Analog Integr Circ Sig Process (2013) 75:267–277
DOI 10.1007/s10470-012-9980-y

An analog on-line-learning K-means processor employing fully parallel self-converging circuitry

Renyuan Zhang · Tadashi Shibata

Received: 16 February 2012 / Revised: 25 September 2012 / Accepted: 16 October 2012 / Published online: 3 November 2012

Abstract A hardware-efficient on-line-learnable processor was developed for the K-means clustering of highly dimensional vectors. Based on our proposed sample updating strategy, an incremental number of sample vectors can be clustered by a constant set of VLSI circuits. In order to speed up the learning process, we developed an analog fully parallel self-converging circuitry to implement the K-means algorithm. Upon receiving a sample vector on-line, the K-means learning autonomously proceeds and converges within a single system clock cycle (0.1 μs at 10 MHz). Furthermore, the chip-area and inner connection explosion problem was solved by using the proposed architecture. A proof-of-concept processor was designed and verified by the HSPICE and Nanosim simulations. The images from an actual database were used as learning samples in the form of 64 dimensional feature vectors. From the simulation results, all the samples were clustered into correct categories with a randomly ill initialization. In addition, the number of sample vectors can be freely increased.

Keywords K-means · On-line-learning · Highly dimensional · Fully parallel · Self-converge

1 Introduction

The K-means algorithm is one of the most typical unsupervised machine learning algorithms for pattern clustering [1], which was originally developed in the computing science area. Unsupervised learning algorithms have been successfully implemented for image processing problems such as the image segmentations [2] and image-sets categorizations [3] using software programs. As a machine learning algorithm, K-means clustering requires a self-learning process, which is very computationally expensive and time consuming. Consequently, VLSI implementations of the K-means algorithm [4, 5] were considered for the applications of portable electronic equipments or those requiring high speed performances.

Several works employing digital VLSI circuits realized image segmentation based on the K-means algorithm [6–9]. In these approaches, the Manhattan distance was employed as a distance measure. Another digital VLSI implementation [10] has even achieved the fast K-means clustering of five dimensional vectors based on the Euclidean distance calculation, which is much more expensive in silicon. However, considering high-level applications of image processing [3], the image-sets categorization for instance, the number of dimensions of sample vectors becomes much larger since the vector describes and represents a lot of important features in the entire image. Due to this reason, the distance calculation becomes much more expensive. Even if the powerful hardware as GPU is applied [11], it takes much time to calculate the distances between vectors having a large number of dimensions. One of possible solutions to speed up the learning processes is applying analog circuits to carry out the core functions, which has been applied in the supervised learning such as the support vector machine [12]. However, since the learning process requires a large number of numerical iterations, these clock-based-iterative implementations were still time consuming. To solve this problem, an analog fully parallel and self-converging implementation of machine learning was reported in our previous work [13] for SVM algorithm.

All these previously presented technologies have been developed for the fixed sample space. In the real world, the

R. Zhang (✉) · T. Shibata
Department of Electrical Engineering and Information Systems,
The University of Tokyo, 7-3-1 Hongo, Bunkyo,
Tokyo 113-8656, Japan
e-mail: tyoninen@if.t.u-tokyo.ac.jp

sample space is always huge, even unpredictably incremental. Thus, a on-line learning strategy [14] has been developed for the K-means algorithm. Its learning process kept receiving on-line samples when the sample space was expended. It has been proved that, this on-line learning strategy is helpful to deal with the incremental sample space and the ill initialization. However, this method was realized by a software program, and the number of dimensions of sample vectors was only two.

The purpose of this work is to implement the on-line learning K-means algorithm by VLSI circuits for the categorization of high dimensional sample vectors. A fast self-converging analog circuitry in fully parallel is used as a clustering learner [15]. A modified scheme of K-means algorithm is implemented by this learning circuitry, which is essentially similar to the original K-means mechanism but more efficient for the fully parallel implementation. In addition, an efficient on-line learning strategy is proposed based on this K-means learning circuitry with the consideration of a fixed hardware resource. In order to verify the proof-of-concept processor, the images of two objects selected from the COIL-20 database [16] are converted into 64-D feature vectors as learning samples. Upon receiving an on-line sample vector, the learning process autonomously proceeds and self-converges within 0.1 μs at a system clock frequency of 10 MHz. According to the Nanosim simulation results, all the images were categorized into correct classes with a randomly ill initialization.

The rest parts of this paper are organized as follows: an efficient on-line learning K-means algorithm is proposed in the Sect. 2; in the Sect. 3, the processor organization and its operational principle are presented; the Sect. 4 shows the HSPICE and Nanosim simulation results for the performances of our proposed analog circuitry and the whole processor, respectively; conclusions are made in the Sect. 5.

2 Algorithm applied in this work

The K-means algorithm is widely used in pattern recognition, data mining, and image segmentation as a very powerful learning tool. It is a typical unsupervised clustering method. Using this algorithm, learning samples given in the form of multidimensional vectors can be partitioned into a limited number (K) of clusters according to their feature similarity without supervision.

2.1 Basic idea of the original K-means algorithm

The basic idea of original K-means algorithm is illustrated in Fig. 1. The dotted circles represent the categorization form in each step along with the locations of centroid vectors marked by the stars. The arrow lines reflect distances between pattern vectors and centroid vectors in

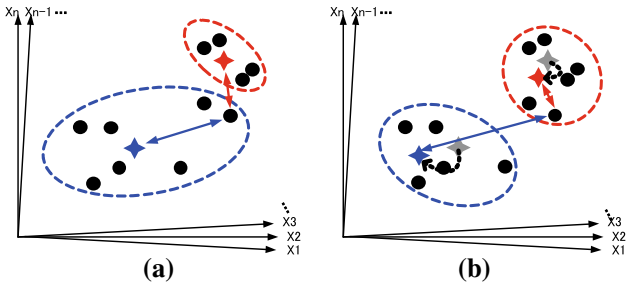

Fig. 1 Basic idea of original K-means clustering algorithm

this sample space. A large number of sample vectors $\mathbb{X} = \{x_1, x_2, \ldots, x_n\}$ with n dimensions are given. Considering the K value as 2, the target is to cluster the sample vectors into two categories. The unsupervised learning process is described as follows:

(1) Initializing the categorization randomly and computing the centroid vector for each cluster;
(2) Evaluating the distances between a specific vector and all centroid vectors as shown in Fig. 1(a);
(3) Resetting the categorization label for this specific vector according to the distance comparisons;
(4) Repeating steps 2 and 3 till all the given vectors have been re-clustered for this round;
(5) Computing the new centroid vectors [the change of centroid is illustrated by the location shift of star mark in Fig. 1(b)] and back to step 2;
(6) Results converging and stopping.

