifted version of $h$. After the estimated system function, $\hat{h}(m,n) = w_{c_1c_2}(m + c'_1, n + c'_2)$, is determined, the reconstructed input, $\hat{f}(m,n)$, will be determined optimally with respect to the minimization of the utilized criterion function. Note that when the network converges, all center excitation parts of matrix $w_{ij}$ will have the same shape and represent the same system function.

For system identification, the criterion function for matching is defined to measure the distance between the normalized system output $g(m,n)$ and the output of the learned system function $w_{ij}(m+i', n+j')$ from the estimated $f(m,n)$. The criterion $e_{ij}$ for the neuron $(i,j)$ is defined by $e_{ij} \equiv e(w_{ij}, f)$, where

$$e(w_{ij}, f) = \sum_{(m,n)\in F} g(m,n) \log \frac{g(m,n)}{w_{ij}(m+i', n+j') * f(m,n)}, \tag{2}$$

subject to $0 \leq w_{ij}(m+i', n+j') \leq 1$, $0 \leq f(m,n) \leq 1$, $\sum_{(m+i',n+j')\in H} w_{ij}(m+i', n+j') = 1$, and $\sum_{(m,n)\in F} f(m,n) = 1$. Note that in equation (2) $w_{ij}(m+i', n+j')$ is equal to $h(m,n)$. All values outside $H$ are set to zeros. In our later experiments, we will preset $H$ to a $11 \times 11$ support and values of $w_{ij}$ outside $H$ to zeros. The criterion function $e$ in (2) is defined following the expression of Kullback-Leibler information criterion [11]. It provides a quantitative measure of how closely the shape of $w_{ij}(m+i', n+j') * f(m,n)$ matches that of $g(m,n)$, for $(m,n) \in F$.

The best-matching estimate of the system function, $w_{c_1c_2}$, is determined by minimizing the weighted criterion functions centered at the neuron $(c_1, c_2)$

$$E_{c_1c_2} = \sum_{(i,j)\in N} \mathcal{H}_{(c_1,c_2),(i,j)} \cdot e_{ij}, \tag{3}$$

where $\mathcal{H}_{(c_1,c_2),(i,j)}$ is the neighborhood function which weights the interaction of synapse matrices $w_{c_1c_2}$ and $w_{ij}$ during self-organization. The function $E$ in (3) is defined for each neuron $(c_1, c_2)$ (as suggested by [13]).

For input reconstruction, the optimal estimate of the input, $\hat{f}(m,n)$, $(m,n) \in F$ is then determined by further minimizing the criterion $E_{c_1c_2}$ with respect to the input $f$.

## 3 Optimal Solutions of $h$ and $f$

To derive the optimal solutions of $h$ and $f$, we apply an iterative technique to minimize the criterion function $E_{c_1c_2}$ subject to the basic constraints on $h$ and $f$. In each iteration during self-organization, there are four steps, consisting of finding the best-matching candidate $h(m,n) = w_{c_1c_2}(m+c'_1, n+c'_2)$ for system identification, adapting all the synapses with normalization $w_{ij}$, determining the optimal estimate of system input $f$, and adjusting the learning parameters.

We may use an alternative criterion (2) instead of using (3) to find the best-matching neuron (see, for example, [14]). The criterion function $e(w_{ij}, f)$ in (2) achieves its minimum when $w_{ij}(m+i', n+j') * f(m,n)$ matches $g(m,n)$, for $(m,n) \in F$. With this criterion function, the matching step determines the winning neuron $(c_1, c_2)$ whose synapse matrix $w_{c_1c_2}$ is the best-matching candidate for system identification. The neuron $(c_1, c_2)$ is decided by

$$(c_1, c_2) = \arg\min_{(i,j)} \{e(w_{ij}, \hat{f})\}, \quad \text{for } (i,j) \in N, \tag{4}$$

where the late estimated $\hat{f}$ is fixed in this step. The rule (4) is applied in all our iteration steps to decide $(c_1, c_2)$.

