VLSI Neuroprocessors for Video Motion Detection

Ji-Chien Lee, *Member, IEEE*, Bing J. Sheu, *Senior Member, IEEE*,
Wai-Chi Fang, *Member, IEEE*, and Rama Chellappa, *Fellow, IEEE*

***Abstract*—The system design of a locally connected competitive neural network for video motion detection is presented. The motion information from a sequence of image data can be determined through a two-dimensional multiprocessor array in which each processing element consists of an analog neuroprocessor. Massively parallel neurocomputing is done by compact and efficient neuroprocessors. Local data transfer between the neuroprocessors is performed by using an analog point-to-point interconnection scheme. To maintain strong signal strength over the whole system, global data communication between the host computer and neuroprocessors is carried out in a digital common bus. A mixed-signal very large scale integration (VLSI) neural chip that includes multiple neuroprocessors for fast video motion detection has been developed. Measured results of the programmable synapse, and winner-take-all circuitry are presented. Based on the measurement data, system-level analysis on a sequence of real-world images was conducted. A 1.5 × 2.8-cm^2 chip in a 1.2-μm CMOS technology can accommodate 64 velocity-selective neuroprocessors. Each chip can achieve 83.2 giga connections per second. The intrinsic speed-up factor over a Sun-4/75 workstation is around 180.**

I. Introduction

RAPID advances in a very large scale integration (VLSI) technologies have made possible the integration of multiple-million transistors on a single chip. In 1992, the feature size for computer memory technologies is around 0.4 μm [1]. The use of VLSI circuits can greatly reduce the physical size and enhance the performance and reliability of microelectronic systems. In the microprocessor domain, continuous progress on reduced instruction set computers (RISC) enables the introduction of the Intel-i860 chip [2], the SPARC chip from Sun Microsystems, Inc. [3], and the 400-MIPS Alpha chip from Digital Equipment Corporation [4]. In the digital signal processing domain, the TMS-320C40 chip from Texas Instruments, Inc. [5] includes six communication ports to facilitate various data communication schemes. In the dedicated neural computing domain, the 11-million-transistor CNAPS chip from Adaptive Solutions, Inc. and Inova Microelectronics, Inc. [6] includes 64 digital processors for general-purpose neural network execution.

A desirable configuration for an integrated information processing system is shown in Fig. 1. A powerful multimedia data-fusion machine is equipped with several smart interface units to communicate with the analog signals in the real world. These analog signals contain information with some degree of fuzziness [7]. The image acquisition/understanding and advanced display units are used to process visual information. The speech recognizer and synthesizer units are used to process audio information [8]. The microsensor [9] and controller units are used to perform physical actions. Such an intelligent system can be used in offices, factories, and autonomous vehicles. VLSI neural chips can be effectively used in the construction of the interface units.

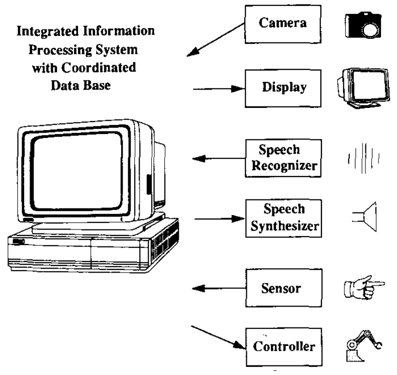

Integrated Information Processing System with Coordinated Data Base
Camera
Display
Speech Recognizer
Speech Synthesizer
Sensor
Controller

Fig. 1. An integrated information system. Information can be transferred between the system and the real world in multimedia format.

Various design approaches are applicable for the construction of neural chips. The analog circuit approach is quite attractive in terms of hardware size, power consumption, and speed [10]. Analog neural networks were used as sensory devices to preprocess the real-world data, as reported by Mead *et al.* [11], [12] and Abidi *et al.* [13]. In addition, many other analog VLSI neural chips have been reported [14]–[16]. One unique feature of a purely analog neural network is the limited computational precision for complex problems. To enhance the performance of analog VLSI neurocomputing, extra analog switching circuits can be used to facilitate the reconfigurability and scalability of analog neural networks [17]. The digital circuit approach offers greater flexibility, scalability, and accuracy than the analog circuit approach. By using logic and memory, a large problem can be partitioned and processed by the digital neural networks. Some general-purpose digital VLSI neural chips were reported [18]–[20].

In this paper, a mixed-signal design approach is used to exploit the massively parallel computational power of the neural network architecture for video motion detection. To solve low-level vision processing problems, multiple neurons and synapses can be clustered together to function as one pixel processing element. By using compact analog circuit design for the neuron and synapse cells, highly parallel computation on the pixel level can be achieved.

Manuscript received July 5, 1991; revised April 23, 1992.

The authors are with the Department of Electrical Engineering (Electrophysics and Systems), Signal and Image Processing Institute, University of Southern California, Los Angeles, CA 90089–0271.

IEEE Log Number 9201749.

II. System Architecture

Motion information extracted from a sequence of time-varying images plays a key role in the image understanding and automated control processes. The requirement of an enormous amount of computational power for analyzing image sequences is always a major barrier to real-world applications of most vision-processing algorithms. By using multiprocessor-based VLSI design, the parallelism embedded in low-level vision processes can be fully explored. The single instruction multiple data (SIMD) architecture is a good example [21], [22]. Two specific multiprocessor-based neural engines based on the SIMD architecture have been reported. The CNAPS machine, from Adaptive Solutions, Inc. [18], consists of an array of processing nodes (PN's). Each PN is an arithmetic processor with its own local memory. The array is sequenced by a system controller. Thus every PN executes the same instruction at a given clock period. The input data and control commands are broadcast to all PN's through the common bus. The output data of the PN's are transmitted through the data bus by the time-multiplexing scheme. A local digital data link exists between adjacent PN's to allow quick data transfer. Due to the simplicity of the broadcast scheme, no complex routing networks are required. The systolic/cellular array processors (SCAP) system, from Hughes Research Lab. in Malibu, CA [23], consists of a 16×16 processor array, a dual-port array memory, and a system controller. The mesh-connection architecture is used. The boundary columns are connected via the wrap-around scheme, and the top and bottom rows are connected to the two ports of the system memory. Data communication can be conducted in the paralleled format or in the pipelined format.

In our design, a mesh-connected two-dimensional neuroprocessor array is used for high-speed video motion detection. Each processoring element can extract the velocity information for one pixel. Interprocessor communication is done by dedicated analog point-to-point interconnections. Data communication between the host computer and array processors is carried out through the digital common bus to preserve signal strength and to achieve simple network scalability. By using this efficient communication among processors, a high computational power per unit silicon area can be achieved.

III. Motion Detection Algorithm

Many features from the images such as points, lines, curves, and optical flow, can be used to estimate motion parameters. Optical flow is the apparent motion of the brightness patterns. Generally, the optical flow corresponds to the motion field [24], and provides important information about the spatial arrangement of the objects, the rate of change of this arrangement in a given scene, and also the perceiver's own movements. Optical flow can thus be used for deriving relative depth of points [25], [26], segmenting images into regions [27], and estimating the object motion in the scene [28].

According to the nature of the measured primitives, existing approaches to optical flow computing can be divided into two types: the image intensity based approach and the token based approach. The intensity based approach relies on the assumption that changes in intensity are strictly due to the motion of the object and uses the image intensity values and their spatial and temporal derivatives to compute the optical flow. By expanding the intensity function into a first-order Taylor series, Horn and Schunck [29] derived an optical flow equation using the brightness constancy assumption and spatial smoothness constraints. An iterative method for solving the resulting equation was also developed. The token based approach is to consider the motion of tokens such as edges, corners, and linear features in an image. The key advantage of the token based approach is that tokens are less sensitive to variations of the image intensity. The token based approach provides the information of the object motion and shape at edges, corners, and linear features. An interpolation procedure has to be included when dense data are required.

Recently, several researchers used neural networks to conduct optical flow computing [30], [31]. To prevent the smoothness constraint from taking effect across strong velocity gradients, a line process has been incorporated into the optical flow equation [31]. The resulting equation is nonconvex and includes the cubic and some higher terms. Instead of using an annealing algorithm which is very time consuming, a deterministic algorithm was used to obtain a near-optimal solution. Convergence of such a network was obtained within a few iteration cycles. Basically, the mixed analog/digital neural network approach is to first use Horn's optical flow equation to find a smoothest solution and then to update the line process by lowering the energy function of the network repeatedly. In the hardware implementation, the resistive network is quite susceptible to device variation effects from the silicon CMOS fabrication processes.

In order to obtain a dense flow field, the intensity based approach is preferable. However, the intensity value may be corrupted by noise appeared in natural images and partial deviatives of the intensity value are sensitive to rotation. It is difficult to detect the rotational objects in natural images based on such measurement primitives. Under the assumption that changes in intensity are strictly due to the motion of the object, Zhou *et al.* [32], [33] use the principal curvatures of the intensity function to compute the optical flow because they are rotation-invariant. The intensity values and their principal curvatures are estimated by using a polynomial fitting technique. Under the assumption of local rigid motion and the smoothness constraint, a self-organizing neural network [34]–[36] was developed to compute the optical flow. A deterministic decision rule was used for the updating of neuron states.

Let the velocity field consist of two components k and l. A set of $(2D_k+1)(2D_l+1)$ modules of neurons are used to represent the optical flow field, where D_k and D_l are the maximum values of velocity components in k and l directions, respectively. For the implementation purposes, the velocity component range is sampled using bins of size Q. As shown in Fig. 2, each module corresponds to a velocity value and contains $N_r \times N_c$ neurons if the images are of size $N_r \times N_c$. All neurons in the same module are self-connected and locally interconnected with other neurons in a neighborhood of size $\Gamma \times \Gamma$. Every pixel is represented by $(2D_k+1)(2D_l+1)$

mutually exclusive neurons which form a hypercolumn for velocity selection. When the neuron at the point (i, j) in the (k, l)th module is one, the actual velocities in the k and l directions at the point (i, j) are kQ and lQ, respectively.

