

Image representation by self-organizing conformal network

Wen-Pin Tai¹,
Cheng-Yuan Liou²

¹Department of Computer Science, Chinese Culture University, 10764, Taipei, Taiwan, ROC

²Department of Computer Science and Information Engineering, National Taiwan University, 10764, Taipei, Taiwan, ROC
e-mail: tai@staff.pccu.edu.tw,
cyliou@csie.ntu.edu.tw

Conformal mappings are incorporated into the self-organization model to represent images harmonically. This network is used to partition an image into quadrilateral regions, where each region contains similar features. We then map each region to a corresponding square region to unify information representation and facilitate computations. This mapping is constructed to preserve spatial information while complying with the conformal property of the network. An approximated image in each square region provides us with an effective representation of the image in both modeling and compression applications. This approach has been particularly developed for large continuous images.

Key words: Neural network – Image modeling – Self-organizing network – Conformal mapping – Image compression

Correspondence to: W.-P. Tai

1 Introduction

Research in image representation has attempted to construct an image system that models the essential components, in spatial or transformed form, of an image. Such a system explores the inherent relationships among image data, perceives the redundancy among them, and determines an appropriate organization for them. The need for general and effective modeling has prompted much research into the neural network approach to explore representations for various applications, including, for example, image compression [1, 2], image segmentation [3], and surface modeling [4]. In this work, a self-organizing network (SON) [5] with conformal mapping techniques for images was studied.

SON has the ability to learn visual information from an image and to form an ordered map that conformally (harmonically) preserves the essential features of the image [6]. This network usually acts as a vector quantizer to organize sensory inputs into ordered clusters on a map. Such maps can provide descriptions of the inputs in terms of a series of quantized cluster centers or representations. Image data usually have varying properties that are continuous among neighboring pixels. This map can roughly approximate continuous inputs in its clustering centers. This limits the application of SON in image analysis.

We have devised an approach based on SON which characterizes image data continuously with harmonic representation. As shown in Fig. 1, this network is utilized to place a partition for an image. The spaces in this partition are adapted according to the image data fed into this network. Based on this, the image is partitioned into connected quadrilateral regions, each containing similar features. This network conformally preserves the features of the image data [6] in a global sense. To save neurons and comply with this global property, we use conformal mappings to transform these quadrilateral regions into a square system, formed by square regions, to expand the local details uniformly. A conformally warped representation of the image in the square system will be obtained. This kind of representation provides us with a uniform and effective description of the image and roughly equalizes the amount of information in various regions. The representation in the square system will facilitate further mapping computations. Figure 1 shows application of the network to image compression.

To transform the data in the quadrilateral regions into data in a square system as in phase II shown in Fig. 1,

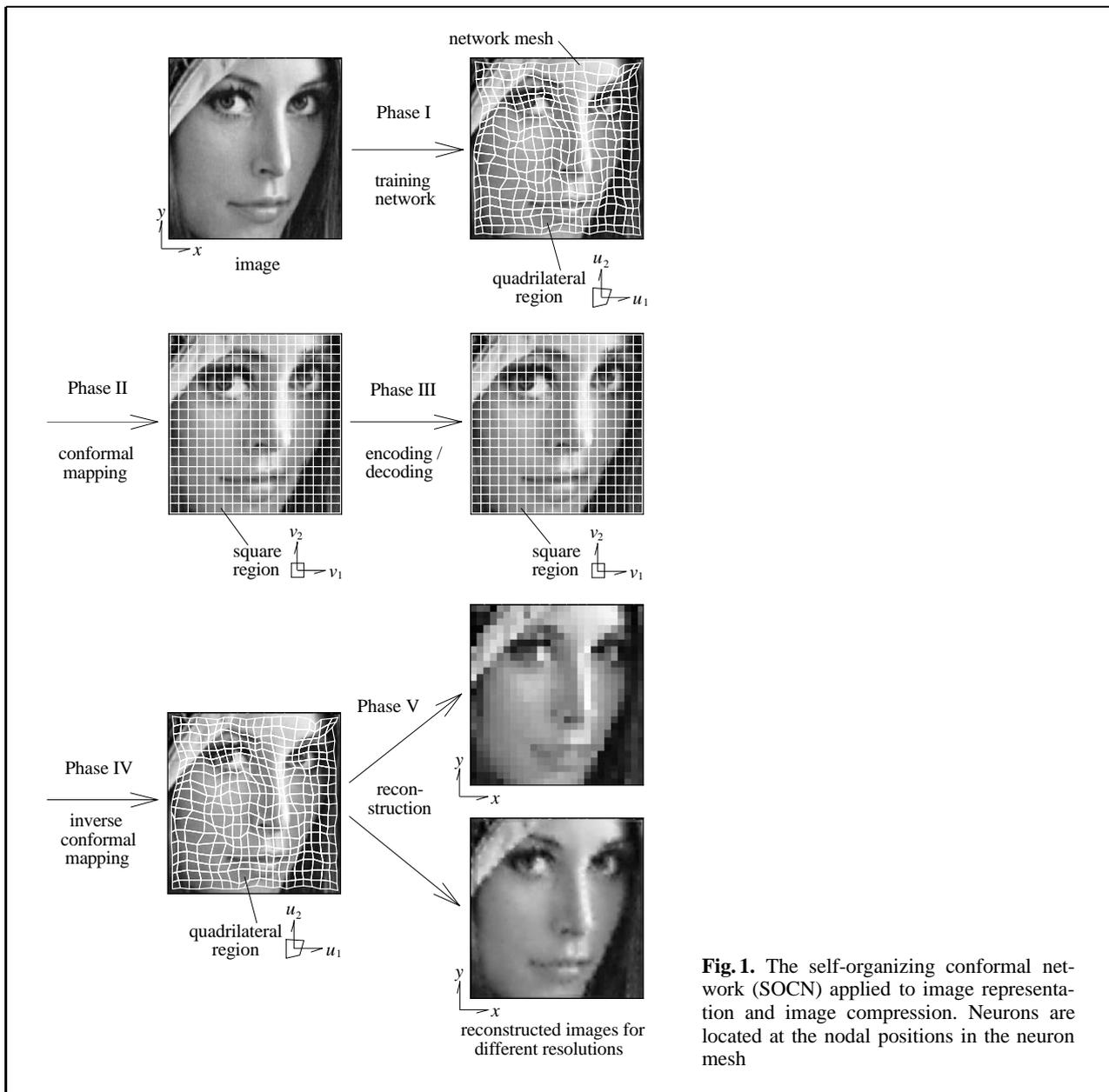


Fig. 1. The self-organizing conformal network (SOCN) applied to image representation and image compression. Neurons are located at the nodal positions in the neuron mesh

we need to construct continuous mapping functions that preserve the image features. These mappings must be one-to-one and analytic so that no information will be lost. Considering the local isotropic property and the conformal property of SON, we can construct conformal mappings (see, for example, [7]) to map these quadrilateral regions to their corresponding square regions. Conformal mappings have been studied as models of image representa-

tions in the brain [8–10], for which image warping with *conformal preservation* has been proposed to model the cortical representation of the image formed on the retina. Experiments in the visual brain use regular shapes to display the mappings [9]. We have further designed a square system to simplify and unify the computations. In the proposed approach, conformal mapping techniques are used to construct a harmonically warped representation in

the square system. Effective numerical techniques are discussed for these mappings.

During the adaptation process in phase I, the partition configuration evolves continuously and conformally. The converged network provides a scaled partition for the input image. The feature contents scale the partition. This network partitions the image into connected quadrilateral regions. Mappings from quadrilateral regions to square regions in phase II are devised so as to satisfy conformality conditions. A warped image representation in a square system is obtained by means of such conformal mappings. Note that conformal mapping satisfies the *harmonic property* [7, 17] used in image processing. We can obtain visual information that effectively characterizes an image and preserves its topological features can be obtained using the network and mapping. Conformality is achieved using SON and mappings in both the global and local senses. Since it is costly to build a network with dense neurons, we use arrayed sparse neurons in the network and use conformal mappings to fill the local detailed space.