Many different types of mathematical representations were introduced to evaluate the distance $D(\mathbb{X}_i, \mathbb{X}_j)$ between vectors $\mathbb{X}_i$ and $\mathbb{X}_j$ such as Euclidean Distance. In some previously proposed VLSI implementations of K-means, the Manhattan Distance was employed since it is computationally simpler than the Euclidean Distance. However, the complex computations for high-dimensional vectors still should be repeated for heaps of times, including average-vector calculations (to search the centroid) and the distance evaluations. These calculations for vectors are usually hungry for the hardware and computing-time especially when the number of vector dimensions is large. Due to this reason, the original K-means algorithm can be hardly implemented by VLSI circuits with fully-parallel learning strategy. In addition, upon receiving a new sample vector, all the operations should be repeated. Thus, the on-line learning strategy is difficult to efficiently apply in this algorithm.

2.2 Hardware-efficient version of K-means algorithm for fully-parallel learning strategy

In order to avoid iterating calculations of vectors, a modified version of K-means algorithm was proposed in this work. The basic idea of our proposed algorithm is

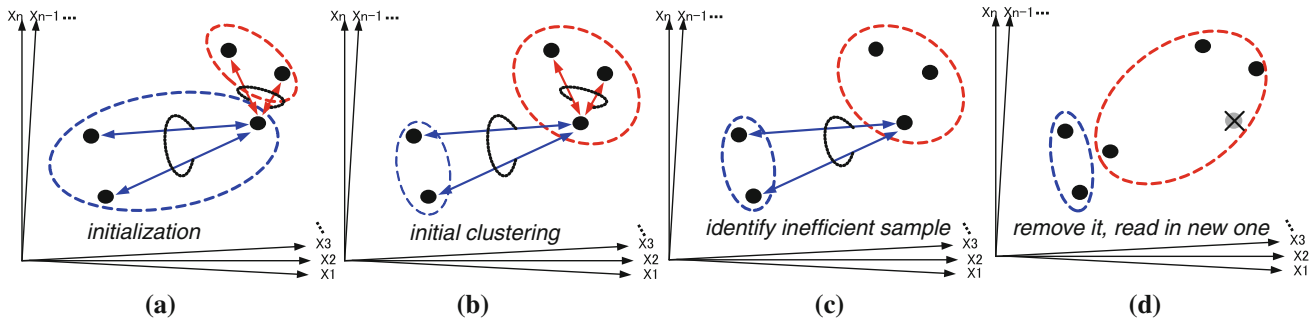

Fig. 2 Modified version of the K-means algorithm applied in this work

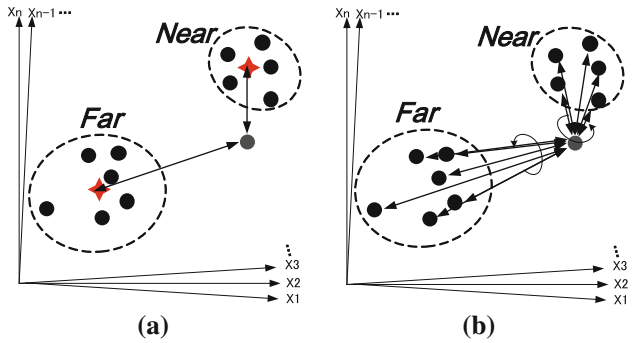

Fig. 3 Distance evaluations from a specific sample vector to different clusters **a** using the representatives of centroid vectors and **b** using the average calculations

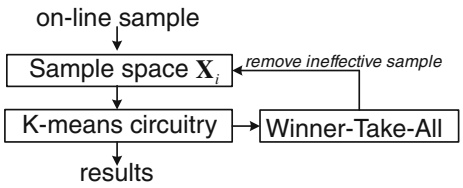

Fig. 4 Organization of proposed on-line learning K-means processor

illustrated by Fig. 2. Instead of iterating operations to search centroid vectors and calculate distances between vectors, all the vector calculations are carried out at the beginning and kept for the whole learning process. The Euclidean Distances from every vector to all the others are calculated (in parallel by analog VLSI circuits in this work) and stored in the form of scalars, which can be directly reused during the learning process. In this manner, the only operation needed to be iterated is changing the category label for each vector, which is easily realized by analog VLSI circuits and possible to be proceeded in parallel. Upon receiving a new sample vector on-line, the efficiency of each current sample vector is evaluated according to: $\min_i \sum_{j(y_j \neq y_i)} \alpha_j D(\mathbb{X}_i, \mathbb{X}_j)$, where y_i is the category label of the i-th sample. The most inefficient sample is updated by the coming sample. The learning process flow is given as follows:

(1) Setting up an initial sample space with a small size;
(2) Calculating the Euclidean Distances from every vector to all the others and initializing the categorization randomly as shown in Fig. 2(a);
(3) Evaluating the similarities from a specific vector to all the categories (by calculating the average distances from the specific vector to all the vectors from the same category);
(4) Resetting the categorization label for this specific vector according to the evaluation as shown in Fig. 2(b);
(5) Repeating steps 3 and 4 till all the given vectors have been re-clustered;
(6) Results converging and updating the sample space as shown in Fig. 2(c), (d).

During the learning operation, the only calculation is to compute the average values among scalars. Therefore, the chip area and processing time for learning processor can be reduced. In the actual VLSI implementation by this work, the similarity evaluations and the label resetting are both proceeded in vector-parallel rather than one by one based on the clock cycle. The temporal results are immediately feedback to the circuity and autonomously converge.

2.3 Comparison

Regarding the original K-means algorithm, the centroid vector (or called gravity) of each cluster is represented when the clustering scheme is changed by learning. The distances from any specific sample vector to the gravities of all the clusters are calculated. Essentially, the distances from a specific sample to all the clusters are evaluated by this calculation. These distance evaluations are compared, and the smallest one is selected to determine the new cluster label of this sample vector. Namely, the representatives of centroid vectors even the values of distances have no direct effects on the updating, but the comparison among distance evaluations is important. In our modified scheme of K-means algorithm, the distances from a specific

sample to different clusters are evaluated by the average calculation of distances from this sample to all the members in a cluster. Comparing Fig. 3(a) with (b), these two types of evaluations give the same comparisons of distances from a specific sample to different clusters in general cases. Therefore, two types of evaluations lead to the same updating operation in those cases. The learning operation will stop and converge till the cluster label for each sample does not change, which is the same as original K-means mechanism. From this point of view, the suggested scheme of clustering method has similar convergence performance to the original K-means algorithm.