Let $V = \{v_{i,j,k,l}, 1 \leq i \leq N_r, 1 \leq j \leq N_c, -D_k \leq k \leq D_k, -D_l \leq l \leq D_l\}$ be a binary set of the neural network with $v_{i,j,k,l}$ denoting the state of the (i, j, k, l)th neuron which is located at point (i, j) in the (k, l)th module, $T_{i,j,k,l;m,n,k,l}$ be the synaptic interconnection strength from neuron (i, j, k, l) to neuron (m, n, k, l), and $I_{i,j,k,l}$ be the bias input.

At each step, the neuron (i, j, k, l) synchronously receives signals from itself, neighboring neurons, and a bias input

$$u_{i,j,k,l} = \sum_{(m-i,n-j)\in S_0} T_{i,j,k,l;m,n,k,l} v_{m,n,k,l} + I_{i,j,k,l} \tag{1}$$

where S_0 is an index set for all neighbors in a $\Gamma \times \Gamma$ window centered at point (i, j). The $u_{i,j,k,l}$ is then processed by the winner-take-all circuitry to determine the velocity of the pixel

$$v_{i,j,k,l} = g(u_{i,j,k,l}) \tag{2}$$

where $g(x_{i,j,k,l})$ is the winner-take-all function:

$$g(x_{i,j,k,l}) = \begin{cases} 1 & \text{if } x_{i,j,k,l} = \max(x_{i,j,p,q}; -D_k \leq p \leq D_k, \\ & \quad -D_l \leq q \leq D_l). \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The network operation will be terminated if the network converges; i.e., the energy function of the network defined by

$$E = -\frac{1}{2} \sum_{i=1}^{N_r} \sum_{j=1}^{N_c} \sum_{k=-D_k}^{D_k} \sum_{l=-D_l}^{D_l} \cdot \left(\sum_{(m-i,n-j)\in S_0} T_{i,j,k,l;m,n,k,l} v_{i,j,k,l} v_{m,n,k,l} + I_{i,j,k,l} v_{i,j,k,l} \right), \tag{4}$$

reaches a minimum.

Two important features of the network should be noted:

i) The synaptic interconnection strength between neurons on different modules are zeros because only the neurons in the same module are connected, i.e.

$$T_{i,j,k,l;m,n,p,q} = 0, \quad \text{for } (k, l) \neq (p, q), \quad \text{if } (i, j) \neq (m, n). \tag{5}$$

ii) A maximum evolution function is used to ensure that only one neuron which has the maximum excitation is fired and the other $(2D_k + 1)(2D_l + 1) - 1$ neurons are turned off.

As reported in [32], a smoothness constraint is used for obtaining a smooth optical flow field and a line process is employed for detecting motion discontinuities. The line process consists of vertical and horizontal lines, L^v and L^h, respectively. Each line can be in either one of the two states: 1 for being active and 0 for being idle. The error function for computing the optical flow from a pair of image frames can be expressed as

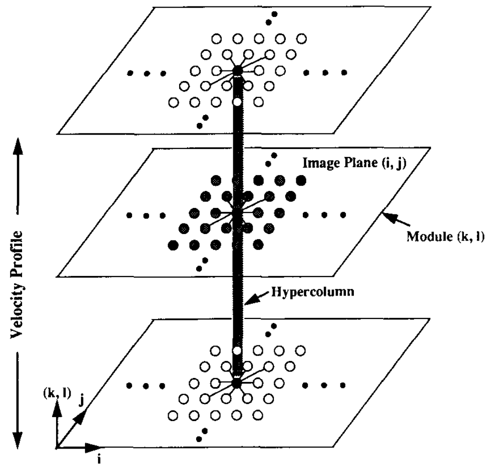


Fig. 2. A competitive neural network is used for optical flow computing. The neurons are arranged in modules according to their velocity selectivity. Indexes i and j denote the image coordinates. (k, l) denotes the velocity coordinate. Each hypercolumn represents the velocity selectivity of one image pixel.

$$\begin{aligned} E = \sum_{i=1}^{N_r} \sum_{j=1}^{N_c} \sum_{k=-D_k}^{D_k} \sum_{l=-D_l}^{D_l} \Bigg[\Big\{ & A[k_{11}(i, j) - k_{21}(i + k, j + l)]^2 \\ & + A[k_{12}(i, j) - k_{22}(i + k, j + l)]^2 \\ & + [g_1(i, j) - g_2(i + k, j + l)]^2 \Big\} v_{i,j,k,l} \\ & + \Bigg\{ \frac{B}{2} \sum_{s\in S} \left(v_{i,j,k,l} - v_{(i,j)+s,k,l}\right)^2 \\ & + \frac{C}{2} \Big[(v_{i,j,k,l} - v_{i+1,j,k,l})^2 (1 - L^h_{i,j,k,l}) \\ & + (v_{i,j,k,l} - v_{i,j+1,k,l})^2 (1 - L^v_{i,j,k,l}) \Big] \Bigg\} \Bigg] \end{aligned} \tag{6}$$

where $k_{11}(i, j)$ and $k_{12}(i + k, j + l)$ are the principal curvatures of the first image, $k_{21}(i, j)$ and $k_{22}(i + k, j + l)$ are the principal curvatures of the second image, $g_1(i, j)$ and $g_2(i + k, j + l)$ are the intensity values of the first and second images, respectively. Here, $S = S_0 - (0, 0)$ is an index set excluding $(0, 0)$. A, B, and C are empirical constants.

The principal curvatures are defined as [37]

$$k_1(i, j) = M + \left(M^2 - G\right)^{1/2} \tag{7}$$

and

$$k_2(i, j) = M - \left(M^2 - G\right)^{1/2} \tag{8}$$

where $k_1(i, j)$ and $k_2(i, j)$ are the principal curvatures, G and M are the Gaussian and mean curvatures given by

$$G = \frac{\partial^2 g(i, j)}{\partial i^2} \cdot \frac{\partial^2 g(i, j)}{\partial j^2} - \left[\frac{\partial^2 g(i, j)}{\partial i \partial j}\right]^2 \tag{9}$$

and

$$M = \frac{1}{2}\left[\frac{\partial^2 g(i,j)}{\partial j^2} + \frac{\partial^2 g(i,j)}{\partial j^2}\right]. \tag{10}$$

A polynomial fitting technique can be used to estimate the derivatives. The k_{11}, k_{21}, k_{12}, and k_{22} values are calculated from the images by the host computer and sent to the neuroprocessor for network evaluation.

In (6), the first term is to find velocity values such that all points of two images are matched as closely as possible in a least-squares sense. The second term, which is weighted by B, is the smoothness constraint on the solution and the third term, which is weighted by C, is a line process to weaken the smoothness constraint and to detect motion discontinuities. The constant A in the first term determines the relative importance of the intensity values and their principal curvatures to achieve the best results. The line process weakens the smoothness constraints by changing the smoothing weights, resulting in space-variant smoothing weights. For example, if all lines are on, the weights will be $B/2$. If all lines are off, the weights at the four nearest neighbors of the center point are increased by $C/2$.

By choosing the interconnection strengths and bias inputs as

$$\begin{aligned} T_{i,j,k,l;m,n,k,l} = -[&48B + C(4 - L^h_{i,j,k,l} - L^h_{i,j+(-1),k,l} \\ &- L^v_{i,j,k,l} - L^v_{i+(-1),j,k,l})]\delta_{i,m}\,\delta_{j,n} \\ &+ C[(1 - L^h_{i,j,k,l})\delta_{i,m}\,\delta_{j+1,n} \\ &+ \left(1 - L^h_{i,j+(-1),k,l}\right)\delta_{i,m}\,\delta_{j+(-1),n} \\ &+ (1 - L^v_{i,j,k,l})\delta_{i+1,m}\,\delta_{j,n} \\ &+ (1 - L^v_{i+(-1),j,k,l})\delta_{i+(-1),m}\,\delta_{j,n}] \\ &+ 2B\sum_{s\in S}\delta_{(i,j),(m,n)+s} \end{aligned} \tag{11}$$

and

$$\begin{aligned} I_{i,j,k,l} = -A\{&[k_{11}(i,j) - k_{21}(i+k,j+l)]^2 \\ &+ [k_{12}(i,j) - k_{22}(i+k,j+l)]^2\} \\ &- [g_1(i,j) - g_2(i+k,j+l)]^2 \end{aligned} \tag{12}$$

where $\delta_{a,b}$ is the Dirac delta function, the error function in (6) is mapped into the energy function of the neural network in (4). Notice that the interconnection strengths consist of constants and line process only. The bias inputs contain all the information from images. When the network reaches a stable condition, the optical flow field is determined by the neuron states. The size of a typical smoothing window is 5 × 5.

Since the first and second terms in (6) do not contain the line process, the updating of the line process is prior to the updating of neuron states. Let $L^{v;\text{new}}_{i,j,k,l}$ and $L^{v;\text{old}}_{i,j,k,l}$ denote the new and old states of the vertical line $L^v_{i,j,k,l}$, respectively. Let $\Psi_{i,j,k,l}$ be the potential of the vertical line $L^v_{i,j,k,l}$ given by

$$\Psi_{i,j,k,l} = \frac{C}{2}\left(v_{i,j,k,l} - v_{i+1,j,k,l}\right)^2. \tag{13}$$

Then, the new state is determined by

$$L^{v;\text{new}}_{i,j,k,l} = \begin{cases} 1 & \text{if } \Psi_{i,j,k,l} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

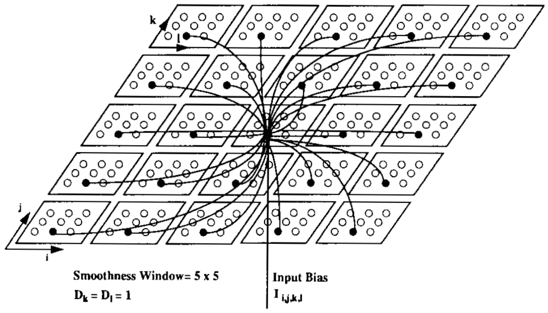


Fig. 3. An artificial neural network. Each small frame denotes a hypercolumn. The neurons in a hypercolumn are uniformly distributed on a plane. In addition to an external bias input, each neuron has a self-feedback, and receives inputs from similar directionally selective neurons at the neighboring hypercolumns.