Application of the network to image compression is discussed in this work. Based on this approach, image data are partitioned into neighboring quadrilateral regions by the network. Then, the image in each region is mapped to a square region in a square system. Image features within each square region are similar and uniformly distributed. Approximations can be applied to remove the redundancy of the image in each square region. The approximated harmonic function, which satisfies the Laplace's equation and meets the boundary conditions of each square region, is used as the reduced representation. This function can be used to obtain reconstructed images with arbitrary resolutions. High compression ratios for continuous images can be achieved. We give simulation results to show the performance obtained using this approach.

This work is organized as follows. In Sect. 2, a general image model is defined. SON is employed to place a scaled partition system on an image. Quadrilateral regions in this partition, square regions in the square system, and the relationships between them are illustrated. We also discuss self-organizing conformal network (SOCN) and analytic mappings. Numerical mapping techniques are included. In Sect. 3, this approach is applied to image compression. Section 4 presents many simulations carried out to verify the performance of this approach, and the results are discussed.

2 Adaptively scaled partition system for an image

2.1 The image representation

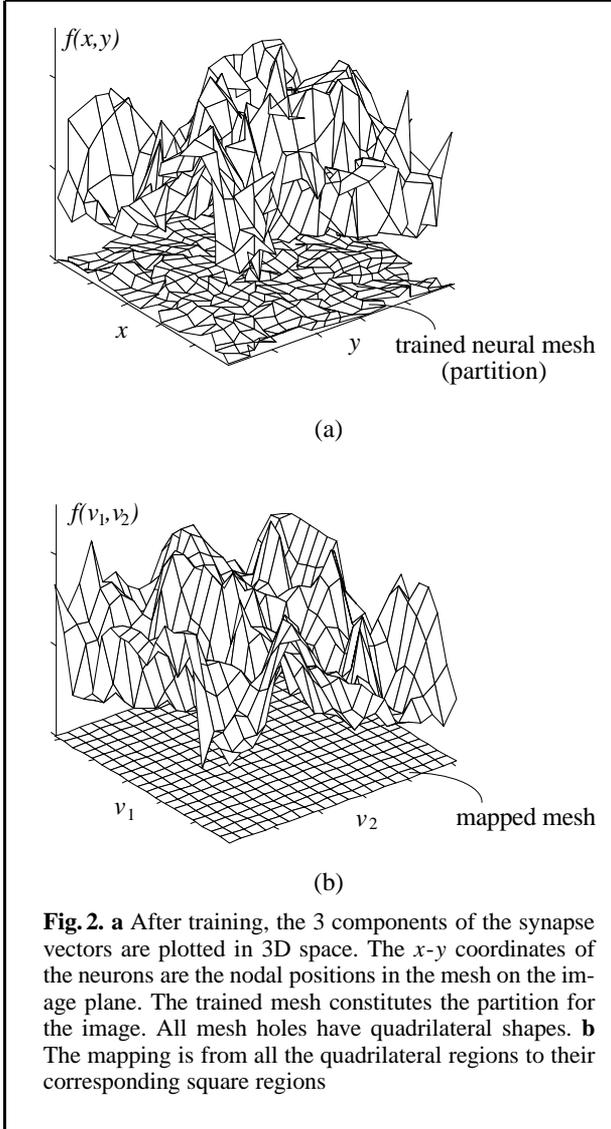
An image can be modeled as a 2D intensity function. Let $f(\mathbf{p})$ denote this function, in which the value of f at a point coordinate $\mathbf{p} = (x, y) \in P$, $P \subseteq R^2$ gives the intensity of the image at this point. The function $f(\mathbf{p})$ may be either continuous along the spatial coordinate \mathbf{p} or digitized spatially. The amplitude of $f(\mathbf{p})$ may be either shaded continuously or quantized.

The coordinate $\mathbf{p} = (x, y)$ and the function $f(\mathbf{p})$ are usually sampled as inputs for further data processing. Consider the input set X , $X \subseteq R^3$ from an image. The sample input $\mathbf{x} = (\mathbf{p}, f(\mathbf{p})) \in X$ is the image feature vector at the coordinate (x, y) . The main concern in image processing is to explore the relationships among these inputs and to determine good representations of these inputs for further analyses.

Since the local image data in nearby pixels are often highly correlated, a whole image can be partitioned into local regions with different center features. Data in the various regions should be continuously decorrelated. This partition can be constructed adaptively according to the image data. To construct this partition, we use SON as a partition system with its adaptive ability to scale partition spaces. This network has a 2D neuron array arranged initially in a regular square topology. Each region can be defined by linking four neighborhood neurons as shown in Fig. 1. This region changes in shape during the process of self-organizing adaptation and evolves into a quadrilateral region. Each quadrilateral region encloses a local image that contains similar features. The boundaries of these regions constitute a partition. The evolution of this partition is now presented.

2.2 Partitioning the image using SON in phase I

To partition the image, we apply SON to process the inputs and to generate the partition. To process the samples from input set X , a set of neurons arranged in a 2D plane with regular rectangular topology is used (see Fig. 1). The synapse vector corresponding to the neuron j is denoted



by $\mathbf{w}_j \in W$, where $W \subseteq R^3$. The first two and the third components of \mathbf{w}_j are set to be adapted to the coordinate and the intensity of the image input, respectively. Learning can be done in the iterations of the best-matching process and the synapse adaptation process. For each iteration, an input $\mathbf{x} \in X$ is randomly selected. The best-matching process determines the winning (or best-matching) neuron c whose synapse vector \mathbf{w}_c is the closest vector to the input \mathbf{x} for all the vectors in W . In this process, the euclidean distance measure is used to measure the match between the input and the synapse vectors. The best-matching neuron can be decided by

means of the minimum distance criterion as follows:

$$\|\mathbf{x} - \mathbf{w}_c\| = \min_{\mathbf{w}_j \in W} \|\mathbf{x} - \mathbf{w}_j\|. \quad (1)$$

After the winning neuron c is found, the adaptation process adjusts the synapse vectors of neurons with varying amounts of updating by

$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \alpha(n) \times h_{c,j}(n) \times [\mathbf{x} - \mathbf{w}_j(n)], \quad \mathbf{w}_j \in W, \quad (2)$$

where n is the time step or iteration number, α is the learning rate, and $h_{c,j}$ is the neighborhood function for the neurons c and j . Both α and $h_{c,j}$ are functions of the time step n . After iterations for different sampled inputs, an ordered representation that approximates the input space can be obtained.

As shown in Fig. 2a, a converged network provides a scaled partition of the image. Synapse vectors of the network represent the feature centers of the image. The first two components of neighboring synapse vectors are the coordinates of the four vertices of a corresponding quadrilateral region. During the self-organizing process, the shapes of these regions are varied, following the synapse adaptation. The converged network partitions the image into a disjoint collection of quadrilateral regions. Each region contains similar features and roughly the same amount of information. This partition completes the process in phase I. Note that many well-developed techniques are available to speed the convergence. We use the formal algorithm.

Figure 2b displays the mapped image in the square system. Using the conformal mapping techniques, discussed in Sect. 2.3, representations in regularly square regions can be obtained.

2.3 Conformal mapping between quadrilateral regions and square regions in phase II

To obtain a uniform representation, we map the image in each quadrilateral region to its corresponding square region. We use conformal mapping functions to preserve the feature properties of the image data and to avoid information loss.