3 Hardware implementation

3.1 Processor organization

The proof-of-concept processor is designed to realize the on-line learning K-means clustering function. Figure 4 illustrates the processor organization, which consists of an analog fully-parallel self-converging K-means circuitry and a winner-take-all (WTA) circuit. Sixteen sample vectors with 64 dimensions are used as a fixed sample space. The analog K-means circuitry clusters these sample vectors into two categories in real-time. The WTA circuit is introduced to evaluate the current sample space, and identify the most inefficient sample, which will be replaced by an new coming sample on-line.

3.2 Analog K-means learning circuitry

In this implementation, N vectors represented by $\mathbb{X}_i$s with high-dimensions (64-dimensional in this actual work) can be autonomously clustered into two categories after a fast K-means learning process. The $Label_i$ with a binary form (one bit for two-category implementations) is introduced to reflect the category index of the ith given vector respectively.

The analog K-means learning circuitry is mainly composed of three blocks using analog VLSI circuits as shown in Fig. 5. Block I contains N sets of distance calculators, which evaluate the distance between two vectors $\mathbb{X}_i$ and $\mathbb{X}_j$ as following equation:

$$D(\mathbb{X}_i, \mathbb{X}_j) = |\mathbb{X}_i - \mathbb{X}_j|^2 = \sum_{k=1}^{64} (x_{ik} - x_{jk})^2, \quad (1)$$

where, $\mathbb{X}$ is a 64-dimensional vector as $\mathbb{X} = \{x_1, x_2, \ldots, x_{64}\}$. The evaluation results are given in current mode and stored in the analog current memory array as block II row by row in time sequence. Each memory cell is connected with two switches (in fact, the number of switches depends on the number of categories for a specific application) controlled by its respective category label. Since the evaluation results are read out in current mode, it is easy to sum them up for each category by a bus wire. The analog dividers in block III are built to compute the average distance from a specific vector to all the others belonging to a same category. As it is shown in Fig. 2, the N_A and N_B represent the numbers of vectors in category A and category B, respectively. According to the distances from a given vector to two categories, the categorization result for this vector in the form of a binary label is output by using a simple analog comparator. All the calculations and operations are carried out using analog circuits. Thus, the operations can be accomplished in real-time and in parallel by compact circuits. Since only the comparisons among distances are concerned rather than the accurate values, the accuracy of analog circuits is acceptable in this implementation.

During the learning process, the categorization results in the form of binary labels are immediately feedback into the circuitry to control the switches in block II and count the number of N_A and N_B in block III. These switching operations are also easy to be realized in real-time by analog VLSI circuits. The dynamics of analog free feedback realizes the mechanism of step iterations, which was mentioned in the algorithm description. Without any additional controlling mechanism, the learning process is accomplished autonomously and self-converges with very high speed (within single clock cycle). The block II contains much more cells compared with other blocks. However, the values stored and processed by block II are all in the form of scalars, which means the chip area of single cell in block II is much smaller than that of block I. Furthermore, it is easy to collect the information by row bus-wires. Therefore, the chip area and interconnect explosion problem can be prevented. Theoretically, a triangular array in block II is sufficient to store necessary values since $D(\mathbb{X}_i, \mathbb{X}_j) = D(\mathbb{X}_j, \mathbb{X}_i)$. However, this distance value contributes to the label updating of two different patterns. The switches and wire connection associated to these two contributions are separate. The elimination of the redundant capacitors does not greatly reduce the chip area of this block, but increase the complexity of inner connection. Therefore, a full array is designed in block II. In the case of category expansion, more bus-wires for categories could be added.

Our proposed analog circuit to calculate the distance between two vectors $\mathbb{X}_i$ and $\mathbb{X}_j$ with 64 dimensions is shown by Fig. 6. In order to make it flexible and able to combine with digital circuits even computer programs, we also design a digital-to-analog converter (DAC) for the processor. The 64-D pattern vectors are converted into analog voltage signals. For instance, the vector $\mathbb{X}_i$ can be

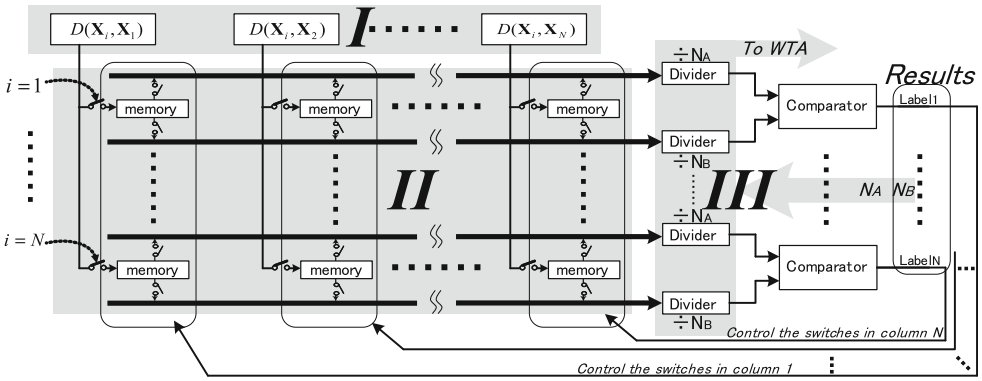

Fig. 5 Architecture of proposed fully-parallel self-converging K-means learning circuitry with a configuration of 2-category clustering

represented by voltages as $\{v_{i1}, v_{i2}, \ldots, v_{i64}\}$. Assuming $v_{i1} > v_{j1}$, the potentials v_1 and v_2 of node 1 and 2 will be shifted by a small enough current bias I_b as following equations:

$$\begin{aligned} v_1 &\approx v_{i1} + v_{thn} + |v_{thp}|, \\ v_2 &\approx v_{j1} + v_{thn} + |v_{thp}|, \end{aligned} \tag{2}$$

where, v_{thn} and v_{thp} are the threshold voltages of n-type and p-type MOS transistors respectively. Considering $v_{i1} > v_{j1}$, the transistors in the branch where current I_1 flows are in the strong inversion region, the transistors in the branch where current I_2 flows are in the weak inversion region. Then, $I_1 \gg I_2$ and the current I_1 can be obtained according to the following equation:

$$I_1 = K_n(v_1 - v_3 - v_{thn})^2 = K_p(v_3 - v_{j1} - |v_{thp}|)^2, \tag{3}$$

where, K_n and K_p are the amplifier factors of n-type and p-type MOS transistors respectively. Solving this equation with the consideration of Eq.2, the potential of node 3 v_3 can be expressed as:

$$v_3 = \frac{v_{i1}\sqrt{\frac{K_n}{K_p}} + v_{j1} + |v_{thp}|\sqrt{\frac{K_n}{K_p}} + |v_{thp}|}{1 + \sqrt{\frac{K_n}{K_p}}}. \tag{4}$$