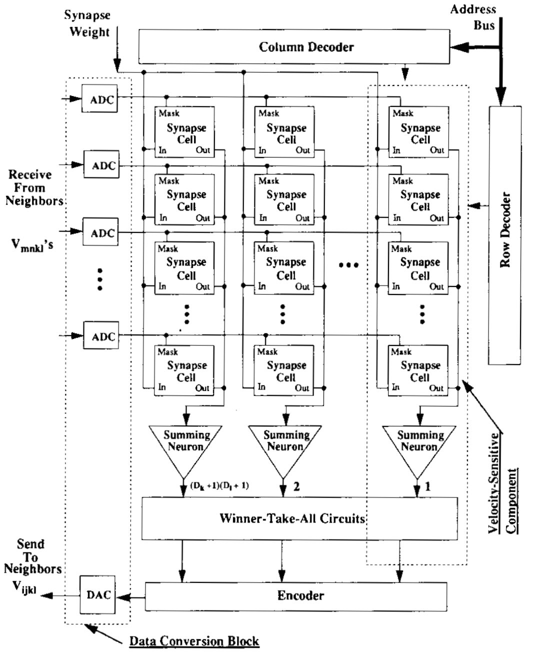
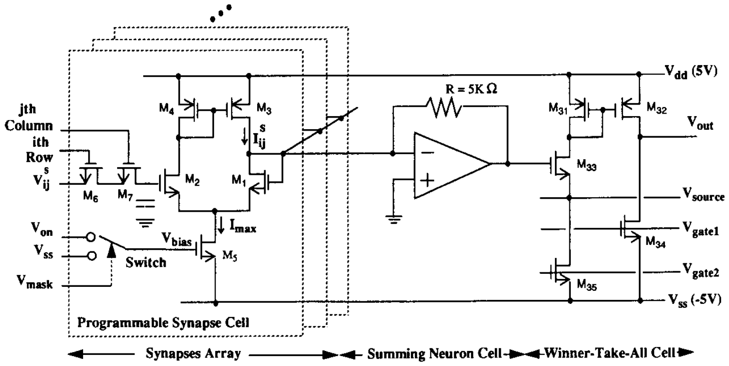


Fig. 4. Functional diagram of one neuroprocessor. It contains a velocity-sensitive component array and data conversion block. The velocity of one image pixel can be determined by one neuroprocessor.

Whenever the states of neurons $v_{i,j,k,l}$ and $v_{i+1,j,k,l}$ are different, the vertical line $L_{i,j,k,l}$ will be active provided that the parameter C is greater than zero. If $C = 0$, then all lines are inactive, which means that no line process exists in the network operation. The choice of C is closely related to selecting the smoothness parameter B in (6). A similar updating scheme is also used for the horizontal lines. In the prototype neural chip design, computation for the terms which are weighted by the parameter C is not included.

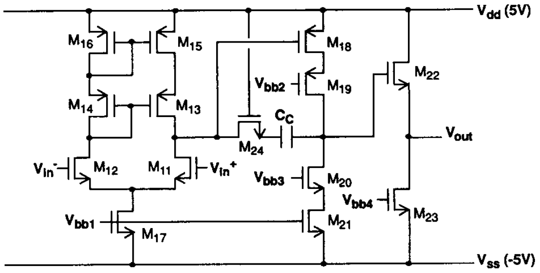
The state of each neuron is synchronously evaluated and updated according to (1) and (2). The initial states of the neurons are set as

$$v_{i,j,k,l} = \begin{cases} 1 & \text{if } I_{i,j,k,l} = \max(I_{i,j,p,q;} - D_k \le p \le D_k, \\ & \qquad -D_l \le q \le D_l) \\ 0 & \text{otherwise} \end{cases} \tag{15}$$



Transistor	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_{31}	M_{32}	M_{33}	M_{34}	M_{35}
Size in um	3/39	3/39	4/4	4/4	5/3	4/2	4/2	8/4	8/4	8/2	8/2	20/8

Fig. 5. Circuit schematic of the velocity-sensitive component. It contains a programmable synapse array, a summing neuron, and a winner-take-all cell.



Transistor	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}	M_{17}	M_{18}
Size in um	16/2	16/2	6/2	6/2	20/4	20/4	40/4	46/4
Transistor	M_{19}	M_{20}	M_{21}	M_{22}	M_{23}	M_{24}	C_C	
Size in um	90/2	26/2	52/4	600/2	400/4	50/2	1.5pF	

Fig. 6. Circuit schematic of the operational amplifier used for summing neuron.

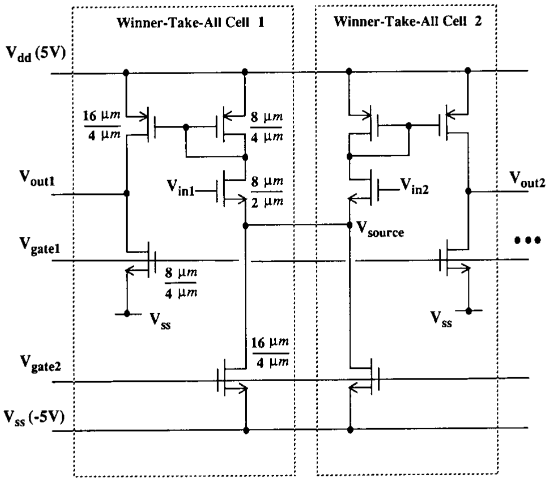


Fig. 7. Two winner-take-all cells are connected as a differential operational amplifier.

where $I_{i,j,k,l}$ is the bias input. The initial conditions are completely determined by the bias inputs. If there are two maximal bias inputs at point (i, j), then only the neuron corresponding to the smaller velocity is initially set to 1 and the other one is set to 0. This is consistent with the minimal mapping theory [38]. In the updating scheme, the minimal mapping theory is also used to handle the case of two neurons having the same largest inputs.

IV. The Neural-Based Neuroprocessor Design

A. VLSI Architecture

To implement the electronic neural network processor, a VLSI architecture has been developed which maps the three-dimensional neural network configuration onto a two-dimensional plane. As shown in Fig. 3, each small frame represents one velocity-selective hypercolumn which contains $(2D_k + 1)(2D_l + l)$ velocity-sensitive components. Each hypercolumn is locally interconnected with the $\Gamma \times \Gamma - 1$ neighboring hypercolumns. The hypercolumn is designed as a neuroprocessor within which the velocity selectivity of an image pixel can be conducted. Mixed analog/digital design technologies are utilized for the neuroprocessor design to achieve compact and programmable synapses and neurons for massively paralleled neural computation [39].

To simplify the two-dimensional interconnection design for computation of optical flow, the analog point-to-point interconnection for local communication and the digital common bus for global communication are used. Since velocity information of one pixel is affected by its neighbors, each neuroprocessor receives information from the neighboring neuroprocessors during the network operation. Data communication between these locally interconnected neuroprocessors is one key factor on the overall system performance. There are three different methods to accomplish the local data communication with trade-offs on the operation speed and silicon area.

The first method is to use the digital bit-parallel point-to-point interconnection. The $(2D_k+1)(2D_l+1)$-bit $v_{i,j,p,q}$'s, where $-D_k \leq p \leq D_k, -D_l \leq q \leq D_l$, are transmitted using the word-wide point-to-point interconnections. The data transfer speed is very fast. However, the total number of interconnection lines for each neuroprocessor is as large as $(2D_k+1)(2D_l+1)(\Gamma \times \Gamma)$. The silicon area for the interconnection routing is large. The required large pin count becomes a major constraint for hardware implementation.

The second method is to use the digital bit-serial point-to-point interconnection. The $v_{i,j,p,q}$'s are sent in a bit-serial order by using a time-multiplexing technique. The total number of interconnection lines is reduced by a factor of $(2D_k+1)(2D_l+1)$. However, the time required for data transfer increases with the same factor [40]. In addition, the required hardware overhead for time-multiplexing includes a one-bit latch for each synapse cell, the multiplexing control signals, and the associated decoding circuitry.

The third method is to use the analog bit-parallel point-to-point interconnection. The $v_{i,j,p,q}$'s are converted to an analog value, and then sent to the neighboring neuroprocessors. The $v_{i,j,p,q}$'s are converted back into digital values at the receiving sites. The required hardware overhead includes the digital/analog converter or analog/digital converter at the two ends of interconnection wire. Both the analog interconnection method and the multiplexing digital interconnection method are suitable in the neural network processor design. For applications with large D_k and D_l values, the multiplexing digital interconnection method is preferred.

A functional diagram of the velocity-selective neuroprocessor is shown in Fig. 4. It includes a velocity-sensitive component array, and a data conversion block. The array has $(2D_k+1)(2D_l+1)$ velocity-sensitive components which are laterally connected through the winner-take-all circuit. The velocity of the neuroprocessor is determined by competition which is performed by the winner-take-all circuit. Only one velocity component which has the maximum excitation will be the winner to represent the velocity of that pixel. The data conversion block is used for the analog point-to-point interprocessor interconnection.