For a 2D image, we perform mapping using techniques employed in complex analysis. We reformulate the spatial coordinate $\mathbf{p} = (x, y) \in R^2$ into its

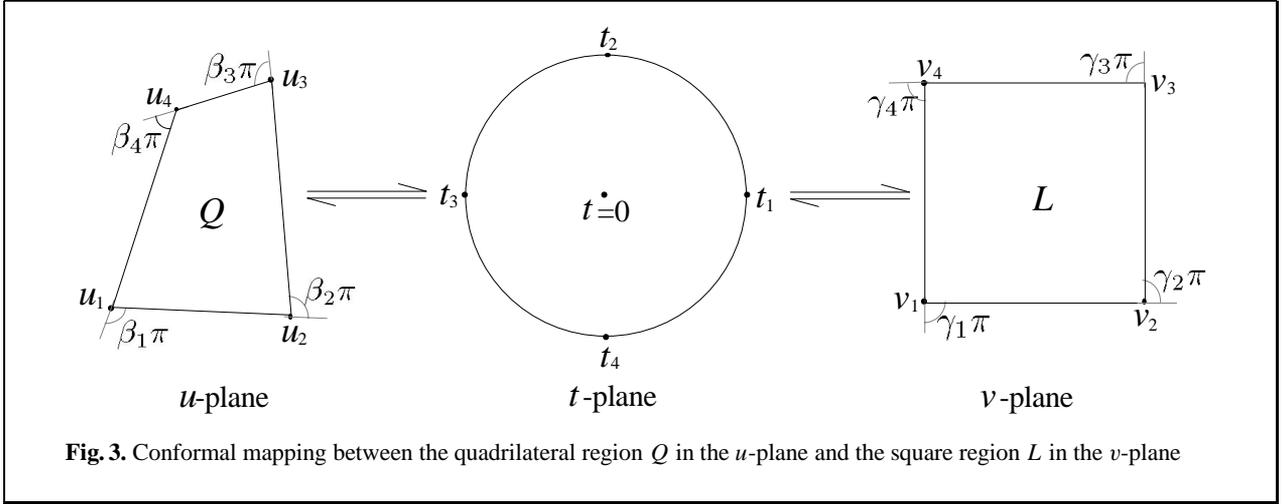


Fig. 3. Conformal mapping between the quadrilateral region Q in the u -plane and the square region L in the v -plane

complex expression $(x + iy)$. Let the conformal mapping from a quadrilateral region Q to a square region L be $K_{Q,L}$, where $K_{Q,L} : u \mapsto v$, $u \in Q$ and $v \in L$ are complex variables. With the function K , the one-to-one correspondence between these two regions can be defined. The Schwarz-Christoffel transformation (see, for example, [11]) can be used to construct the mapping between quadrilaterals. Note that a square is also a quadrilateral.

Consider a quadrilateral Q with vertices labeled u_j , $1 \leq u_j \leq 4$, in the u -plane and a square L with vertices labeled v_j , $j = 1 \dots 4$, in the v -plane. Following the Schwarz-Christoffel transformation, we transform the boundary ∂Q of the quadrilateral region Q along with its vertices into the outer circle of a unit disk in the t -plane and then transfer the circle to the boundary ∂L of the square region L . The unit disk in the t -plane is used as an intermediate set to aid construction of the mapping. Figure 3 shows the mappings between the complex planes.

Two types of mappings are shown in Fig. 3. For the first type, we consider mappings from the unit disk in the t -plane to corresponding regions in the u -plane and v -plane. Let $\beta_j\pi$, $-1 \leq \beta_j < 1$, be the exterior angle of the quadrilateral region Q at u_j , and let $\gamma_j\pi$, $-1 \leq \gamma_j < 1$, be that of the square region L at v_j . For quadrilaterals, we have the relationships between $\beta_j\pi$ and $\gamma_j\pi$, which are:

$$\sum_{j=1}^4 \beta_j = 2 \quad \text{and} \quad \sum_{j=1}^4 \gamma_j = 2, \quad (3)$$

respectively. The Schwarz-Christoffel formula defines the mapping from t to u as

$$u = u_c + C_1 \int_0^t \prod_{j=1}^4 \left(1 - \frac{t'}{t_j}\right)^{-\beta_j} dt' \quad (4)$$

and the mapping from t to v as

$$v = v_c + C_2 \int_0^t \prod_{j=1}^4 \left(1 - \frac{t'}{t_j}\right)^{-\gamma_j} dt', \quad (5)$$

where t_j , $j = 1 \dots 4$, are the designed points on the boundary of the unit disk, and C_1 , C_2 , u_c , and v_c are the complex parameters of the mappings. v_c is the center of the square. $C_2 = \sqrt{2} + i\sqrt{2}$, is a constant in this mapping. The points $\{t_j, j = 1 \dots 4\}$ are designed to match the four vertices of each region with computational efficiency. In all our simulations, we set $t_1 = 1$, $t_2 = i$, $t_3 = -1$, and $t_4 = -i$.

In our simulations, we used the following formula for (4):

$$u = u_c + C_1 \int_0^t \prod_{j=1}^4 \left(1 - \frac{t'}{t_j}\right)^{-\beta_j} dt' \quad (6)$$

$$\approx u_c + \frac{C_1 t}{2q} \sum_{k=1}^p \sum_{l=1}^q g_k \cdot f_{k,l}, \quad (7)$$

$$f_{k,l} = (1 - y_{k,l})^{-\beta_1} (1 + iy_{k,l})^{-\beta_2} \times (1 + y_{k,l})^{-\beta_3} (1 - iy_{k,l})^{-\beta_4}, \quad (8)$$

$$y_{k,l} = \frac{t}{q} l + \frac{t}{2q} b_k - \frac{t}{2q}. \quad (9)$$

The interval $[0, t]$ was divided into q subintervals. The value of q was set to 12. The length p of the base points b_k and weight factors g_k was set to 12. The Gauss rule for numerical integration was used to determine the values of b_k and g_k , $k = 1 \dots p$. We set $p = q = 12$ in all our simulations. The four parameters in (4) and (5) are determined as follows. v_c is the center of the square. The mappings satisfy the complex conditions at the four points

$$u_k - u_c = C_1 \int_0^{t_k} \prod_{j=1}^4 \left(1 - \frac{t'}{t_j}\right)^{-\beta_j} dt', \quad k = 1 \dots 4, \quad (10)$$

from t_1, t_2, t_3 , and t_4 to u_1, u_2, u_3 , and u_4 , respectively, and

$$v_k - v_c = C_2 \int_0^{t_k} \prod_{j=1}^4 \left(1 - \frac{t'}{t_j}\right)^{-\gamma_j} dt', \quad k = 1 \dots 4, \quad (11)$$

from t_1, t_2, t_3 , and t_4 to v_1, v_2, v_3 , and v_4 , respectively. Note that the vertices $\{u_j, j = 1 \dots 4\}$ and $\{v_j, j = 1 \dots 4\}$ are known vertices. The integration in each case is evaluated by means of the compound Gauss quadrature (see, for example, [13]). Then, the other two parameters (u_c and C_1) can be estimated from numerical solutions of (10) and (11). The mappings from the disk to the square L , or from the square L to the disk, can be prepared in advance (offline preparation).

For the second type of mappings (Fig. 3), we consider conformal mappings from the regions in the u -plane and v -plane to the unit disk in the t -plane. These mappings can be solved with numerical methods. Inverting the Schwarz–Christoffel formulas of (4) and (5), we obtain

$$\frac{dt}{du} = \frac{1}{C_1} \prod_{j=1}^4 \left(1 - \frac{t}{t_j}\right)^{+\beta_j} \quad (12)$$

for the mapping from u to t and

$$\frac{dt}{dv} = \frac{1}{C_2} \prod_{j=1}^4 \left(1 - \frac{t}{t_j}\right)^{+\gamma_j} \quad (13)$$

for the mapping from v to t . The numerical Runge–Kutta method can then be applied to these bound-

ary value problems with the broken exponents β_j and γ_j . Good solutions can be obtained with iterative evaluation [12].