After the winning neuron $(c_1, c_2)$ is found, the adaptation step updates the synapses of neurons with different updates by

$$\begin{aligned} w_{ij}^{\text{new}}(m+i', n+j') &= w_{ij}(m+i', n+j') \\ &+ \alpha \mathcal{H}_{(c_1,c_2),(i,j)} \left( \frac{g(m,n)}{w_{ij}(m+i', n+j') * f(m,n)} * f(-m,-n) \right), \\ &\text{for } (i,j) \in N,\ (m,n) \in H, \end{aligned} \tag{5}$$

where $w_{ij}^{\text{new}}$ denotes the updated synapse matrix of $w_{ij}$ for next iteration and $\alpha$ is the learning rate. In (5), both $\alpha$ and $\mathcal{H}$ are functions of the iteration time. Note that the adaptation rule (5) for $w_{ij}$ is derived by adapting the synapses in the negative gradient direction to minimize the criterion function $E$ defined in (3). The term within the bracket $(\cdot)$ of (5) is obtained by calculating

$$-\frac{\partial e}{\partial w_{ij}(m+i', n+j')}.$$

To reduce the computational load, we may first adapt the synapses of the winning neuron $(c_1, c_2)$, i.e., $w_{c_1c_2}$, according to the rule (5). After the $w_{c_1c_2}$ is updated, the synapses of the other neurons are adjusted using

$$\begin{aligned} w_{ij}^{\text{new}}(m+i', n+j') &= w_{ij}(m+i', n+j') \\ &+ \alpha \mathcal{H}_{(c_1,c_2),(i,j)} \Big( w_{c_1c_2}^{\text{new}}(m+c'_1, n+c'_2) - w_{ij}(m+i', n+j') \Big), \\ &\text{for } (i,j) \in N,\ (m,n) \in H. \end{aligned} \tag{6}$$

The adaptation rule (6) is utilized to support the identification operation (5) in the self-organizing network for a general sense [14]. When using the rule (6), the adapted synapses will have the similar results as those using the rule (5). When we use (4) instead of finding the neuron with minimum (3), we choose a Dirac delta function as the function $\mathcal{H}$. To be consistent, the $\mathcal{H}$ in (5) and (6) should be a Dirac delta function. This kind choice of $\mathcal{H}$ in (5) and (6) will cause difficulty in convergence. For the convergence in the learning, we use a well-behaved neighborhood function $\mathcal{H}$ in (5) and (6). This inconsistency will be diminished in the final convergence stages.

Because of the basic constraints on $h$, the normalization of all synapse matrices $w_{ij}$ is always necessary. After each adaptation, the synapse matrices are then adjusted by

$$w_{ij}^{\text{new}}(m+i', n+j') = \frac{w_{ij}(m+i', n+j')}{\sum_{(m',n')\in H} w_{ij}(m'+i', n'+j')}, \quad \text{for } (i,j)\in N,\ (m,n)\in H, \tag{7}$$

where the denominator sums up all the components in the matrix $[w_{ij}]$ for normalization. Note all $w_{ij}(m+i', n+j')$ for $(m,n)$ outside $H$ are set to zeros and are omitted in our algorithm. Because of the non-negativity of $g(m,n)$, $f(m,n)$, and $w_{ij}(m+i', n+j')$ in (5) and (6), the normalized value $w_{ij}^{\text{new}}(m+i', n+j')$ will keep non-negative.

After the normalized synapse matrix, $w_{c_1c_2}$, as the best-matching candidate for the system function is determined, the components of function $h$ are assigned to those corresponding components of $w_{c_1c_2}$ by $\hat{h}(m,n) = w_{c_1c_2}(m+c'_1, n+c'_2)$, $(m,n)$ in $H$. We then further evaluate the reconstructed system input $f$ for consistency with using (4) and we minimize the criterion function $e(w_{c_1c_2}, f)$ with respect to $f$ instead of the function $E_{c_1c_2}$. The new estimated input $f$ is

$$f^{new}(m,n) = f(m,n) + \beta\left(\frac{g(m,n)}{h(m,n)*f(m,n)} * h(-m,-n)\right), \quad \text{for } (m,n)\in F, \tag{8}$$

where $\beta$ is the adjusting rate. The term within the bracket $(\cdot)$ in (8) is obtained by calculating

$$-\frac{\partial e}{\partial f(m,n)}.$$

Following the adaptation in (8), the value of $\hat{f}(m,n)$ is normalized by

$$f^{\text{new}}(m,n) = \frac{f(m,n)}{\sum_{(m',n')\in F} f(m',n')}, \quad \text{for } (m,n)\in F, \tag{9}$$