As shown in Fig. 5, the velocity-sensitive component is constructed with one synapse array, one summing neuron, and one winner-take-all cell. The synapse array contains $\Gamma \times \Gamma + 1$ programmable synapses. The synapse weights $T_{i,j,k,l;m,n,k,l}$ are stored as charge packets on capacitors and must be refreshed periodically [17], [41]. The binary outputs $v_{m,n,p,q}$ from the neighboring neuroprocessors are routed to the corresponding mask ports of the synapse cells to conduct the network operation. A summing neuron functions as a parallel current-mode adder. Each summing neuron with its associated programmable synapse array perform a complete inner-product computation. The binary outputs of the winner-take-all circuit represent the velocity status.

The synapse weights and bias inputs are calculated by the host computer or a digital coprocessor and stored in a digital static-RAM. The 8-bit digital/analog converter transforms the digital representation of the synapse weights into analog values for charging the weight-storage capacitances of the synapse matrix. A two-port static-RAM and differential amplifier-based synapse design allows network retrieving and learning processes to occur concurrently.

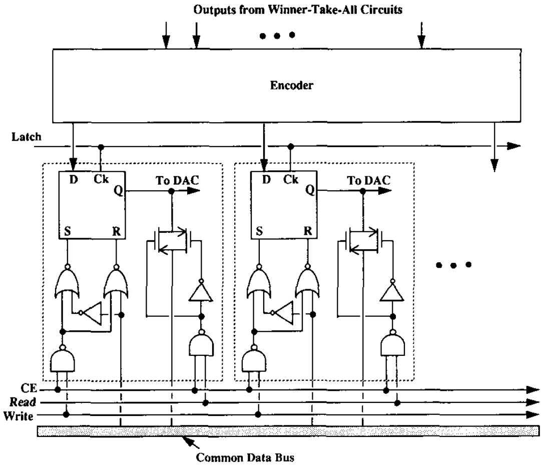


Fig. 8. The data latch is used to store the final results of the image pixels. The contents of latch can be read/write by host computer through the common data bus.

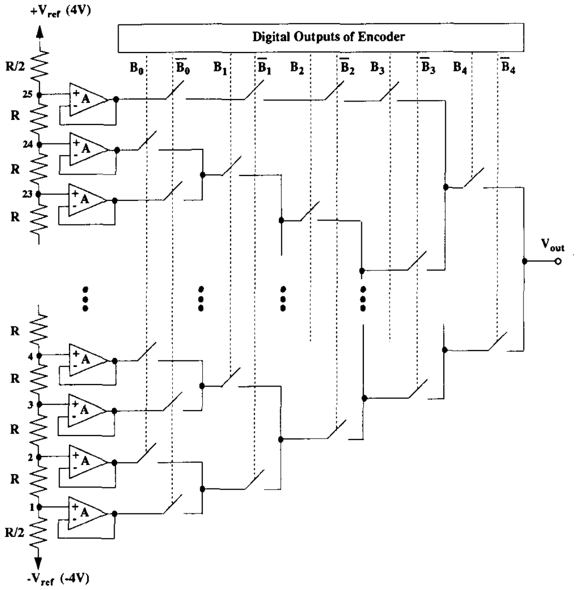


Fig. 9. A voltage-scaling digital-to-analog converter. The unity-gain followers are used to buffer the resistor string from loading.

B. Detailed Circuit Design

In Fig. 5, a transconductance amplifier consisting of transistors M_1-M_5 produces synapse output current $I^s_{i,j}$ according to mask voltage V_{mask} and weight voltage $V^s_{i,j}$. The bias voltage V_{bias} controls the dynamic range of synapse cells by adjusting the bias current in the transconductance amplifiers. When the V_{mask} is a logic 1, the V_{bias} is connected to V_{on}

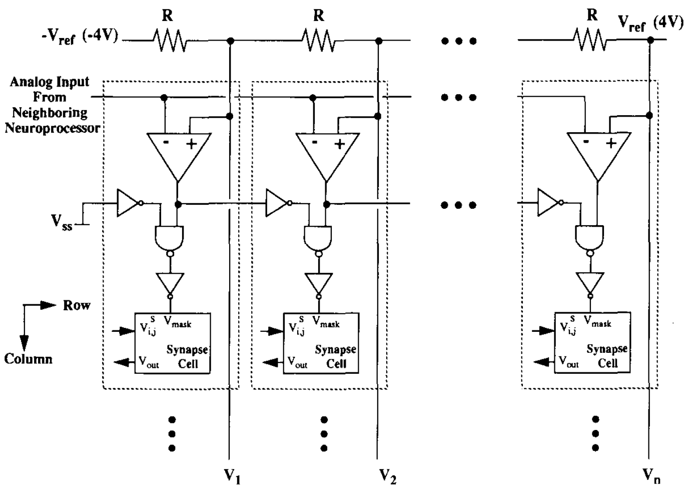


Fig. 10. A parallel and distributed analog-to-digital converter. One common reference voltage and one scaling resistor-chain are used. The comparators and associated digital decoding circuits are distributed to each synapse cell.

to provide the amplifier with a specific bias current $I^{\max}$. When the V_{mask} is at logic 0, the V_{bias} is connected to the negative power supply so that no synapse output current is produced. Therefore, the V_{mask} performs a masking operation on the synapse weight voltage $V^s_{i,j}$. The mask voltage of each synapse cell is directly related to the value of the $v_{m,n,k,l}$ which represents the velocity information of the neighboring pixels. The maximum synapse conductance is decided by device sizes of the differential pair and the bias current $I^{\max}$, while the minimum synapse conductance is determined by the resolution of the weight value on the MOS capacitance. The synapse output currents are summed up and converted to the voltage format at the summing neuron. This compact synapse circuit performs a two-quadrant multiplication. The polarity of the synapse output depends on the value of weight voltage $V^s_{i,j}$. An 8-bit resolution can be easily supported in the DRAM-style synapse cell. The required refreshing time is around 100 ms. Recently, detailed design of four quadrant Gilbert multiplier for synapse cells were reported [17], [42]. Multiple differential pairs and current-mirror circuits make the wide-range operation possible. If more than 8-bit resolution is required for the synapse function, large-geometry MOS transistors, and shorter refreshing time will be needed. In the EEPROM-style synapse cell [43], [44], at least a 6-bit resolution can be obtained.

The summing neuron functions as a current-to-voltage converter and is realized by using a two-stage operational amplifier and a feedback resistor. Circuit schematic diagram of the two-stage operational amplifier is shown in Fig. 6. Transistors M_{13} and M_{14} form an improved cascode stage to increase the voltage gain and M_{24} operates as a resistor for proper frequency compensation. The amplifier voltage gain of 100 dB can be achieved.

The outputs of the winner-take-all circuit are binary values. Only one winner cell with the maximum input voltage will have the logic-1 output value. The other cells will have the logic-0 output value. The winner-take-all circuitry functions as a multiple-input parallel comparator. Fig. 7 shows the circuit schematic diagram of two winner-take-all cells. The high-resolution and expandability of this winner-take-all circuit makes it suitable for many competitive learning neural networks [45]–[48]. Winner-take-all circuits with transistors biased in the subthreshold region were reported in the literature. Such low-power winner-take-all circuits [49], [50] have slower speed response and are suitable for special applications if the power consumption is crucial.

After the winner-take-all circuit, the $(2D_k + 1)(2D_l + 1)$ binary outputs represent the velocity information of one image pixel. Combinational logic gates are used to encode these $(2D_k + 1)(2D_l + 1)$ binary signals and to store the result into a data latch. Fig. 8 shows digital circuits of the data latch with the associated read/write control logic. The final velocity result is read by the host computer from the data latches through the digital common bus.

Fig. 9 shows a voltage-scaling digital-to-analog converter [51] which is used to convert the encoded binary code to the analog value and send it to the neighboring neuroprocessors. The voltage-scaling converter uses a series of resistors connected between V_{ref} and $-V_{\text{ref}}$ to provide intermediate voltage values. For an N-bit converter, the resistor string would have $2^N + 1$ resistor segments. In Fig. 9, a total of 26 resistor segments are used. The resistor is implemented in the P-well diffusion layer of the MOS fabrication process. The sheet resistance of the P-well layer is around 2 K$\Omega/\Box$ from the MOSIS 1.2-μm CMOS P-well process [52], [53]. Unity-gain followers are used to buffer the resistor string from conductive loading. Each tap is connected to a switching tree whose switches are controlled by the bits of the digital word. Each switch is implemented by a CMOS transmission gate.

When the analog velocity information from the neighboring neuroprocessors is received, a total of $\Gamma \times \Gamma - 1$ analog-to-digital converters are used to convert these analog values back to the binary values with $(2D_k + 1)(2D_l + 1)$ bits.