The conformal mapping $K_{Q,L}(u)$ can be obtained from u to t and from t to v successively. The inverse conformal mapping $K_{L,Q}^{-1}(v)$ can be obtained similarly from v to t and from t to u .

With this conformal mapping, the image in the quadrilateral region Q can be mapped to a warped image in the square region L . Let the input be $\mathbf{x} = (u, f(u)) \in Q$, where u is the point coordinate in the complex plane and $f(u)$ is the intensity at u . The corresponding mapped data $\mathbf{x}' = (v, f(u(v)))$ in L can be obtained with conformal mapping from u to v , i.e., $v = K_{Q,L}(u)$ as already described.

Since u is continuous in the spatial coordinate, this mapping can be applied to continuous images such as photos from cameras or fine art paintings. We devise this technique particularly for those paintings on curved surfaces in the projects. When we use exact mappings, there is no distortion during reconstruction of the original image from the warped image in the square regions. This approach is capable of processing both digital and continuous images. For a digital image, a point in the v -plane may not correspond to a pixel point in the u -plane. We record all the v coordinates that correspond to all the pixel point. We then interpolate the needed points in the v -plane, such as the grid points or the equal spaced points along the boundary, ∂L , of L . This interpolation is not necessary for continuous images. In all our simulations, we interpolated points along the boundary of L in the v -plane that were equally spaced along ∂L and included the four vertices of L for discrete images.

2.4 The quadrilaterals with modulus one

For any two quadrilateral regions, a conformal mapping between them may not exist. If one does exist, then these two quadrilaterals are conformally equivalent and have the same modulus. A square region has a modulus of 1 by definition. To unify the conformality property, we limit all the quadrilateral regions in the u -plane to a modulus of 1.

To transform a quadrilateral region into a square region with a modulus of 1, the quadrilateral must have modulus 1. Consider the unit disk in the t -plane. The coordinates of t_j , $j = 1 \dots 4$, are chosen so as to equally divide up the unit circle, i.e.,

$t_1 = 1$, $t_2 = i$, $t_3 = -1$, and $t_4 = -i$. We will now show how to obtain quadrilaterals with a modulus of 1 in SON.

Because the region L is a square, the mapping of (5) from t to v can be simplified as follows:

$$v = v_c + C_2 \int_0^t (1 - t'^4)^{-\frac{1}{2}} dt' \quad (14)$$

$$\approx v_c + \frac{C_2 t}{2q} \sum_{k=1}^p \sum_{l=1}^q g_k \cdot f_{k,l}, \quad (15)$$

$$f_{k,l} = (1 - y_{k,l}^4)^{-\frac{1}{2}}, \quad (16)$$

$$y_{k,l} = \frac{t}{q} l + \frac{t}{2q} b_k - \frac{t}{2q}, \quad (17)$$

$$C_2 = \sqrt{2} + i\sqrt{2}. \quad (18)$$

We set $p = q = 12$. The value of v_c in (14) is the center of the square.

During the self-organizing process, the shapes of the quadrilateral regions are dynamically reformed following (1) and (2). These quadrilaterals may not be conformally equivalent to a square whose modulus is 1. To obtain quadrilaterals with a modulus of 1 in the self-organizing process, we must further reform the quadrilaterals accordingly after each iteration. We modify the first two components, which are the coordinates of vertices and neurons, of all synapse vectors separately to move the quadrilateral vertices. We modify these coordinates for each quadrilateral Q following (1) and (2). The current quadrilaterals are first used in (4) to estimate u_c and C_1 . The vertices u_j of a quadrilateral Q are modified or relaxed by

$$u'_j = u_c + C_1 \int_0^{t_j} (1 - t')^{-\beta_1} (1 + it')^{-\beta_2} (1 + t')^{-\beta_3} \times (1 - it')^{-\beta_4} dt', \quad j = 1 \dots 4. \quad (19)$$

Relaxation of the network configuration using (19) is applied intensively during the final iterations to obtain a good modulus.

For a quadrilateral that contains x , we modify the vertices u_j , $j = 1 \dots 4$, in the complex expression, of

the quadrilateral into u'_j by means of

$$u'_j = u_c + C_1 \int_0^{t_j} (1 - t')^{-\beta_1} (1 + it')^{-\beta_2} (1 + t')^{-\beta_3} \times (1 - it')^{-\beta_4} dt' \quad (20)$$

$$\approx u_c + \frac{C_1 t_j}{2q} \sum_{k=1}^p \sum_{l=1}^q g_k \cdot f_{k,l}, \quad (21)$$

$$f_{k,l} = (1 - y_{k,l})^{-\beta_1} (1 + iy_{k,l})^{-\beta_2} (1 + y_{k,l})^{-\beta_3} \times (1 - iy_{k,l})^{-\beta_4}, \quad (22)$$

$$y_{k,l} = \frac{t_j}{q} l + \frac{t_j}{2q} b_k - \frac{t_j}{2q}, \quad (23)$$

$$j = 1 \dots 4, \quad (24)$$

where t_j , β_k , u_c , and C_1 are obtained using the conformal mapping process. The base points b_k and weight factors g_k , $k = 1 \dots p$, are determined using the Gauss rule for numerical integration. We used $p = 12$ and $q = 12$ in all our simulations. The Gauss quadrature method is used to determine the values of b_k and g_k .

When the network converges, the partition of the image into the quadrilateral regions with a modulus of 1 can be obtained approximately with this modification. Including (4) and (19) with (1) and (2) in each iteration, the algorithm not only adaptively explores the features of the inputs, but also preserves the conformal property in the network [6]. We name this network SOCN. Since it is costly to include the modification of (19) from the beginning of the evolution, we include modification during the later evolution stage. Modification using (19) still may not guarantee that the modulus will be one. We add a diagonal edge to partition the quadrilateral into two triangles whenever this quadrilateral does not meet this requirement. Mapping of two triangles can be accomplished by modifying the quadrilateral mapping shown in Fig. 3. This modification is done by merging any two neighboring nodes of the quadrilateral into one node.

3 Image compression in phases III, IV, and V

This proposed image representation in the square regions can be applied to different image analyses, particularly image compression. We discuss phases III, IV, and V in this section.

In image compression, Kohonen's SON has been applied with a great deal of success in the design of codebooks for vector quantization [1]. Instead, as a vector quantizer, our approach is devised to approximate the image in a warped coordinate system. With the conformal mappings, the warped image in the square regions preserves the visual information of the original image and provides a generalized representation of the image.

The image portion in each square region has a similar feature and uniform property. The redundant information in each region can be reduced. Considering the continuity of the mapped image in a square region L , we use the Laplace equation to model the mapped image in L . This is because the Laplace equation meets the conformality requirement. We can keep conformality consistency in all the approach. The image data, $\{f(u(v))\}$, where $v=v_1+iv_2$, $v \in L$, can be fitted or interpolated with a function $\phi(v_1, v_2) \approx f(u(v))$ that satisfies the Laplace equation

$$\frac{\partial^2 \phi}{\partial v_1^2} + \frac{\partial^2 \phi}{\partial v_2^2} = 0 \text{ in } L, \quad (25)$$

and the boundary conditions on ∂L . Note that a linear function of (v_1, v_2) , which is usually used for linear interpolation (or filtering), satisfies the Laplace (25). Sinusoidal functions are special solutions of (25). These functions are the bases of most modern compression techniques. General solutions of (25) are much richer and can provide many harmonic images. We now describe a simple way to obtain an approximated image in each region L . To compress the data in L , we sample $4M$ points (including the four vertices) along the boundary ∂L with equally spaced sampling. These $4M$ points may be interpolated from the mapped image $f(u(v))$. Note that this interpolation is not necessary for continuous images. We can always find exact mapping points of these $4M$ boundary points for a continuous image. The reason for using the boundary points at our disposal is that we can partition a large painting into subframes. Then, we compress each subframe. We connect the decompressed subframes to recover the original painting. This kind of compression requires an algorithm with seamless excellence. We require each subframe to match these boundary points exactly along the partition border. This requirement will guarantee seamless compression. To our knowledge, there exists no other exact seamless approach for partitioned images.