Substituting Eq. 4 into Eq. 3, the current flowing through this branch can be given by:

$$I_1 = \alpha(v_{i1} - v_{j1})^2, \tag{5}$$

where, $\alpha = \frac{K_n K_p}{(\sqrt{K_n}+\sqrt{K_p})^2}$. Since $I_1 \gg I_2$, the total current generated by the 1-D squaring circuit is almost equal to I_1 ($I_1 + I_2 \approx I_1$). In the case of $v_{i1} < v_{j1}$, similar derivation can be applied to obtain:

$$I_1 + I_2 \approx I_2 = \alpha(v_{j1} - v_{i1})^2. \tag{6}$$

Generally, the current generated by the 1-D squaring circuit is $\alpha|v_{i1} - v_{j1}|^2$. Collecting the current generated by all dimensions, the final output to evaluate $D(\mathbb{X}_i, \mathbb{X}_j)$ can be obtained as the following equation:

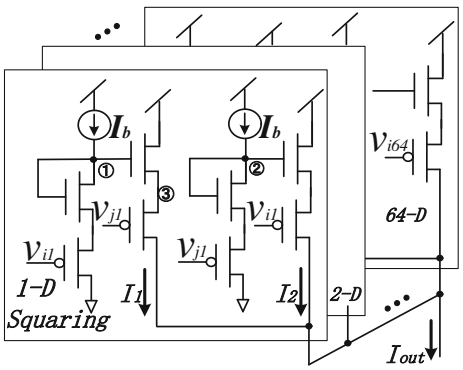

Fig. 6 Circuit schematic of the distance calculator for 64-D vectors

$$I_{out} = \alpha \sum_{k=1}^{64} |v_{ik} - v_{jk}|^2. \tag{7}$$

This current which evaluates the distance $D(\mathbb{X}_i, \mathbb{X}_j)$ is stored in an analog current memory as illustrated in Fig. 7. Each memory cell is combined with two analog switches controlled by the categorization label (the number of switches depends on the number of categories). The label of a specific vector determines the current value stored in its respective cell should contribute to category "A" or "B". The bus-wires collect all results in current mode and feed them into the analog dividers to calculate the average distances from this vector to all the others of each category.

An advanced topology of Translinear circuit [17] is applied in this work as an analog divider as shown in Fig. 8. According to the Translinear principle, the output current of this circuit is obviously obtained by: $I_{out} = I_{in}\frac{I_r}{I_{cnt}}$, where I_r is a constance current reference and I_{cnt} can be generated by our proposed vector counter. The switches in this counter are controlled by the labels for all the vectors. For instance, the I_{cnt} generated by the counter for category A can be obtained as $I_{cnt} = N_A \cdot I_r$, where N_A is the number of vectors from category A. Finally, the output current is given by: $I_{out} = \frac{I_{in}}{N_A}$ (for the dividers on bus-wire of category A) or $I_{out} = \frac{I_{in}}{N_B}$ (for the dividers on bus-wire of category B). These average values are used to generate new labels by

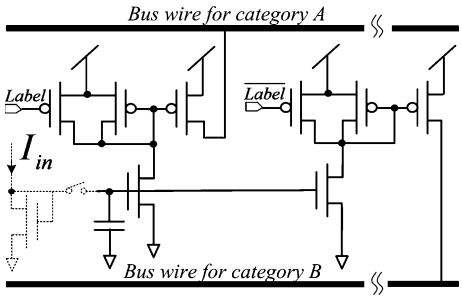

Fig. 7 Analog current memory cell with switching function

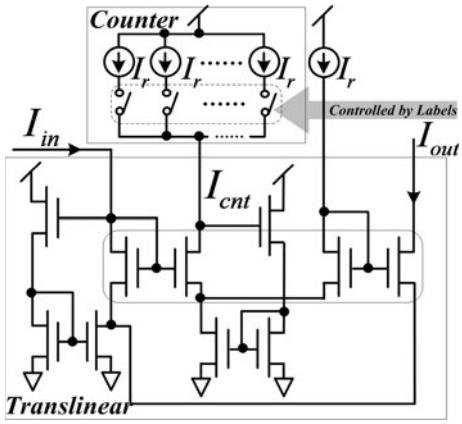

Fig. 8 An advanced Translinear circuit [17] as an analog divider with the vectors counter

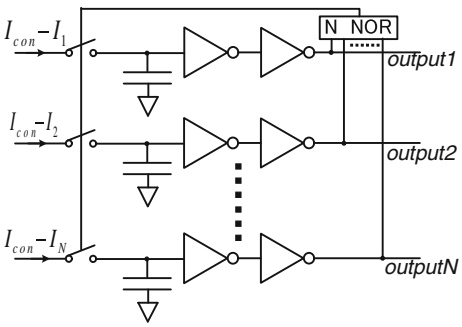

Fig. 9 Winner-take-all circuit to evaluate the efficiency of samples

the simple comparators. The new labels are immediately feedback to the operational blocks till results converge.

3.3 Winner-take-all circuit to evaluate the efficiency of samples

A winner-take-all (WTA) circuit shown in Fig. 9 is designed to select the most inefficient sample, which is the closest one to the opposite cluster. The current I_i represents the average distance from the i-th sample to all the samples of the different cluster $\sum_{j(y_j \neq y_i)} D(\mathbb{X}_i, \mathbb{X}_j)/N_{diff}$, where N_{diff} is the number of samples in the different cluster. The block

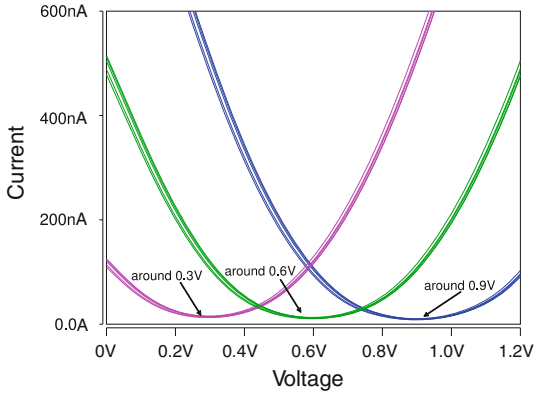

Fig. 10 Performances of distance calculator (the output current against input voltage with different center values)

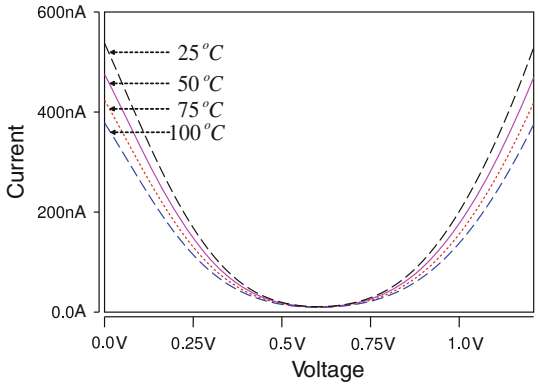

Fig. 11 Temperature effects on the Euclidean distance calculation

"NOR" is used to accept the output from each candidate, and broadcast the result to control the charging switches for all the candidates. All the outputs are initialized as "0" at the beginning. Then, the charging switches are all turned on. When the voltage on any one of capacitors reaches the threshold of buffer inverter, the result of block "NOR" becomes "0". As a result, all the switches are turned off, and the outputs are locked.