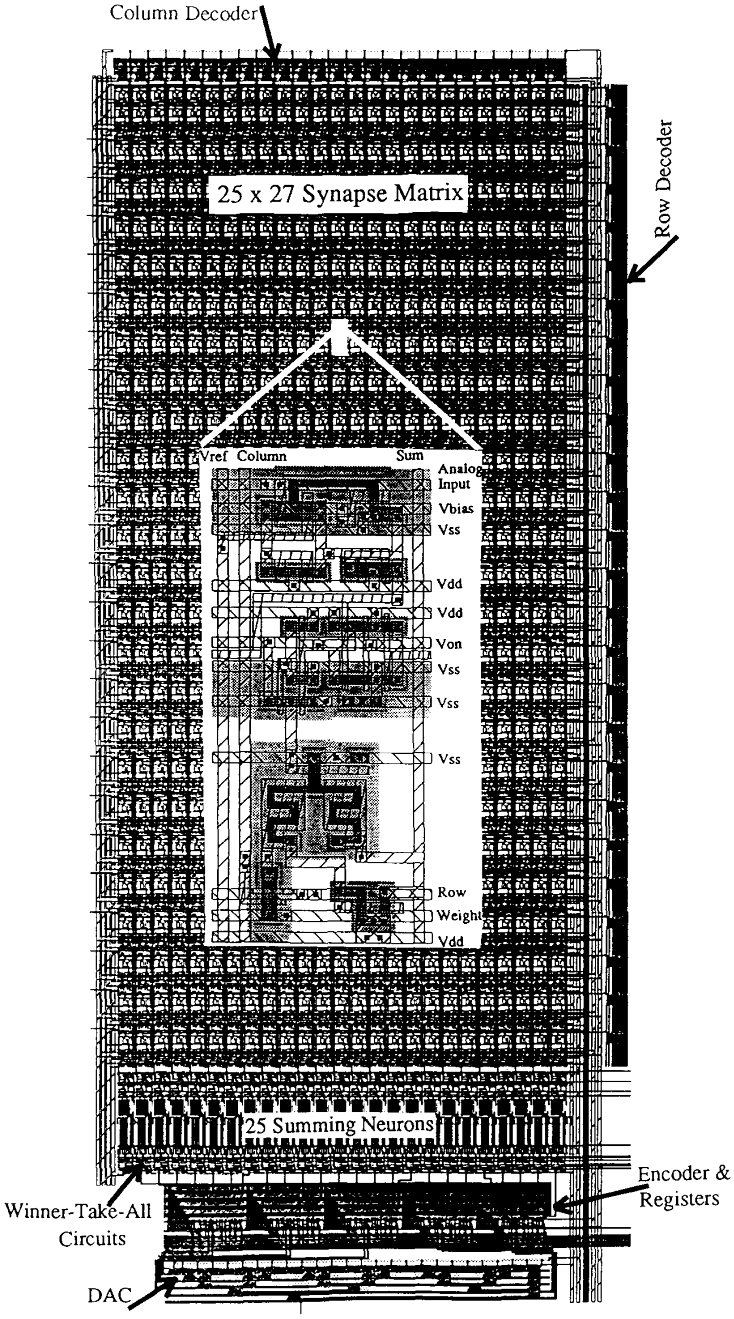


Fig. 11. The layout of one velocity-selective neuroprocessor. It contains 25 neurons, 25 × 27 synapse matrix and is able to detect the moving object with 25 different velocities. It occupies 2482 × $5636\lambda^2$ silicon area. The synapse cell is shown in the insert.

Only one of these bits is logic-1 and the others are logic-0. To achieve high-speed performance and a compact silicon area, a parallel and distributed analog-to-digital converter has been designed. One voltage scaling resistor-chain is used. As shown in Fig. 10, the comparators and the associated digital decoding circuitries are distributed into the synapse cells. The comparators included in the same velocity-sensitive component use the same reference voltage provided by the resistor-chain. The distributed decoding circuitries make sure that only one of $(2D_k + 1)(2D_l + 1)$ binary outputs is logic 1 and the others are inhibited to logic 0.

V. Experimental Results

In the prototype neuroprocessor chip design, $D_k = D_l = 2$ and a size of 5×5 smoothing window are used. The

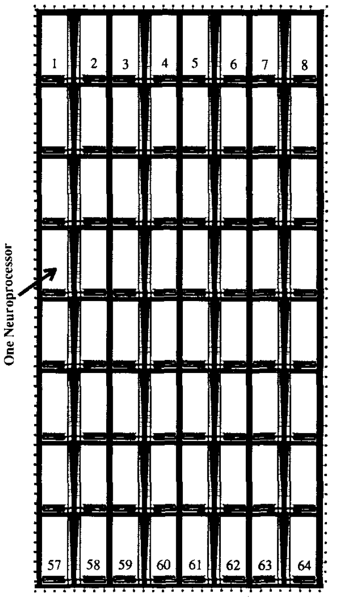


Fig. 12. The layout of VLSI neural chip. It consists of 64 neuroprocessors and occupies 1.5 x 2.8 cm^2 silicon area.

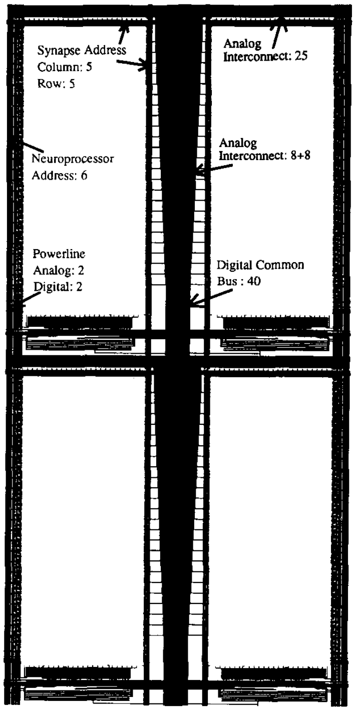


Fig. 13. The detailed layout of interconnects among four neuroprocessors.

TABLE I
Performance Comparison of Two Interconnect Methods

	Design Approach	
Performance	Using Bit-Parallel Analog Interconnect Method	Using Bit-Parallel Digital Interconnect Method
Chip Size (cm^2)	1.5 x 2.8	1.5 x 2.8
Number of Neuro-processors	64	12
Interconnection Routing Area (%)	23	85
Pin count	178	3,250
Network Ineration Time (ns)	522	250
Speed Performance (connection/second)	8.03 x 10^{10}	2.08 x 10^{10}

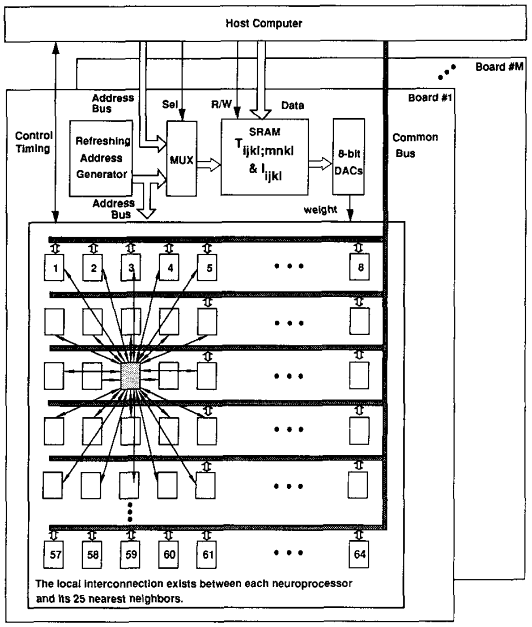


Fig. 14. The system diagram for high-speed motion detection using multiple VLSI neural chips. Each VLSI neural chips can accommodate 64 neuroprocessors with 1.2 μm CMOS technology and 1.5 x 2.8 cm^2 chip area. Each neuroprocessor in the VLSI neural chip can communicate with its neighbors through the analog point-to-point interconnections. The standard IC parts such as SRAM and 8-bit DAC are used for refreshing of synapse weights.

physical layout of the velocity-selective neuroprocessor for one image pixel using the scalable CMOS design rules is shown in Fig. 11. It occupies an area of $2,482 \times 5,636\lambda^2$ and contains 25 neurons, 25 x 27 synapse cells, and is able to detect the moving object with 25 different velocities. In the hardware implementation, two rows of synapses are used to increase the resolution of synapse weights coming from the bias inputs and also to enhance the fault tolerance of the network. With an advanced 1.2-μm CMOS technology, 64 neuroprocessors can be accommodated into one VLSI neural chip of 1.5 x 2.8 cm^2 in size. The chip layout is shown in Fig. 12. It requires a 178-pin PGA package. The analog interprocessor data communication requires 128 pins. The detailed layout of interconnects among four neuroprocessors is shown in Fig. 13. The interconnection routing area occupies 23% of the chip area. A performance comparison against the digital bit-parallel point-to-point interconnection method is listed in Table I. In the digital bit-parallel method, each data link requires 25 lines. Only 12 neuroprocessors can be

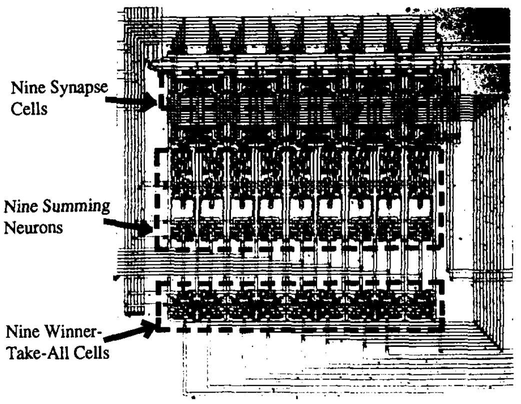


Fig. 15. The layout of the test module which includes key circuit blocks.

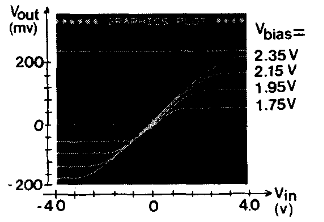


Fig. 16. Measured results of programmable synapse characteristics. A big bias voltage can provide a large dynamic range.

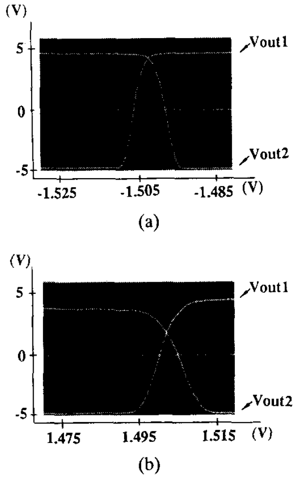


Fig. 17. Measured results of winner-take-all circuits. (a) one input as indicated in the x-axis changes linearly from –1.53 V to –1.48 V, the second input is connected to 1.5 V, and the other seven inputs are kept at –1.525 V. (b) one input as indicated in the x-axis changes linearly from 1.47 V to 1.52 V, the second input is connected to 1.5 V, and the other seven inputs are kept at 1.475 V.

accommodated in the same chip area and 85% of chip area will be used for the interconnection routing purpose.