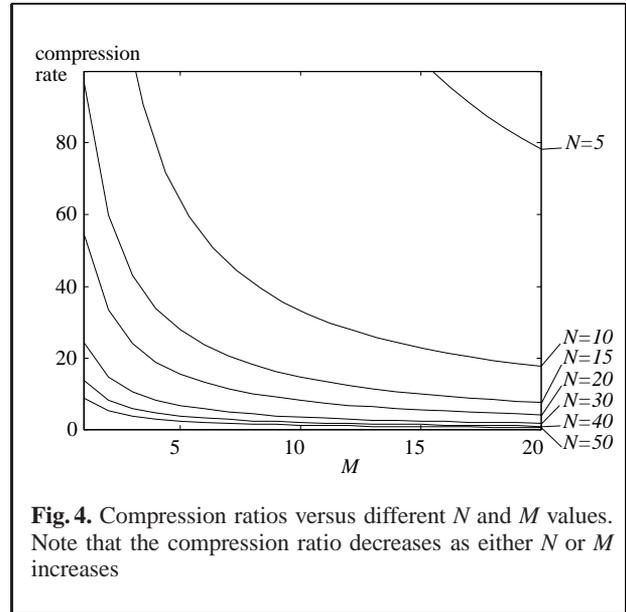
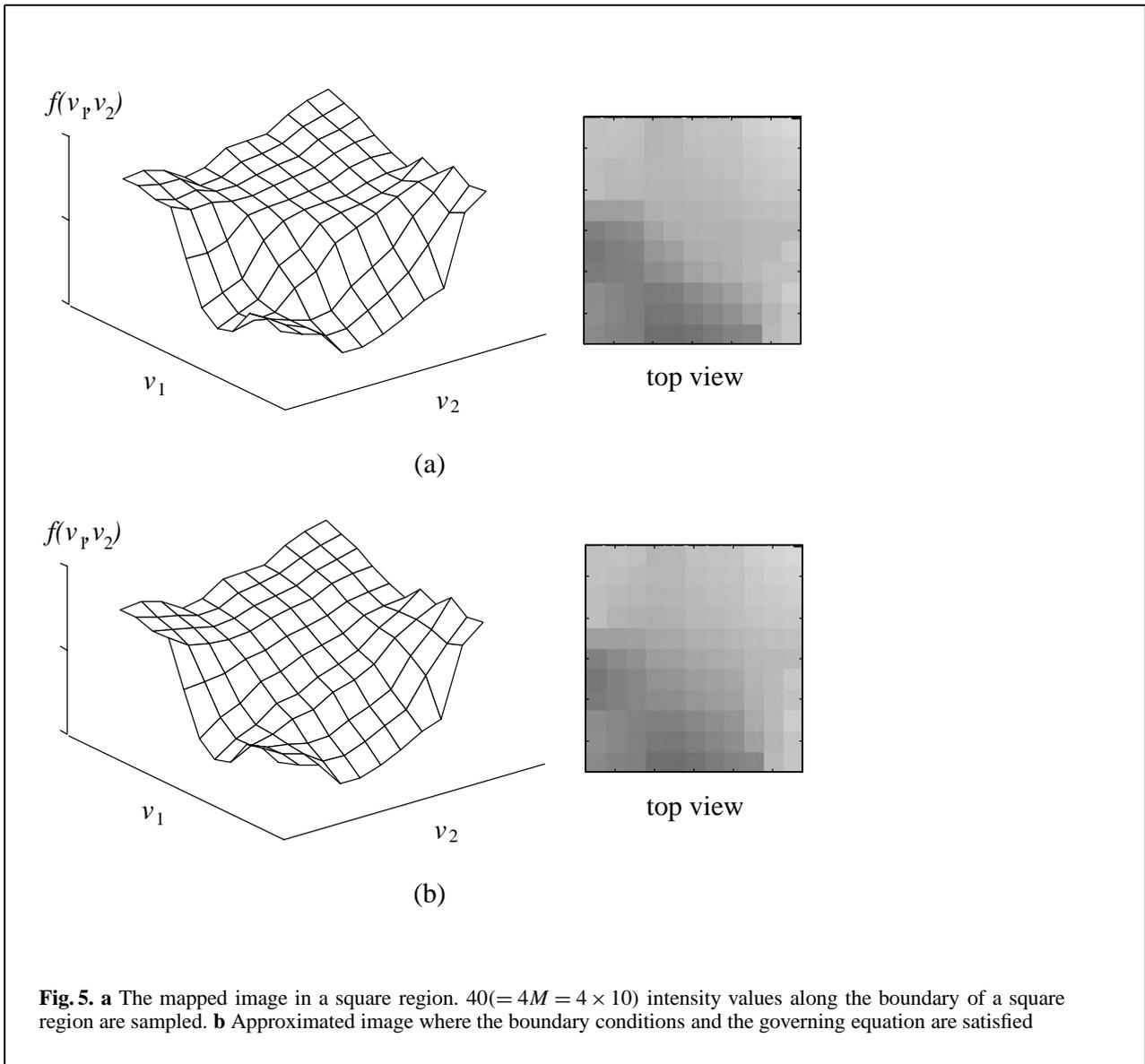


Fig. 4. Compression ratios versus different N and M values. Note that the compression ratio decreases as either N or M increases

For a network with $N \times N$ neurons, there are $N + N(N-1)(2M-1)$ data in the compressed representation. In addition, there are $3(N-1)^2$ and $4(N-1)^2$ real data for the exterior angles of the quadrilateral region Q and for the complex parameters, u_c and C_1 , of the conformal mappings from the u -plane to the t -plane, respectively. There are a total of $2 + 2(N-1)(MN + 3N - 3)$ real codes.