4 Experiments

A 0.18 μm 1.8 V CMOS technology is employed to construct the proof-of-concept processor for 64-D vectors categorization. The HSPICE and Nanosim simulations were introduced to verify the performances of our proposed analog circuits and the on-line learning K-means processor.

4.1 Performances of the analog K-means learning circuitry

The performances of analog circuits applied in our proposed processor are verified by the HSPICE simulations. Considering

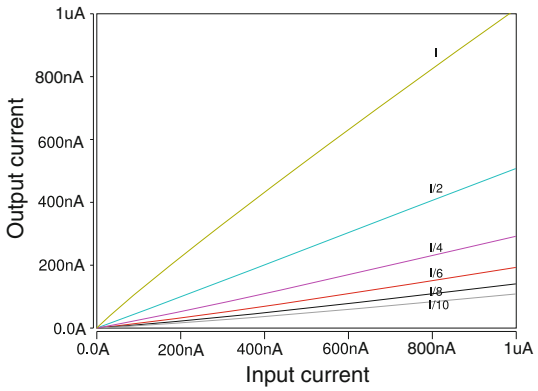

Fig. 12 Performances of analog divider (the output current against input current)

the distance between vectors $\mathbb{X}_i$ and $\mathbb{X}_j$, the kth dimension was selected randomly. The voltage v_{jk} was set as a "center" value. For other dimensions, $v_{il} = v_{jl}|_{l \neq k} = 0$. By sweeping the voltage v_{ik}, the output current can be obtained as shown in Fig. 10, which reflects the distance generated by this dimension.

The performance robustness against fabrication process variations is also shown in this simulation. The Monte Carlo simulation with five trials is performed to verify the mismatch problem. We focus on the threshold voltage V_{th} for this simulation since it is the most sensitive factor to process variations. Considering the model proposed by the previous works [18], the variation of V_{th} can be described by the following equation:

$$\sigma^2_{\Delta V_{th}} = \sigma^2_{\Delta V_{th}}|_{V_{BS}=0}\left(1-\frac{V_{BS}}{\phi_B}\right) + \frac{2\sqrt{q^3\epsilon_{si}\phi_B N_A}}{3WLC^2_{ox}}\left(\sqrt{1-\frac{V_{BS}}{\phi_B}}-\left(1-\frac{V_{BS}}{\phi_B}\right)\right). \quad (8)$$

The variation of V_{th} is related to the mismatch problem during the fabrication process and the dynamic bias V_{BS}. From the results of some related works [18], the worst case of V_{th} variation was evaluated as $\sigma(\Delta V_{th}) \approx 3\% V_{th}$ in a 0.18 μm technology, where V_{th} is the typical value of the threshold voltage. In this simulation, it is assumed that the

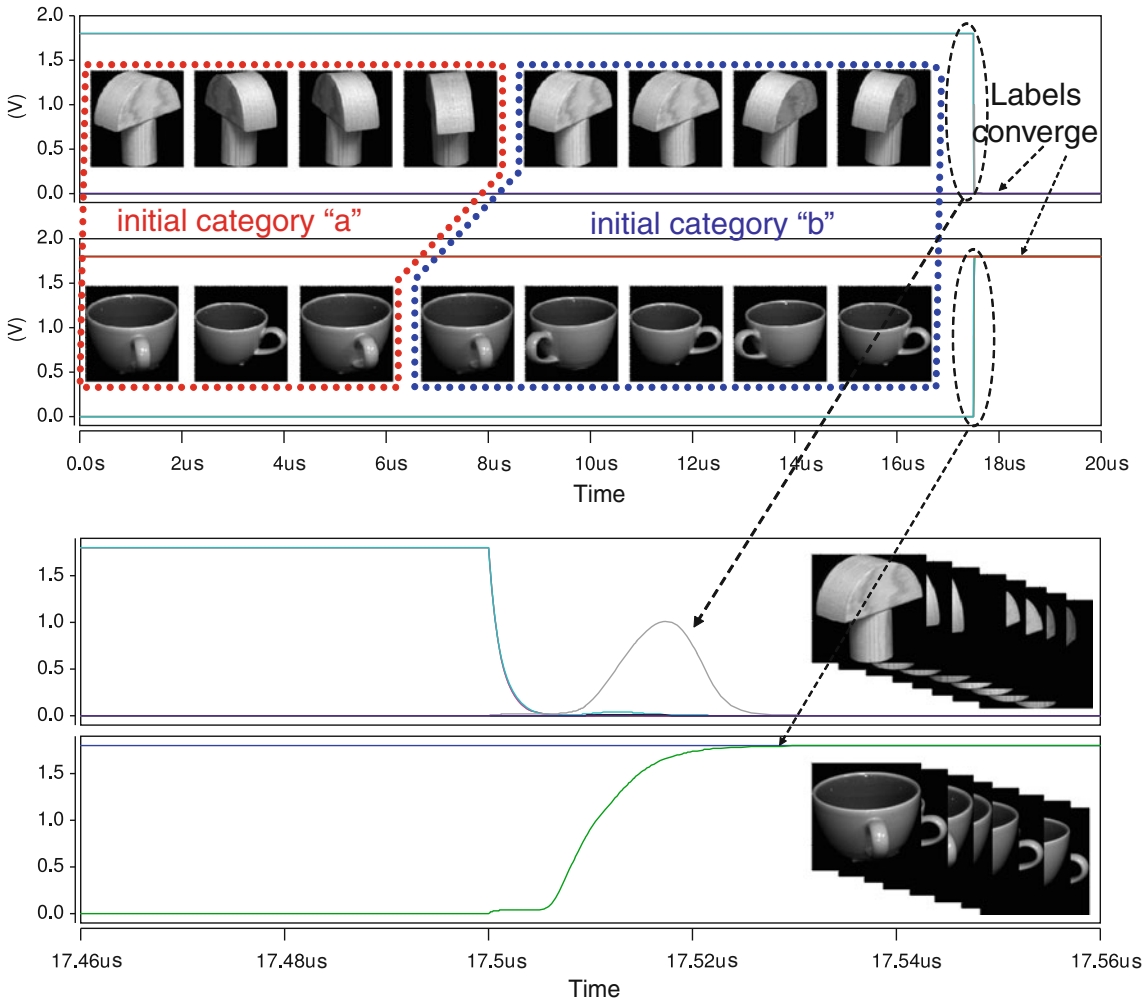

Fig. 13 Learning performances with an illy random initialization of category labels