With 128 VLSI neural chips and many supporting standard IC parts such as SRAM's and 8-bit DAC's for storing the weight information and dynamically refreshing of the synapse cells, computation of optical flow from an image with 64 × 128

TABLE II
CIRCUIT RESPONSE TIME

	Measured Results
Analog-to-Digital Conversion	73 ns
Synapse Multiplication	120 ns
Neuron thresholding	20 ns
Winner-Take-All Operation	38 ns
Encoder & Data Latch	8 ns
Digital-to-Analog Conversion	84 ns+, 263 ns*
Total	343 ns+, 522 ns*

Note: T_{ox} = 202 Å in MOSIS 1.2-um CMOS technology.
+ with an output loading of 5 pF
* with an output loading of 50 pF

TABLE III
PERFORMANCE OF VLSI MOTION ESTIMATION SYSTEM

Synapse Weight Loading Time (into SRAM)	2,080 us (50 ns per write)
Network Execution Time (for 36 iterations)	18.792 us
Neuron State Read Out Time	409.6 us (50 ns per read)
Total Processing Time	2.509 ms

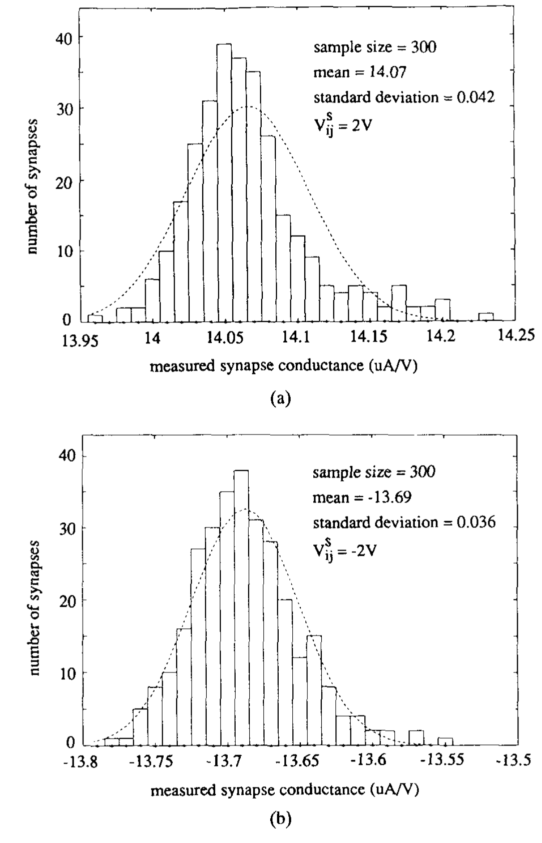
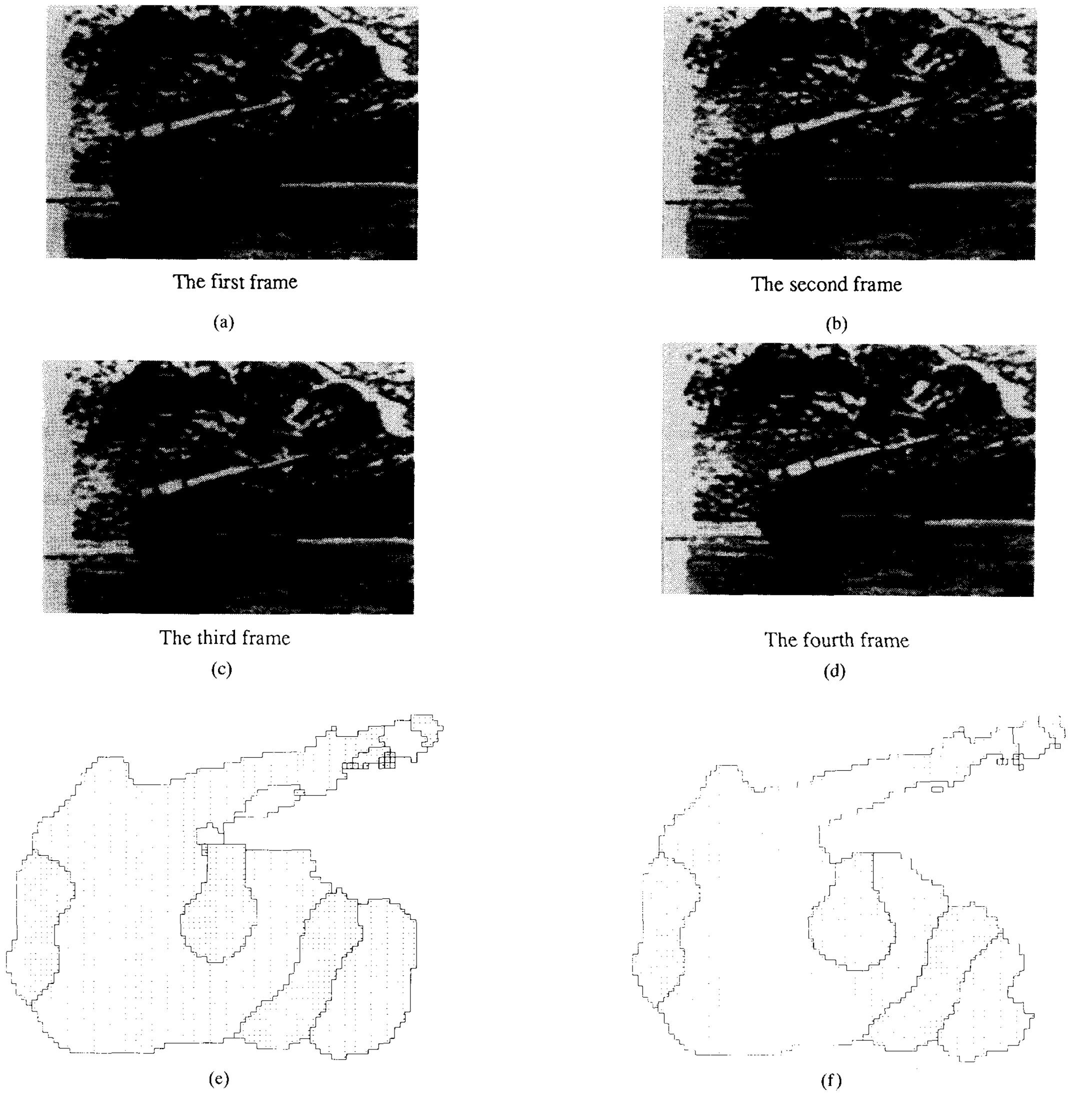


Fig. 18. The statistical distribution of measured synapse output conductances. (a) The synapse conductances can be described by a Gaussian distribution with a mean value of 14.07 μA/V and a standard deviation of 0.042 μA/V at weight voltage $V^s_{i,j}$ = 2 V. (b) The synapse conductances can be described by a Gaussian distribution with a mean value of –13.69 μA/V and a standard deviation of 0.036 μA/V at weight voltage $V^s_{i,j} = -2$ V.



The first frame

(a)

The second frame

(b)

The third frame

(c)

The fourth frame

(d)

(e)

(f)

Fig. 19. System-level analysis result on a sequence of four missile launcher images. (a) The first, (b) second, (c) third, and (d) fourth frame. (e) Obtained result using the parameters $A = 4$, $B = 850$, $C = 0$, $D_k = 7$, $D_l = 1$, and after 36 iterations. The effects of process variation on synapse weights are included through the Gaussian function. (f) Obtained result using same parameters as those in 19(e) except that the effects of process variation on synapse weights are not included.

pixels and 256 gray levels can be performed at a rate of 30 frames per second. The proposed system set-up for fast motion detection using multiple VLSI neural chips is shown in Fig. 14.

To obtain the electrical properties of the basic circuit blocks, a test structure containing key circuit components was fabricated with a 2-μm CMOS process from Orbit Semiconductor, Inc. through the MOSIS Service of USC/Information Sciences Institute at Marina del Ray, CA and tested. The picture of the test structure is shown in Fig. 15. Measured transfer curves of the synapse cell with different bias voltages are shown in Fig. 16. The dynamic range of the synapse cell is controlled by the bias voltage. Experimental data on the winner-take-all circuit are shown in Fig. 17. The circuit consists of nine winner-take-all cells. Two experiments were conducted. In Fig. 17(a), one input sweeps linearly from -1.53 to -1.48 V, the second input is connected to -1.5 V, and the other seven inputs are kept at -1.525 V. In Fig. 17(b), one input sweeps linearly from 1.47 to 1.52 V, the second input is connected to 1.5 V, and the other seven inputs are kept at 1.475 V. The winner-take-all function is successfully implemented with a resolution of 15 mV.

The processing time for one network iteration is around 522 ns. Each iteration cycle includes synapse multiplication, neuron summing, winner-take-all operation, data storage on latches, digital/analog and analog/digital conversion, and interprocessor data transfer. SPICE [54] simulation results on various circuit blocks are listed in Table II. The large response time of the synapse multiplication is due to the significant capacitance loading on the current-summation line. For the digital/analog conversion simulations, 5 pF and 50 pF effective capacitance loadings are estimated for interchip data communication and off-chip data communication, respectively. The major delay will come from the off-chip interprocessor data communication. The total computing power of 8.32×10^{10} connections per second can be achieved by using one VLSI neural chip containing 1600 neurons, 41 600 synapses cells, and operated at a master clock rate of 2 MHz. Based on the results of Table II the speed comparison of a system using 128 VLSI neural chips with a Sun-4/75 SPARC workstation is listed in Table III. The speedup factor is very large.

System-level analysis has been conducted to illustrate the performance of the motion detection chip. The mismatch effect of analog synapse components has been included. Fig. 18 shows the statistical distribution of measured synapse output conductances. A total of 300 synapses was measured. In Fig. 18(a), the synapse conductances can be described by a Gaussian distribution with a mean value of 14.07 μA/V and a standard deviation of 0.042 μA/V at weight voltage $V_{i,j}^s = 2$ V. In Fig. 18(b), the synapse conductances can be described by a Gaussian distribution with a mean value of -13.69 μA/V and a standard deviation of 0.036 μA/V at weight voltage $V_{i,j}^s = -2$ V. During computer analysis, the effects of process variation on synapse weights are included through the use of Gaussian function.