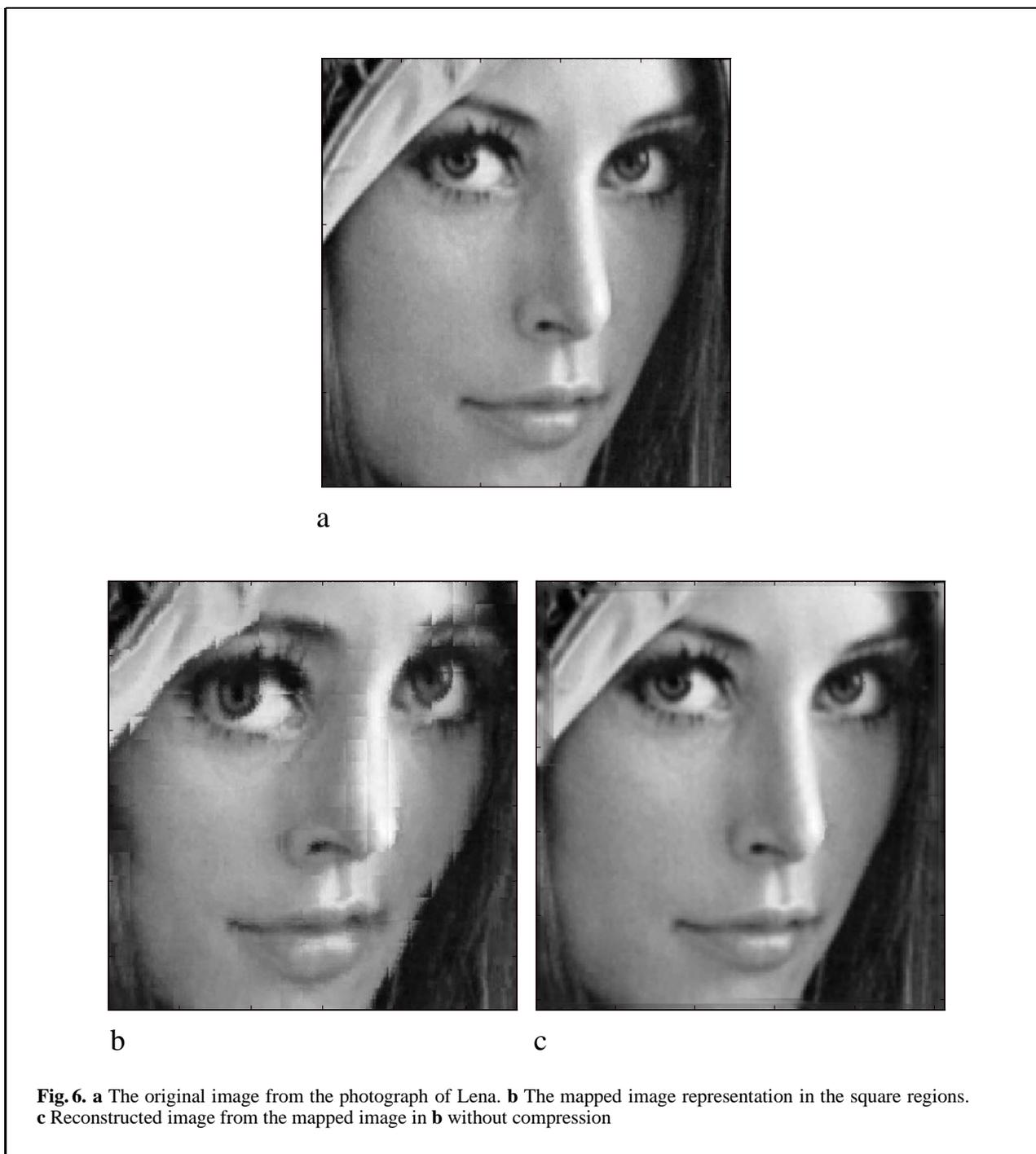
The $3(N-1)^2$ and $4(N-1)^2$ real parameters may be reduced for computational efficiency. These parameters are not necessary, and they can be calculated from the first two components of the synapses. These coordinate components have $2N^2$ real data. Thus, there are a total of $(N+1)^2 + 2MN(N-1)$ codes for image compression with this method. The compression ratio is higher than that of the previous coding method. Figure 4 shows the compression ratios for different N and M values. The choices of the N and M values depend on the level of difficulty of the image. Actually, we design these values case by case.

When a high resolution or continuous image is reconstructed with this approach, the value of M may be large. $4M$ boundary samples are used to reconstruct the $(M+1)^2$ grid points within a square region. Thus, the compression ratio is roughly equal to $\frac{(M+1)^2}{4M}$, which is linearly proportional to the value of M . M can be as high as we want for a continuous image. Usually, we set M greater than 255.



To reconstruct the image from these codes, we use numerical techniques described in [14] to solve this boundary value problem. We obtain an approximated image in each square region L that satisfies the governing (25) and has the $4M$ boundary values. Figure 5 shows an example in which $M = 10$ and the approximated image is solved to satisfy the boundary values. Note that there are many alternative ways to obtain $\phi(v_1, v_2)$. A delicate method is to place sources and sinks [15] within L . The potential fields of both sources and sinks satisfy the governing (25). The locations and

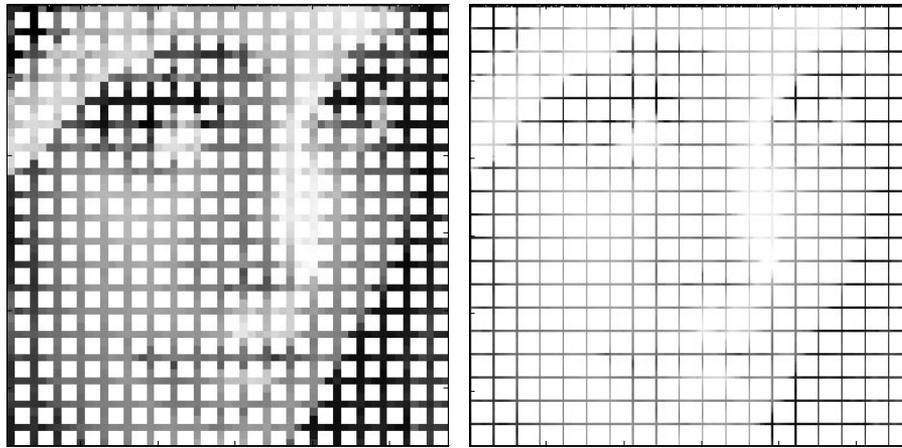
strengths of these sources and sinks are determined through approximation of $f(u(v))$ such that their combined potential field, $\phi(v_1, v_2)$, is close to $f(u(v))$ within L . This results in fine reconstructions. The computation cost is highly dependent on the algorithm used to obtain an approximated $\phi(v_1, v_2)$. We do not discuss this kind algorithm here. This completes the phase III operations. With the conformal mapping $K_{L,Q}^{-1}$ from the v -plane to the u -plane, the original image contained in the quadrilateral regions can be recovered in phases IV and V.



4 Simulations and discussion

Since our approach is designed for continuous images, simulation on a common digital image gives a sense of the approach's compression ca-

pability. Two-dimensional gray-scale images with 256×256 8-bit pixels were used in the tests. Since our approach is designed for continuous images, simulation on a common digital image will give a sense on the approach's compression capabil-



7a

7b



8a

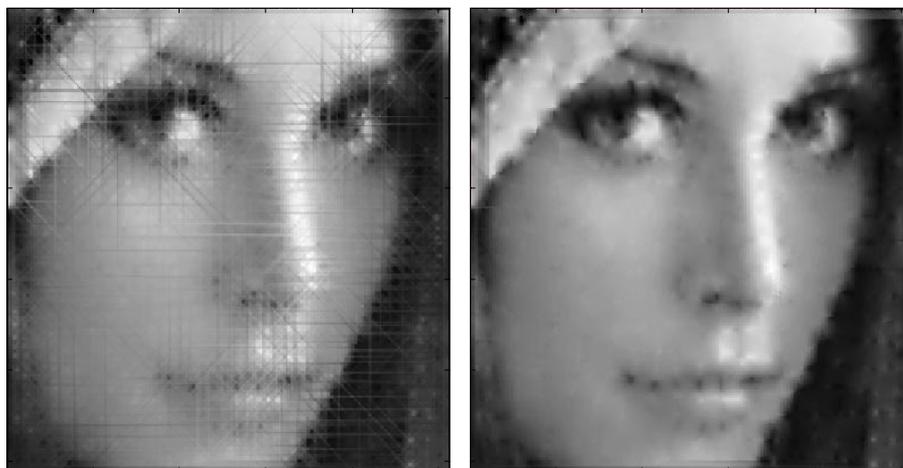
8b

Fig. 7a,b. Sampled intensity data along the square regions' boundaries for **a** the photograph with $M = 3$ and **b** the photograph with $M = 10$. The samples are plotted in the square system

Fig. 8a,b. Reconstruction of the images in square regions from the sampled data on the square regions' boundaries for **a** the photograph with $M = 3$ and **b** the photograph with $M = 10$

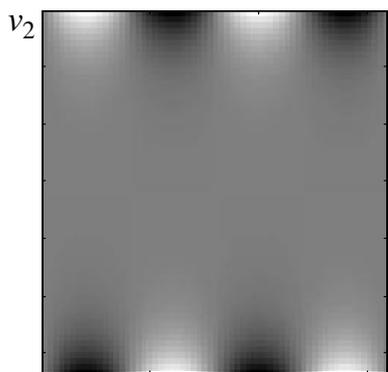
ity. The initial configuration of the network was a 20×20 ($N = 20$) square grid pattern, regularly arranged in the image space. Training parameters were the same for all the simulations. We set the total iteration number to 2000, the learning rate $\alpha(n) = 0.001^{n/2000}$, and the neighborhood function $h_{c,j}(n) = \exp(-d(c,j)^2/(\alpha(n) \times 20)^2)$, where $d(c,j)$ is the distance metric between the winning neuron c and the neuron j in the network plane.