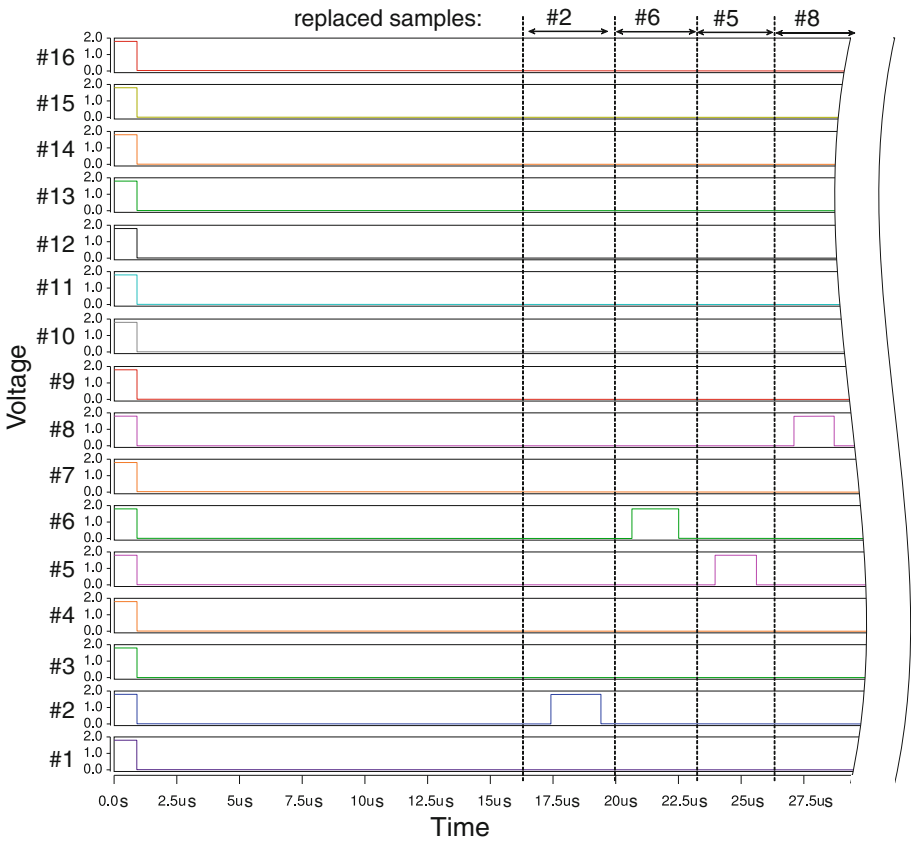

Fig. 14 Indexes to select the most inefficient sample

variations follow a Gaussian distribution. An excessive value of $\sigma(\Delta V_{th})$ is set as 5 %. The fluctuation of Euclidean distance calculation is illustrated in Fig. 10.

Figure 11 shows the temperature effects on the Euclidean distance calculation. The operational temperatures are set as $25, 50, 75$ and $100\,^{\circ}\mathrm{C}$ for this simulation. From the simulation results, the temperature effects on the Euclidean distance calculation are approximately symmetrical to the center value. Namely, the temperature has equal effect on the calculation of each dimension for every learning sample, which is negligible to the learning result theoretically.

The simulation results shown in Fig. 12 present the performances of the analog divider, which carries out the average current $I_{out} = \frac{I_{in}}{N}$, where N is the amount of vectors. By setting different amounts of $N = 1$, 2, 4, 6, 8 and 10 for instances, the output current with the ratios of $I_{out} = I_{in}, \frac{I_{in}}{2}, \frac{I_{in}}{4}, \frac{I_{in}}{6}, \frac{I_{in}}{8}$ and $\frac{I_{in}}{10}$ can be obtained as presented in Fig. 12.

The images selected from an actual database COIL-20 are employed to verify our proposed fully-parallel learning architecture. Before the learning process, all the images are converted into 64-dimensional vectors applying the PPED method [19] (projected principle edge distribution). A set of classic serial DACs has also been designed to translate this digital information into analog voltage signals. In this work, ten clock cycles are needed to convert a pattern vector with a bit-length of eight. The Nanosim simulation results are presented by Fig. 13 with the illy random initialization for category labels. The binary signals "1" (1.8 V in this work) and "0" (0 V) are used to reflect the category label for each vector. Four images of "umbrella" and three images of "cup" are initialized as category "a" by setting their label voltages as 0 V; other four images of "umbrella" and five images of "cup" are initialized as category "b" by setting their label voltages as 1.8 V. After the learning session, the label voltages for all the images of "umbrella" and "cup" become 0 and 1.8 V, respectively. The learning process autonomously proceeds and self-converges within one clock cycle (0.1 μs as a system clock frequency of 10 MHz). The label signals during the learning process are zoomed in and shown in the lower windows of Fig. 13. Without any additional controlling mechanism, they are self-reset for several rounds and completely converges. The dynamic fluctuation of these signals reflects the temporary alteration of category labels due to the ill initialization. This phenomenon is comparable with the flips of labels in the numerically iterative