A set of four successive image frames directly produced by a Sony XC-77 CCD camera was used as the input data. Fig. 19(a)-(d) shows four successive image frames of a mobile missile launcher moving from left to right against a stationary background. The size of each image frame is 130×160 pixels. The maximum displacement of the mobile missile launcher between the time-varying image frame is 7 pixels. To estimate the principle curvatures and intensity values, a 5×5 window and a third order polynomial was used for all frames. By setting $A = 4, B = 850, C = 0, D_k = 7$, and $D_l = 1$, the velocity field was obtained after 36 iterations. The parameter A is set to 4, because four successive image frames are used. The parameter B is chosen by using trial-and-error method. The parameter C is set to 0 in the prototype design to simplify the neuron-state updating scheme of the network. Fig. 19(e) shows the final result of using synapse weights obtained by including the effects of process variation. Comparing with the result in Fig. 19(f), which the effects of process variation are not included, the motion information of the moving object still can be successful detected.

VI. Conclusion

A mixed-signal two-dimensional mesh-connected architecture for high-speed motion detection has been presented. A compact and efficient VLSI neuroprocessor which including 25 neurons and 25×27 synapse cells is able to estimate the motion of each pixel with 25 different velocities. Multiple neuroprocessors can be connected as a two-dimensional mesh to fully exploit the massively parallel computational power of neural networks. In this architecture, the local computation is processed in analog neuroprocessor and the local data communication is performed in parallel. Each 1.5×2.8-cm^2 VLSI neural chip from a 1.2-μm CMOS technology can operate at a rate of 83.2 giga connections per second.

Acknowledgment

Discussions with Dr. Y.-T. Zhou on artificial neural networks, and Dr. B. W. Lee on neural circuit design are highly appreciated. J. Choi provided detailed information on winner-take-all circuit design. The authors would like to thank the reviewers for valuable comments and suggestions.

References

[1] H. Koike *et al.*, "A 30 ns 64 Mb DRAM with built-in self-test and repair function," in *Tech. Dig. IEEE Int. Solid-State Circ. Conf.*, San Francisco, CA, Feb. 1992, pp. 150–151.

[2] L. Kohn and N. Margulis, "Introducing the Intel i860 64-bit microprocessor," *IEEE Micro Mag.*, vol. 9, no. 4, pp. 15–30, Aug. 1989.

[3] *SPARC Architecture Manual*, Sun Microsystems, Inc., Mountain View, CA, 1987.

[4] D. Dobberpuhl *et al.*, "A 200MHz 64b dual-issue CMOS microprocessor," in *Tech. Dig. IEEE Int. Solid-State Circ. Conf.*, San Francisco, CA, Feb. 1992, pp. 106–107.

[5] *Second-Generation TMS 320 User's Guide*, Texas Instruments, Inc., Dallas, TX, 1990.

[6] M. Griffin, G. Tahara, K. Knorpp, R. Pinkham, and B. Riley, "An 11-million transistor neural network execution engine," in *Tech. Dig. IEEE Int. Solid-State Circ. Conf.*, San Francisco, CA, Feb. 1991, pp. 180–181.

[7] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 1, pp. 3–28, 1978.

[8] R. W. Brodersen, "Low-power design of a wireless multimedia terminal," in *Proc. Elec. Eng. Dep. Res. Symp.*, University of California, Los Angeles, Feb. 1992.

[9] C. H. Mestrangelo and R. S. Muller, "A thermal absolute-pressure sensor with on-chip digital front-end processor," in *Tech. Dig. IEEE Int. Solid-State Circ. Conf.*, San Francisco, CA, Feb. 1991, pp. 188–189.

[10] Y. Tsividis, "Analog MOS integrated circuits—certain new ideas, trends, and obstacles," *IEEE J. Solid-State Circ.*, vol. 22, pp. 317–321, June 1987.

[11] M. Mahowald and C. Mead, "Silicon retina," in *Analog VLSI and Neural Systems*, C. Mead, Ed. Reading MA: Addition-Wesley Publishing Company 1989.

[12] R. Lyon and C. A. Mead, "An analog electronic cochlea," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. 36, pp. 1119–1134, July 1988.

[13] H. Kobayashi, J. White, and A. Abidi, "An analog CMOS network for Gaussian convolution with embedded image sensing," in *Tech. Dig. IEEE Int. Solid-State Circ. Conf.*, San Francisco, CA, Feb. 1990, pp. 216–217.

[14] B. Boser and E. Sackinger, "An analog neural network processor with programmable network topology," in *Tech. Dig. IEEE Int. Solid-State Circ. Conf.*, San Francisco, CA, Feb. 1991, pp. 184–185.

[15] T. Morishita, Y. Tamura, and T. Otsuki, "A BiCMOS analog neural network with dynamically updated weights," in *Tech. Dig. IEEE Int. Solid-State Circ. Conf.*, pp. 142–143, San Francisco, CA, Feb. 1990.

[16] J. Van der Spiegel, P. Mueller, D. Blackman, P. Chance, C. Donham, R. Etienne, and P. Kinget, "An analog neural computer with modular architecture for real-time dynamic computations," *IEEE J. Solid-State Circuits*, vol. 27, no. 1, pp. 82–92, Jan. 1992.

[17] S. Satyanarayana, Y. P. Tsividis, and H. P. Graf, "A reconfigurable VLSI neural network," *IEEE J. Solid-State Circuits*, vol. 27, pp. 67–81, Jan. 1992.

[18] D. Hammerstrom, "A VLSI architecture for high-performance, low-cost, on-chip learning," *IEEE/INNS Proc. Int. Joint Conf. Neural Networks*, vol. II, San Diego, CA, 1990, pp. 537–544.

[19] D. A. Orrey, D. J. Myers, and J. M. Vincent, "A high performance digital processor for implementing large artificial neural networks," *Proc. IEEE Custom Integrated Circuits Conf.*, San Diego, CA, May 1991, pp. 16.3.1–16.3.4.

[20] B. J. Sheu, "VLSI Neurocomputing with analog programmable chips and digital systolic array chips," presented at the *Int. Symp. Circuits and Systems (invited paper)*, Singapore, June 1991.

[21] K. S. Fu and T. Ichikawa, *Special Computer Architectures for Pattern Recognition*. Boca Raton, FL: CRC Press, 1982.

[22] K. Hwang and F. Briggs, *Computer Architecture and Parallel Processing.* New York: McGraw Hill: 1984.

[23] K. Przytula, W. Lin, and V. Kumar, "Partitioned implementation of neural networks on mesh connected array processors," in *VLSI Signal Processing IV*, H. Moscovitz, K. Yao, and R. Jain, Eds. Piscataway, NJ: IEEE Press, 1991.

[24] B. Horn, *Robot Vision.* Cambridge, MA: The MIT Press, 1986.

[25] K. Prazdny, "Egomotion and relative depth map from optical flow," in *Biological Cybernetics.* Berlin, Germany: Springer-Verlag, 1980, vol. 36, pp. 87-102.

[26] W. Simpson, "Depth discrimination from optical flow," *Perception*, vol. 17, pp. 497-512, 1988.

[27] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 17, pp. 384-401, July 1985.

[28] B. Ballard and O. Kimbal, "Rigid body motion from depth and optical flow," in *Computer Graphics and Image Processing.* New York: Academic Press, 1983, vol. 22, pp. 95-115.

[29] B. Horn and B. Schunck, "Determining optical flow," in *Artificial Intelligence.* Amsterdam, Netherlands: North-Holland Publishing Co., 1981, vol. 17, pp. 185-203.

[30] N. Grzywacz and A. Yuille, "Massively parallel implementations of theories for apparent motion," *Tech. Rep. AI Memo 888, CBIP Memo 016*, MIT Artificial Intelligence Lab. and Center for Biological Information Processing, June, 1987.

[31] J. Hutchinson, C. Koch, J. Luo, and C. Mead, "Computing motion using analog and binary resistive networks," *IEEE Computer Mag.*, pp. 52-63, Mar., 1988.

[32] Y. Zhou and R. Chellappa, "Computation of optical flow using a neural network," in *IEEE Proc. Int. Conf. Neural Networks*, San Diego, CA, 1988, vol. 2, pp. 71-78.

[33] Y.-T. Zhou and R. Chellappa, "A network for motion perception," in *Proc. IEEE/INNS Int. Joint Conf. Neural Networks*, San Diego, CA, June 1990, vol. II, pp. 875-884.

[34] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, pp. 1464-1480, Sept. 1990.

[35] T. Kohonen, *Self-Organization and Associative Memory, 2nd ed.* New York: Springer-Verlag: 1988.

[36] S. Grossberg, "Competitive learning: from interactive activation to adaptive resonance," *Cognitive Science*, vol. 11, pp. 23-63, 1987.

[37] B. O'Neil, *Elementary Differential Geometry.* New York: Academic Press: 1966.

[38] S. Ullman, *The Interpretation of Visual Motion.* Cambridge, MA: The MIT Press, 1979.

[39] B.W. Lee and B.J. Sheu, *Hardware Annealing in Analog VLSI Neurocomputing.* Boston, MA: Kluwer Academic Publishers, 1991.

[40] P. Denyer and D. Renshaw, *VLSI Signal Processing: A Bit-Serial Approach.* Reading, MA: Addison-Wesley, 1985.

[41] B.W. Lee and B.J. Sheu, "A compact and general-purpose neural chip with electrically programmable synapses," in *Proc. IEEE Custom Integrated Circuits Conf.*, Boston, MA, May 1990, pp. 26.61-26.6.4.

[42] B.J. Sheu, J. Choi, and C.-F. Chang, "An analog neural network processor for self-organizing mapping," *Tech. Dig. IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, Feb. 1992, pp. 136-137, 266.

[43] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network (ETANN) with 10240 "floating gate" synapses," *IEEE/INNS Proc. Int. Joint Conf. Neural Networks*, Washington, DC, 1989, vol. 2, pp. 191-196.