A photograph of Lena was tested. The results are shown in Fig. 6. Figure 6a shows the original image, and Fig. 6b shows the corresponding image representation in the square regions. Figure 6c shows the reconstructed images obtained without compression by means of inverse conformal mappings of the image in Fig. 6b. Note that we did not apply any data reduction techniques to obtain Fig. 6c, which shows that the representation in Fig. 6b is a feasible and effective representation.

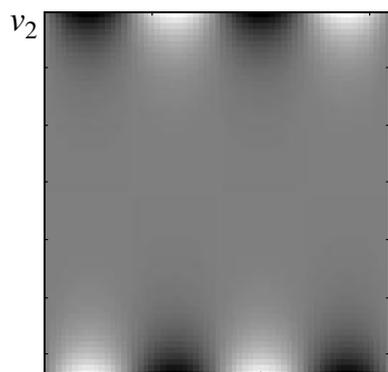


9a

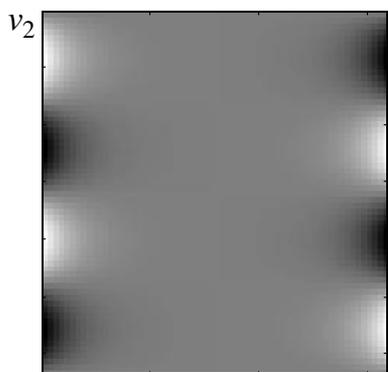
9b



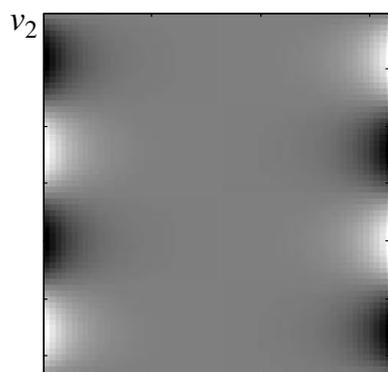
10a



10b



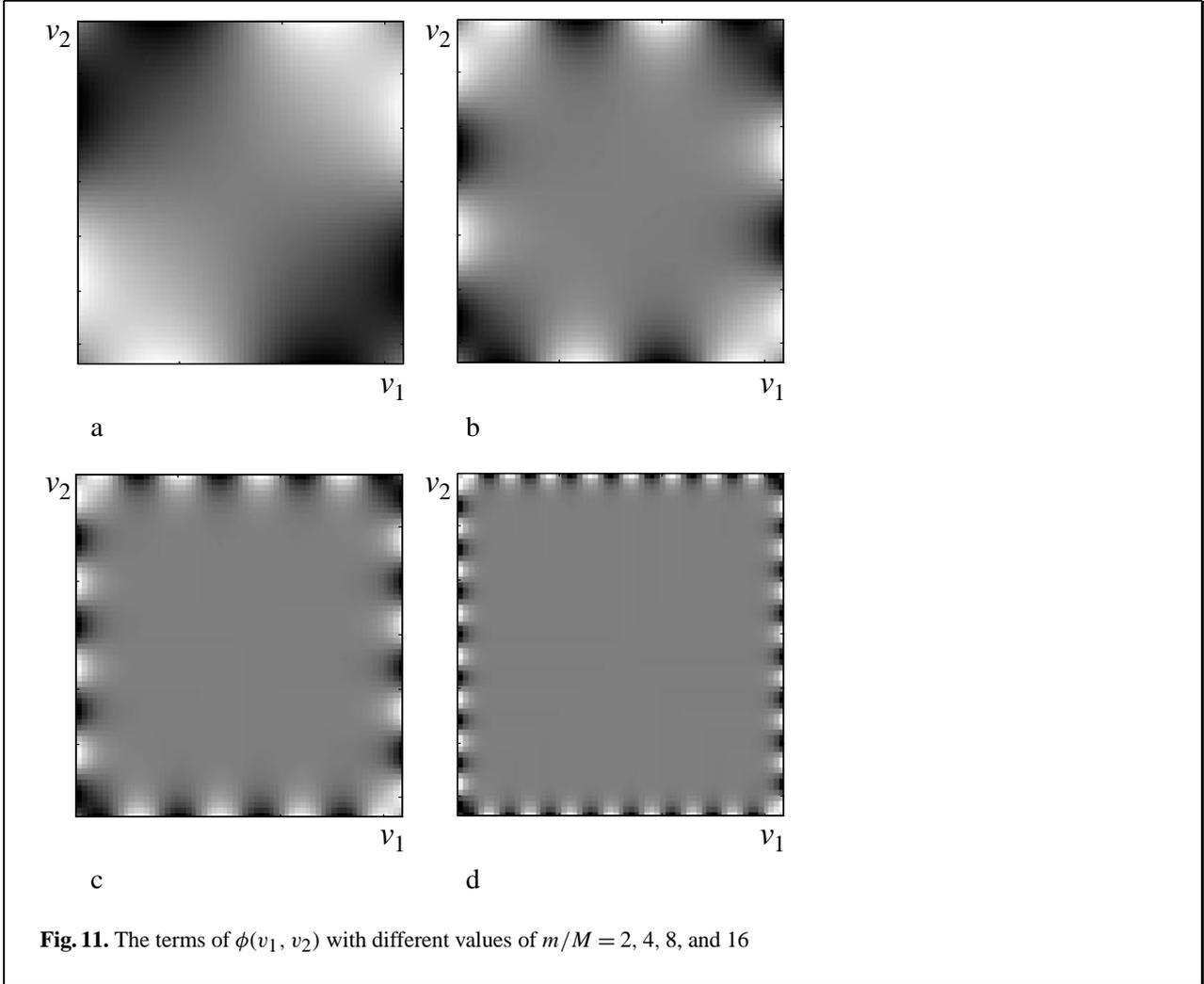
10c



10d

Fig. 9a,b. The reconstructed images for **a** the photograph with $M = 3$, and **b** the photograph with $M = 10$ by the conformal mappings

Fig. 10. a The term $\sin(m\pi v_1/M) \sinh(-m\pi v_2/M)$, **b** the term $\sin(m\pi v_1/M) \sinh(m\pi v_2/M)$, **c** the term $\sin(m\pi v_2/M) \sinh(-m\pi v_1/M)$, and **d** the term $\sin(m\pi v_2/M) \sinh(m\pi v_1/M)$ in (v_1, v_2) plane, where $m/M = 2$



Based on this representation, we applied the discussed encoding methods to compress the mapped data and reconstruct the image from the codes. Figure 7a and b display the sampled boundary data along the square regions' boundaries with $M = 3$ and $M = 10$. Figures 8 and 9 display the corresponding approximated images in square regions and decoded images, respectively, using the encoded samples shown in Fig. 7. To compress the data with a higher compression ratio, we first set $M = 3$ for this case, i.e., sampling 12 points along the boundary of each square region. The codes were sampled along the boundary of each square region. For a network with 20×20 neurons, the compression ratio was $256 \times 256 / 2721 = 24.09$. For finer compression with $M = 10$, the compression ratio was

$256 \times 256 / 8041 = 8.15$. The decompressed results obtained with these codes are shown in Fig. 8a and b. Figure 9a and b show the reconstructed images in the quadrilateral regions obtained by means of inverse conformal mappings. No significant seam distortion exists between any two connected quadrilateral regions in these reconstructed images.