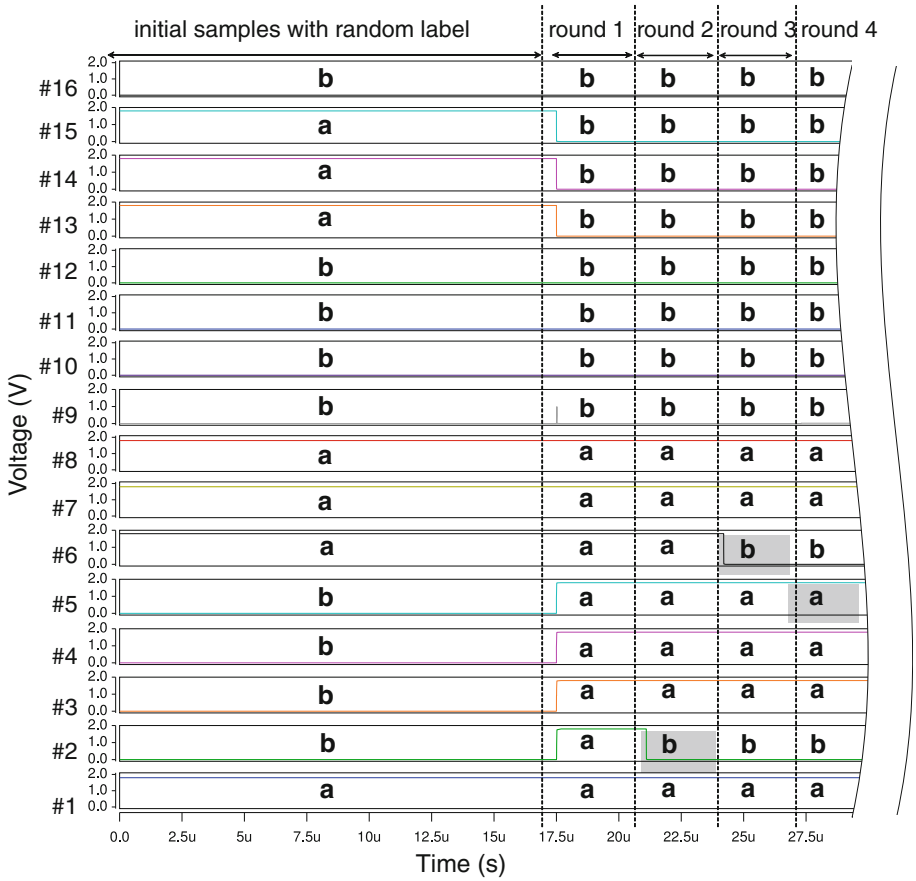

Fig. 15 On-line learning process in different rounds. The *shadow* marks a sample which has been replaced by a new sample during the previous round

Table 1 Performance comparisons

	Ref. [8]	Ref. [10]	This work
Implementation	FPGA and CPU	Digital	Analog
Number of devices	N/A	414 K gates	20 K Trans
Distance measurement	Manhattan	Euclidean	Euclidean
Number of dimensions	2	18	164
Number of iterations	25	16	Self-converging
Speed (vectors/s)	$<4.93 \times 10^6$*	1.38×10^6	10×10^6
Number of samples	2,905	76.8×10^3	On-line

* Some of the calculations were pre-processed by the software program

approaches. In the traditional applications based on clock-driven iterations, the learning speed are seriously related to the amount of iterations [20]. Assuming the amount of iterations and learning samples are l and N, respectively, the learning process requires a number of clock cycles as $l \times N$ for clock-based works. In other words, the learning process of our proposal is $l \times N$ times faster than the traditional clock-based processors at least.

4.2 Simulation results of on-line learning process

The K-means on-line learning processor has a fixed hardware set for sixteen sample vectors. A small number of sample vectors are initially read in the learning circuitry, and the learning operation proceeds for the first round. During each round of learning process, the most ineffective sample is indexed by a selection-signal as shown in Fig. 14. Upon receiving a new sample on-line, this ineffective sample will be replaced by the new sample according to the selection-index. When the sample space is updated, the learning operation proceeds as the next round. The cluster-label for each sample is presented in Fig. 15 in several on-line learning rounds. Considering plenty of different learning samples (random initialization and on-line-updating samples), the learning operations successfully converge in every round of the entire on-line sequence. The shadow marks a sample which has been replaced by a new sample during the previous round. From the Nanosim simulation results, all the on-line test samples are clustered into correct categories.

4.3 Comparisons

The performance comparisons among this work and some other works are shown in Table 1. The learning complexity is greatly related to the number of dimensions of the sample vectors. Thus, previously reported works [8, 10] were difficult to be implemented in the highly dimensional applications. By using the proposed architecture, a larger number of dimensions and samples, even a higher learning speed were achieved with less complexity of hardware. Furthermore, the self-convergence is helpful to improve the learning performances compared with the traditional approach, which stops learning by setting a fixed number of iterations.

4.4 Reliability

Generally, the accuracy of calculational analog circuits is poorer than that of software programs and digital circuits. Plenty of factors influence the precision of the storage and calculations for analog signals, which might lead to defective results of K-means learning.

The error of calculations is mainly resulted by the mismatch and deviation problems on all the devices (especially the threshold voltages of MOS transistors) as it is shown in Fig. 10. On the other hand, the storage of sampling voltages also results in errors. Since the capacitors are used as analog memories, the switch channel charges and the leakage current of capacitors cause the voltage drops ΔV_{sw} and ΔV_{leak}, respectively. In the K-means learning circuitry, voltages representing the element values of patterns are broadcasted to block I controlled by sixteen sets of switches. Considering the originally stored voltage as V_{orig}, the effect of switch channel charge can be described as $\Delta V_{sw}/V_{orig} = \frac{16C_{sw}}{C_{sampling}+16C_{sw}}$, where $C_{sampling}$ and C_{sw} are the voltage sampling capacitance and the equivalent capacitance due to switch channel charge, respectively. In this sense, large sampling capacitance and small size of switching transistors are suggested to reduce the voltage drops. The voltage drops due to the leakage current of sampling capacitors are related to the entire time of usage. Therefore, this effect appears when the proposed processor operates in the on-line learning mode and the number N of on-line patterns is very large. Since the digital data with a bit-length of eight is used as input, the expected accuracy of analog voltages is $V_{DAC}/256$ (about 4 mV in this work) after D/A conversion. The information of patterns will be correctly kept in the analog memories if the voltage drop is smaller than 4 mV. Assuming the leakage current density and capacitance density are J and C_{den}, respectively, the reliable condition becomes $\frac{10NJ}{C_{den}f} < \frac{V_{DAC}}{256}$, where f is the operational frequency.

The effects of all these errors to the clustering results are also related to the learning patterns. Fortunately, in the K-means theory, the similarity comparisons among patterns are concerned rather than the absolutely accurate calculations. In some related works [21], it is reported that the poor calculational accuracy has acceptable effect on K-means learning result.

5 Conclusions

An analog on-line learning K-means processor was developed for the categorization of highly dimensional vectors. Upon receiving a new sample vector on-line, the K-means processor proceeds the learning operation within a single system clock cycle (0.1 μs), and the sample space is updated. By using the proposed architecture, a fully parallel and self-converging circuitry is constructed within a compact chip area. An images database was introduced to verify the learning performances of K-means processor. From the circuit simulation results, all the vectors which represent real images are self-clustered into correct categories on-line.

Acknowledgments The authors would like to thank the VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Rohm Corporation, Toppan Printing Corporation, Synopsys, Inc., Cadence Design Systems, Inc. and Mentor Graphics, Inc.