The selection of the proper parameters for the learning process is usually determined by experiences. In general, the learning parameters $\alpha$ and $\beta$ are non-increasing function of iteration time. The effective range of the neighborhood function $\mathcal{H}$ diminishes during the self-organizing process. Note that when the network converges, all center excitation parts of $w_{ij}$ will represent the shifted blur function and have the similar shape approximately.

# 4 Simulation Results

The self-organizing approach to both system identification and input reconstruction is applied to process blurred images. Simulation results are presented in this section. Figure 2 and 3 show the original image (a $256 \times 256$ 8-bit photograph) and the blurred image, respectively. In Figure 3, the image is blurred by a $7 \times 7$ blur function which has a geometrically decreasing factor 0.7 from the center. The blurred image is also degraded by white noise at 30 dB SNR. The procedures to reconstruct the blurred image are summarized in Figure 4.

In the self-organizing network, there are $32 \times 32$ ($N_1 = 32$ and $N_2 = 32$) neurons with $32 \times 32 \times 256 \times 256$ synapses, each neuron has a $256 \times 256$-dimensional synapse matrix which resembles the $256 \times 256$ shift blur function on the output image. Note that in our algorithm we do not need $256 \times 256$ synapse matrix for each neuron. We use the center excitation parts of the shifted blur functions which are $11 \times 11$ synapse matrices instead of using $256 \times 256$ synapse matrices to implement the identification. The values of $i'$ and $j'$ are set as $i' = 8 \times (i-1) + 5$ and $j' = 8 \times (j-1) + 5$, respectively, where $i = 1 \ldots N_1$ and $j = 1 \ldots N_2$. The initial synapses are set to variously normalized 2D shifted Gaussian functions. The total iteration times are 800.

Figure 5 and 6 show the values of averaged $E_{c_1c_2}$ over the $32 \times 32$ neurons and $e(w_{c_1c_2}, \hat{f})$ at different iterations, respectively. Simulation results show that this approach converges in appropriate number of iterations. The reconstructed image by this approach is displayed in Figure 7. Figure 8 shows the center region ($11 \times 11$) of the best estimated blur function in the self-organizing network.

From the results, this approach appears to be potential to solve the general deconvolution problem with the few constraints, to find the optimal solutions of the system function and the reconstructed input, to avoid trapping in local minima, and to be robust to the noise and the overestimation of blur function. Simulation results demonstrate the performance of this approach.

## References

[1] Stockham, Jr., T.G., Cannon, T.M., and Ingebretsen, R.B., "Blind deconvolution through digital signal processing," *Proceedings of the IEEE*, vol. 63 (4), pp. 678-692, 1975.

[2] Kundur, D. and Hatzinakos, D., "Blind image deconvolution," *IEEE Signal Processing Magazine*, vol. 13 (3), pp. 43-64, 1996.

[3] Tekalp, A.M., Kaufman, H., and Woods, J., "Identification of image and blur parameters for the noncausal blurs, " *IEEE Trans. ASSP*, vol. 34, pp. 963-972, 1986.

[4] Biemond, J., van der Putten, F.G., and Woods, J.W., "Identification and restoration of images with symmetric noncausal blurs," *IEEE Trans. Circuits System*, vol. 23 (3), pp. 385-394, 1988.

[5] Katsaggelos, A.K., "Iterative image restoration algorithms," *Optics Engineering*, vol. 28 (7), pp. 735-748, 1989.

[6] Ayers G.R., and Dainty J.C., "Iterative blind deconvolution method and its application," *Optics Letters*, vol. 13 (7), pp. 547-549, 1988.

[7] McCallum, B.C., "Blind deconvolution by simulated annealing," *Optics Communications*, vol. 75 (2), pp. 101-105, 1990.