To evaluate the quality of the reconstructed images, one type of the signal-to-noise ratio (SNR) was calculated as

$$\text{SNR} = 10 \log_{10} \frac{\sum_x \sum_y (f(x, y))^2}{\sum_x \sum_y (f(x, y) - \hat{f}(x, y))^2}, \quad (26)$$

where $f(x, y)$ is the original intensity value at the point (x, y) , and $\hat{f}(x, y)$ is the reconstructed inten-

sity value at the same point (x, y) . In Fig. 9a and b, the SNRs are 22.32 dB and 20.26 dB, respectively. From the simulation results, we observe that data near the edges of the images were lost by SON. Serious distortion has also appeared in the edge neurons. This phenomenon is called the boundary effect [16] in SON. In our simulations, we could easily add boundary regions along the edges to fix the lost portions. Links between boundary neurons and their closest points on the edges of the image constitute partitions for the boundary regions. We can also expand the distribution of image data beyond the edges to allow the network to cover the original image. The expanded data will be discarded from the reconstructed image.

This approach provides techniques for an adaptive partitioning of the image, conformal transformation for harmonic representation, and sampling in the mapped domain. This approach is particularly designed for compression of paintings and works of fine art on both planes and 3D surfaces. The computational cost of our approach is much higher than that of existing compression methods. Many other compression techniques can be combined with our techniques. A scalar quantizer can also be employed to further reduce the number of bits. Data in different regions may require different numbers of bits. In addition, the entropy coding techniques, for example, the Huffman coding, can be used to code the bit stream for higher compression ratios. Since this is a totally new technique, many interesting issues are currently being studied in the project.

Finally, we discuss the solution of the Laplace equation in a square region L and its boundary ∂L to observe the basic difference between our approach and the sinusoidal function based compression approach. To find the solution $\phi(v_1, v_2)$,

$$\frac{\partial^2 \phi}{\partial v_1^2} + \frac{\partial^2 \phi}{\partial v_2^2} = 0 \text{ in } L, \\ L = \{(v_1, v_2) | 0 \leq v_1 \leq M, \quad 0 \leq v_2 \leq M\}, \quad (27)$$

the boundary conditions on ∂L ,

$$\begin{aligned} \phi(v_1, 0) &= b_1(v_1), \\ \phi(v_1, M) &= b_2(v_1), \\ \phi(0, v_2) &= b_3(v_2), \quad \text{and} \\ \phi(M, v_2) &= b_4(v_2), \end{aligned} \quad (28)$$

are considered. The functions b_1, b_2, b_3 , and b_4 denote the known boundary functions along the bound-

ary ∂L . $4M$ points were equally spaced along ∂L in our simulations.

After applying superposition and separation of variables [18], the solution of $\phi(v_1, v_2)$ has the form:

$$\begin{aligned} \phi(v_1, v_2) = & \sum_m \sin\left(\frac{m\pi v_1}{M}\right) \left(A_m \sinh\left(\frac{m\pi(M-v_2)}{M}\right) \right. \\ & \left. + B_m \sinh\left(\frac{m\pi v_2}{M}\right) \right) \\ & + \sum_n \sin\left(\frac{n\pi v_2}{M}\right) \left(C_n \sinh\left(\frac{n\pi(M-v_1)}{M}\right) \right. \\ & \left. + D_n \sinh\left(\frac{n\pi v_1}{M}\right) \right), \quad (29) \end{aligned}$$

where

$$\begin{aligned} A_m &= a_m \operatorname{csch}(m\pi), \\ B_m &= b_m \operatorname{csch}(m\pi), \\ C_n &= c_n \operatorname{csch}(n\pi), \\ D_n &= d_n \operatorname{csch}(n\pi), \end{aligned} \quad (30)$$

and a_m, b_m, c_n , and d_n are the expanded coefficients in the Fourier sine series for b_1, b_2, b_3 , and b_4 , respectively, i.e.,

$$\begin{aligned} b_1(v) &= \sum_m a_m \sin\left(\frac{m\pi v}{M}\right), \\ b_2(v) &= \sum_m b_m \sin\left(\frac{m\pi v}{M}\right), \\ b_3(v) &= \sum_n c_n \sin\left(\frac{n\pi v}{M}\right), \quad \text{and} \\ b_4(v) &= \sum_n d_n \sin\left(\frac{n\pi v}{M}\right). \end{aligned} \quad (31)$$

Figure 10a–d shows the four terms that comprise the expanded expression (29) for $\phi(v_1, v_2)$. Figure 11a–d shows four examples for the terms of $\phi(v_1, v_2)$ in (29) with different values of m/M .

Acknowledgements. This work has been supported by National Science Council under projects NSC 82-0408-E-002-255 and NSC 84-2213-E-002-012.

References

1. Dony RD, Haykin, S (1995) Neural network approaches to image compression. *Proceedings of the IEEE* 83:288–303
2. Mougeot M, Azencott R, Angeniol B (1991) Image compression with back-propagation: improvement of visual restoration using different cost functions. *Neural Networks* 4:467–476
3. Jamison T, Schalkoff R (1988) Image labeling: a neural network approach. *Image Vision Comput* 6:203–214
4. Li SZ (1993) Self-organization of surface shapes. *Proceedings of the International Joint Conference on Neural Networks*, Nagoya, Japan, 2:1173–1176
5. Kohonen T (1995) *Self-organizing maps*. Springer, Berlin
6. Liou CY, Tai WP (2000) Conformality in the self-organization network, *Artificial Intelligence Journal*, to appear
7. Ivanov VI, Trubetskov MK (1995) *Handbook of conformal mapping with computer-aided visualization*. CRC Press, Boca Raton, Fla
8. Schwartz EL (1985) On the mathematical structure of the retinotopic mapping of primate striate cortex. *Science* 227:1066
9. Frederick C, Schwartz EL (1990) Conformal image warping. *IEEE Comput Graph Appl* 10:54–61
10. Landau P, Schwartz EL (1992) Computer simulation of cortical polymaps: a proto-column algorithm. *Neural Networks* 5:187–206
11. Henrici P (1974) *Applied and computational complex analysis*. John Wiley, New York
12. Trefethen LN (1980) Numerical computation of the Schwarz-Christoffel transformation. *SIAM J Sci Stat Comput* 1:82–102
13. Davis PJ, Rabinowitz P (1984) *Methods of numerical integration*. Academic Press, Orlando
14. Tuma JJ (1989) *Handbook of numerical calculations in engineering*. McGraw-Hill, New York
15. Newman JN (1980) *Marine hydrodynamics*. MIT Press, Cambridge
16. Kohonen T (1982) Self-organizing formation of topologically correct feature maps. *Biol cybern* 43:59–69
17. Thompson D'AW (1952) *On growth and form*. Cambridge University Press, Cambridge, second edition. p 1048
18. Folland GB (1992) *Fourier analysis and its applications*. Wadsworth & Brooks/Cole, Cole, Calif



WEN-PIN TAI was born in Taiwan on February, 3, 1968. He received the B.S. degree, the M.S. degree, and the Ph.D. degree in Computer Science and Information Engineering from National Taiwan University in 1990, 1992 and 1997, respectively. Currently he is the Assistant Professor in the Department of Computer Science at Chinese Culture University. His research interests include neural networks, digital signal processing, and numerical analysis.



CHENG-YUAN LIOU was born in Taiwan on November 14, 1951. He received the B.S. degree in Physics from National Central University in 1974, the M.S. degree in Physical Oceanography from National Taiwan University in 1978, and the Ph.D. degree in Ocean Engineering from the Massachusetts Institute of Technology in 1985. From 1986 to 1987, he was Visiting Associate Professor in the Institute of Applied Mechanics, National Taiwan University, where he taught courses in stochastic processes and digital signal processing and did research in acoustic signal processing. In 1988, he joined the faculty of the same university, where he is currently Professor in the Department of Computer Science and Information Engineering. His research subjects center on neural networks, music perception. Recently, he began a project involving synthetic singing and structured representation of arts.