[44] B.W. Lee, B.J. Sheu, and H. Yang, "Analog floating-gate synapses for general-purpose VLSI neural computation," *IEEE Trans. Circuits Syst.*, vol. 38, June 1991.

[45] J. Dayhoff, *Neural Network Architectures.* New York: Van Nostrand Reinhold, 1990, pp. 97-114.

[46] D. Rumelhart and D. Zipser, "Feature discovery by competitive learning," in *Parallel Distributed Processing*, D. Rumelhart and J. McClelland, Eds. and PDP Group. Cambridge, MA: MIT Press 1986, pp. 151-193.

[47] R. Hecht-Nielsen, *Neurocomputing.* Reading, MA: Addison-Wesley, 1990, pp. 64-70.

[48] J. Choi, B.J. Sheu, and S.M. Gowda, "Analog VLSI neural network implementations of hardware annealing and winner-take-all functions," presented at the *34th Midwest Symp. Circuits and Syst. (invited paper)*, Monterey, CA, May 1991.

[49] J. Lazzaro, S. Ryckebush, M.A. Mahowald, and C.A. Mead, "Winner-take-all network of O(N) complexity," *Advances in Neural Information Processing Systems—I.* San Mateo, CA: Morgan Kauffman 1989, pp. 703-711.

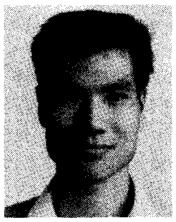
[50] A.G. Andreou, K.A. Boahen, P.O. Pouliquen, A. Pavasovic, R.E. Jenkins, and K. Strohbehn, "Current-mode subthreshold MOS circuits for analog VLSI neural systems," *IEEE Trans. Neural Networks*, vol. 2, pp. 205-213, Mar. 1991.

[51] A.B. Grebene, *Bipolar and MOS Analog Integrated Circuit Design.* New York: Wiley 1984.

[52] G. Lewicki, "Foresight: A fast turn-around and low cost ASIC prototyping alternative," presented at the *IEEE ASIC Conf. Exhibit*, Rochester, NY, Sept. 1990, pp. 6.8.1-6.8.2.

[53] C. Tomovich, "MOSIS—A gateway to silicon," *IEEE Circuits and Devices Mag.*, vol. 4, no. 2, pp. 22-23, Mar. 1988.

[54] B. Johnson, T. Quarles, A.R. Newton, D.O. Pederson, and A. Sangiovanni-Vincentelli, *SPICE3 Version 3E1 Users Guide*, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Apr. 1991.



Ji-Chien Lee (M'86) was born in Taiwan in 1954. He received the B.S. and M.S. degrees in electronics engineering from National Chiao Tung University, Hsin-Chu, Taiwan, in 1976 and 1980, respectively. He obtained the Ph.D. degree in electrical engineering from University of Southern California, Los Angeles, CA, in 1991.

From 1980 to 1985 he worked as a senior engineer on semiconductor memories and application-specific VLSI circuits at Electronics Research and Service Organization (ERSO) in Hsin-Chu, Taiwan. From 1985 to 1986, he studied in the Electrical Engineering Department at Carnegie Mellon University, Pittsburgh, PA. From 1986 to 1991, he was a graduate research assistant in Signal and Image Processing Institute at University of Southern California, working on image processing VLSI circuits. In addition, he helped to manage the VLSI Signal Processing Laboratory from 1990 to 1991. He received fellowship support form Industrial Technology Research Institute, Taiwan during the early years of his graduate study. He worked on analog integrated circuits in the Semiconductor Division of Hughes Aircraft Company in Irvine, CA from June to August, 1988. He is currently a Senior Scientist in Computer & Communication Research Labs. of Industrial Technology Research Institute, Hsin-Chu, Taiwan, working on visual information processing technologies. He has coauthored several papers on international scientific and technical journals and conferences. His research interests include high-speed communication, parallel computing, multimedia systems, and intelligent machines.

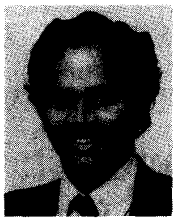
Bing J. Sheu (S'81–M'85–SM'91) was born in Taiwan in 1955. He received the B.S.E.E. degree (honors) in 1978 from the National Taiwan University, the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1983 and 1985, respectively.

In 1981, he was involved in custom VLSI design for a speech recognition system at Threshold Technology Inc., Cupertino, CA. From 1981 to 1982, he was a Teaching Assistant in the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. From 1982 to 1985, he was a Research Assistant in the Electronics Research Laboratory, University of California, Berkeley, working on digital and analog VLSI circuits. In 1985, he joined the faculty in Electrical Engineering Department at University of Southern California and is currently an Associate Professor. He is also an Honarary Consulting Professor for the National Chiao Tung University, Hsin-Chu, Taiwan. He has been an active researcher in several research organizations at USC including Center for Neural Engineering (CNE), Signal and Image Processing Institute (SIPI), Institute for Robotics and Intelligent Systems (IRIS), and DARPA-supported National Center for Integrated Photonic Technology (NCIPT). He serves as the Director of VLSI Signal Processing Laboratory. Since 1983, he has served as a consultant to the microelectronic and information processing industry. His research interests include advanced computer technologies, VLSI signal and image processing, artificial neural networks, portable communication circuits, and intelligent machines.

Dr. Sheu was a seven-time recipient of the Distinguished Book-Coupon Award while at the National Taiwan University. He also received the 1987 NSF Engineering Initiation Award and, at UC Berkeley, the Tse-Wei Liu Memorial Fellowship and the Stanley M. Tasheira Scholarship Award. He was also a recipient of the 1990 and 1991 Best Presenter Award at the IEEE International Conference on Computer Design. He has published more than one hundred papers in international scientific and technical journals and conferences and is a coauthor of the book *Hardware Annealing in Analog VLSI Neurocomputing* (Kluwer Academic Publishers). He has served on the Technical Program Committees and as the session chairman in IEEE/International Joint Conference on Neural Networks, International Conference on Computer Design, International Symposium on Circuits and Systems, Custom Integrated Circuits Conference, and on the VLSI Committee of the IEEE Signal Processing Society. He is on the editorial board of the Journal of Analog Integrated Circuits and Signal Processing. He also serves as a Guest Editor on custom VLSI technologies for the *IEEE Journal of Solid-State Circuits* (the March 1992 and March 1993 Issues). He serves as an Associate Editor and a Guest Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS and an Associate Editor of the *IEEE Circuits and Devices Magazine.* He is a senior member of the Signal Processing Society, Circuits and Systems Society, Communication Society, Computer Society of IEEE, a member of International Neural Networks Society, Phi Tau Phi, and Eta Kappa Nu.

Wai-Chi Fang (S'81–M'86) was born in Taiwan in 1956. He received the B.S. degree in electronics engineering in 1978 from National Chiao Tung University at Hsin-Chu, Taiwan, and the Master of Engineer and Ph.D. degrees in electrical engineering from University of Southern California in 1987 and 1992, respectively.

From 1988 to 1992, he studied VLSI implementation of image and video compression systems, digital neurocomputing and systolic array-based image understanding in the VLSI Signal Processing Laboratory at University of Southern California. Since 1985, he has also been with the Jet Propulsion Laboratory at Pasadena, California and worked on the architecture and design of high-performance computing and image processing systems. He used advanced CAD tools to generate applications-specific VLSI chips in the full-custom and standard-cell design styles based on the specially modified signal processing algorithms. He is currently involved in satellite image data compression for Cassini Titan Radar Mapper in the Radar Science and Engineering Division. He has published more than fifteen papers in scientific conferences and journals. His research interests include high-speed data compression for multimedia applications, artificial neural networks, image and video processing, signal processing for synthetic aperture radars, and VLSI system design.



Rama Challappa (S'78–M'79–SM'83–F'92) was born in Tanjore, Madras, India in 1953. He received the B.S. degree (with honors) in electronics and communication engineering from the University of Madras in 1975 and the M.S. degree (with distinction) in electrical communication engineering from the Indian Institute of Science, Bangalore, in 1977. He received the M.S. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN in 1978 and 1981, respectively.

From 1979 to 1981, he was a faculty research assistant at the Computer Vision Laboratory, University of Maryland, College Park. From 1981 to 1991, he was a faculty member in the Department of Electrical Engineering-Systems, University of Southern California. From 1988 to 1990, he was also the Director of Signal and Image Processing Institute at USC. Presently, he is a Professor in the Department of Electrical Engineering at the University of Maryland, where he is also affiliated with the Center for Automation Research, Computer Science Department and the Institute for Advanced Computer Studies. He has authored ten book chapters and over a hundred journal and conference papers. Many of his papers have been reprinted in collected works published by IEEE Press, IEEE Computer Society Press, and MIT Press. His current research interests are in signal and image processing, computer vision, and pattern recognition. He is the coeditor of two volumes of selected papers on digital image processing and analysis published in 1985. He was an Associate Editor for the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING from 1987 to 1989. Currently, he is a coeditor-in-chief of *Computer Vision, Graphics, and Image Processing: Graphic Models and Image Processing* and is an associate editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS and IEEE TRANSACTIONS OF IMAGE PROCESSING.

Dr. Challappa received a national scholarship from the Government of India during the period from 1969 to 1975, and he received the 1975 Jawaharlal Nehru Memorial Award from the Department of Education, Government of India. In addition, he received the 1985 National Science Foundation (NSF) Presidential Young Investigator Award and the 1985 IBM Faculty Development Award. In 1990, he received the Excellence in Teaching Award from the School of Engineering at USC. He was the general chairman of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition and of the IEEE Computer Society Workshop on Artificial Intelligence for Computer Vision, both held in San Diego, CA in June 1989. He was a program cochairman for the NSF-sponsored Workshop on Markov Random Fields, also held in San Diego in June 1989.