[8] Kundur, D. and Hatzinakos, D., "A novel recursive filtering method for blind image restoration," *IASTED Int. Conf. on Signal and Image Proc.*, pp. 428-431, 1995.

[9] Wiggins, R.A., "Minimum entropy deconvolution," *Geoexploration*, vol. 16, pp. 21-35, 1978.

[10] Kohonen, T., *Self organization and Associative memory*, Springer-Verlag, Berlin, 1984.

[11] Kullback, S., *Information Theory and Statistics.* Dover, New York, 1968.

[12] Haykin, S. "Blind equalization formulated as a self-organizing learning process," *Proc. the 26th Asilomar Conf. on Sig., Syst., and Comp.*, CA, 1992.

[13] Tolat, V.V., "An analysis of Kohonen's self-organizing maps using a system of energy functions," *Biol. Cybern.*, vol. 64, pp. 155-164, 1990.

[14] Kohonen, T., "Generalizations of the Self-Organizing Maps," *Int. Joint Conf. on Neural Networks*, Japan, pp. 457-462, 1993.


**Figure 2. The original image, a 256 $\times$ 256 8-bit photograph of the Jupiter, for simulations.**

**Figure 3. The image blurred by a 7 $\times$ 7 blur function and degraded by noise at 30 dB SNR.**

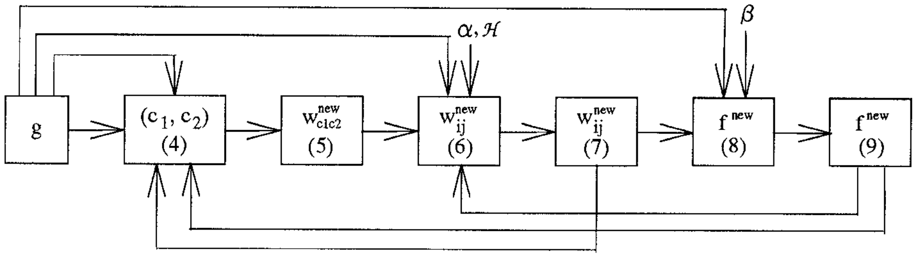


**Figure 4. The system chart for the reconstruction of blurred image.**

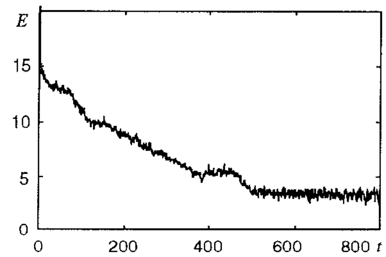


**Figure 5. The values of averaged $E$ for 800 iterations during the deconvolution process by the proposed approach.**

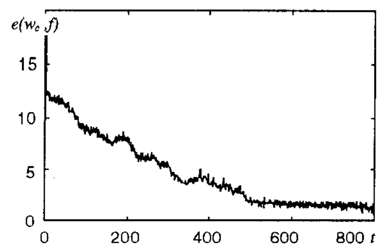


**Figure 6. The values of $e(w_{c_1c_2}, f)$ for 800 iterations during the deconvolution process by the proposed approach.**

**Figure 7. The reconstructed image from the blurred image in Figure 3 by the proposed approach.**

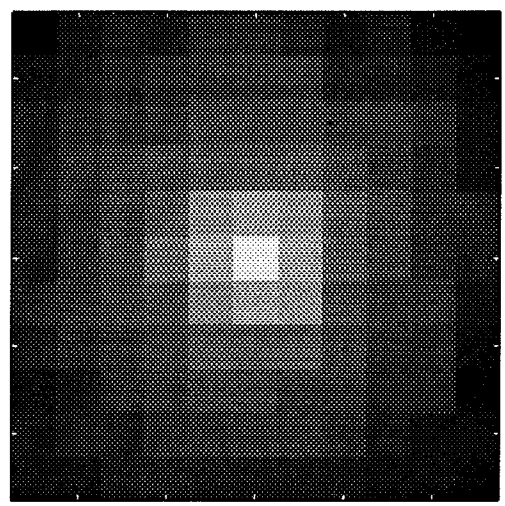

**Figure 8. The center region (11 $\times$ 11) of the best estimated blur function in the self-organizing network.**