References

1. MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability*, (pp. 281–297).
2. Ray, S., & Turi, R. H. (1999). Determination of number of clusters in K-means clustering and application in colour image segmentation. In *Proceedings of 4th International Conference on Advance Pattern Recognition Digital Technology*, (pp. 137–143).
3. Gordon, S., Greenspan, H., & Goldberger, J. (2003). Applying the information bottleneck principle to unsupervised clustering of discrete and continuous image representations. In *Proceedings of the 9th International Conference on Computer Vision 1*, (pp. 370–377).

4. Maliatski, B., & Yadid-Pecht, O. (2005). Hardware-driven adaptive K-means clustering for real-time video imaging. *IEEE Transactions on Circuits and Systems for Video Technology, 15*(1), 164–166.
5. Chen, T.-W., & Chien, S.-Y. (2010). Bandwidth adaptive hardware architecture of K-means clustering for video analysis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 18*(6), 957–966.
6. Chen, T.-W., Sun, C.-H., Bai, J.-Y., Chen, H.-R., & Chien, S.-Y. (2008). Architectural analyses of K-means silicon intellectual property for image segmentation. In *Proceedings of IEEE International Symposium on Circuits and Systems*, (pp. 2578–2581).
7. Maruyama, T. (2006). Real-time K-means clustering for color images on reconfigurable hardware. In *Proceedings of International Conference on Pattern Recognition*, (pp. 816–819).
8. Hussain, H. M., Benkrid, K., Seker, H., & Erdogan, A. T. (2011). FPGA implementation of K-means algorithm for bioinformatics application: an accelerated approach to clustering microarray data. In *NASA/ESA Conference on Adaptive Hardware and Systems*, (pp. 246–255).
9. Ma, Y., & Shibata, T. (2010). A binary-tree hierarchical multiple-chip architecture for real-time large-scale learning processor systems. *Japanese Journal of Applied Physics, 49*, 04DE08–04DE08-8.
10. Chen, T.-W., & Chien, S.-Y. (2011). Flexible hardware architecture of hierarchical K-means clustering for large cluster number. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 11*(8), 1336–1345.
11. Chiosa, I., & Kolb, A. (2011). GPU-based multilevel clustering. *IEEE Transactions on Visualization and Computer Graphics, 17*, 132–145.
12. Kang, K., & Shibata, T. (2010). An on-chip-trainable Gaussian-kernel analog support vector machine. *IEEE Transactions on Circuits and Systems, 57*, 1513–1524.
13. Zhang, R., & Shibata, T. (2012). Fully parallel self-learning analog support vector machine employing compact Gaussian-generation circuits. *Japanese Journal of Applied Physics, 51*(4), 04DE10-1–04DE10-7.
14. Wang, C., Lai, J., & Zhu, J. (2010). A conscience on-line learning approach for Kernel-based clustering. In *IEEE Conference on Data Mining*, (pp. 531–540).
15. Zhang, R., & Shibata, T. (2011). An analog K-means learning processor employing fully-parallel self-converging circuitry. In *International Analog of VLSI Workshop*, (pp. 91–96).
16. Nene, S. A., Nayar, S. K., & Murase, H. (1996). Columbia Object Image Library (COIL-20). Technical Report CUCS-005-96.
17. Minch, B. A. (2008). MOS translinear principle for all inversion levels. *IEEE Transactions on Circuits and Systems, 55*, 121–125.
18. Croon, J. A., Rosmeulen, M., Decoutere, S., Sansen, W., & Maes, H. E. (2001). A simple characterization method for MOS transistor matching in deep submicron technologies. *IEEE Proceedings of International Conference on Microelectronic Test Structures, 14*, 213-218.
19. Yagi, M., & Shibata, T. (2003). An image representation algorithm compatible with neural-associative-processor-based hardware recognition systems. *IIEEE Transactions on Neural Networks, 14*(5), 1144–1161.
20. Wang, X., & Leeser, M. (2007). K-means clustering for multispectral images using floating-point divide. In *IEEE 15th International Symposium on Field-Programmable Custom Computing Machines*, (pp. 151–162).
21. Chen, T.-W., Sun, C.-H, Su, H.-H., Chien, S.-Y., Deguchi, D., Ide, I., et al. (2011). Power-efficient hardware architecture of K-means clustering with Bayesian-information-criterion processor for multimedia processing applications. *IEEE Journal of Emerging and Selected Topics in Circuits and Systems, 1*(3), 357–368.

Reyuan Zhang received the B.S. degree in electrical engineering from Tongji University, Shanghai, China, and the M.S. degree in electronic engineering from Wasada University Japan, in 2007 and 2010, respectively. He is now a candidate of doctor degree in electronic engineering at the University of Tokyo, Japan. His research interests include hardware implementations of human-like intelligence, machine learning and analog VLSI circuits design.

Tadashi Shibata was born in Hyogo, Japan, on September 30, 1948. He received the B.S. degree in electronic engineering and the M.S. degree in material science from Osaka University, Osaka, Japan, and the Ph.D. degree from the University of Tokyo, Tokyo Japan, in 1971, 1973 and 1984, respectively. From 1974 to 1986, he was with Toshiba Corporation, where he worked as a researcher on the R&D of VLSI device and processing technologies. He was engaged in the development of microprocessors, EEPROMs, and DRAMs, primarily in the process integration and the research of advanced processing technologies for their fabrication. From 1984 to 1986, he worked as a Production Engineer at one of the most advanced manufacturing lines of Toshiba. During the period of 1978 to 1980, he was Visiting Research Associate with Stanford Electronics Laboratories, Stanford University, Stanford, CA, where he studied laser beam processing of electronic materials including silicides, polysilicon, and superconducting materials. From April 1986 to May 1997, he was an Associate Professor with the Department of Electronic Engineering, Tohoku University, where he was engaged in the research and development of ultra-clean technologies. His main interest was in the area of low-temperature processing utilizing very-low-energy ion bombardment for promoting thin-film growth processes as well as in the development of the ultraclean ion implantation technology to form defect-free ultra-shallow junctions by low-temperature annealing. Since the invention of a new functional device Neuron MOS Transistor in 1989, he has been engaged in the development of new-concept circuits and systems working in analog-digital merged decision-making principle. In May 1997, he became Professor with the Department of Information and Communication Engineering, The University of Tokyo. From April 1999 to March 2008, he was Professor with the Department of Frontier Informatics, School of Frontier Science, The University of Tokyo. Since April 2008, he has been a Professor with the Department of Electrical Engineering and Information Systems, School of Engineering, The University of Tokyo. His current research interest is to develop human-like intelligent computing systems based on the state-of-the-art silicon technology and the psychologically-as well as biologically-inspired model of the brain. Dr. Shibata is a member of Japan Society of Applied Physics, the Institute of Electronics, Information, and Communication Engineers, and the IEEE Electron Devices Society, Circuits and Systems Society and Computer Society.