PAPER Special Section on Parallel and Distributed Computing and Networking

Acceleration of FDTD Method Using a Novel Algorithm on the Cell B.E.

Sho ENDO^{†a)}, Jun SONODA^{††b)}, Motoyuki SATO^{†††}, and Takafumi AOKI[†], Members

SUMMARY Finite difference time domain (FDTD) method has been accelerated on the Cell Broadband Engine (Cell B.E.). However the problem has arisen that speedup is limited by the bandwidth of the main memory on large-scale analysis. As described in this paper, we propose a novel algorithm and implement FDTD using it. We compared the novel algorithm with results obtained using region segmentation, thereby demonstrating that the proposed algorithm has shorter calculation time than that provided by region segmentation.

key words: FDTD method, Cell B.E., DMA transfers, parallel computing with region segmentation

1. Introduction

The finite-difference time-domain (FDTD) method [1], which is very time-consuming, is widely used for electromagnetic (EM) wave analysis. It has been implemented on a supercomputer [2], a PC cluster [3], and a dedicated computer [4], [5] to accelerate FDTD processing.

The Cell Broadband Engine processor (Cell B.E.) has also been developed by IBM Corp., Sony Corp., and Toshiba Corp. as a heterogeneous multi-core processor. The Cell B.E. has a computational capability of 204.8 Giga floating point number operations per second (G-FLOPS) for single-precision operations and 14.6 G-FLOPS for doubleprecision.

The Cell B.E. has been used for FDTD method calculations. B. Li et al. [7] implemented the two-dimensional FDTD method using two-level parallelization, distributed memory, and SIMD parallelization. They used local stores on the SPE as distributed memory, and achieved linear speedup along with increased SPEs. However, the analysis region size has remained limited by the local store capacity. M. Xu et al. [8], [9] implemented the two-dimensional FDTD method on the IBM Blade QS20 (3.2 GHz) and reported computation speed that is 14.14 times that of an Opteron (AMD Inc.) processor and 7.05 times that of an Opteron processor. The analysis region was allocated to the

Manuscript received November 19, 2010.

Manuscript revised June 9, 2011.

[†]The authors are with the Graduate School of Information Sciences, Tohoku University, Sendai-shi, 980–8579 Japan.

a) E-mail: endo@aoki.ecei.tohoku.ac.jp

main memory, which was used as shared memory. The SPEs perform FDTD computation by exchanging blocks between local stores and the main memory. In fact, DMA transfers are necessary between a local store on the SPE and the main memory. Actually, this has come to represent a bottleneck of FDTD calculations. As a result, it has been impossible to exploit the computational power of the Cell B.E. This problem has been pointed out for FDTD on the shared-memory architecture [6].

As described in this paper, we have applied a novel method to the FDTD method on the Cell B.E. The major contribution of this paper is the improvement of scalability by solving the bottleneck of DMA transfers.

In the novel method, the SPEs are used as a pipeline. Although the pipelined implementations of FDTD method have been presented in the area of special purpose hardware design and FPGAs [4], [5], they are pipelined only at the level of arithmetic operation. In contrast, the FDTD algorithm is pipelined in our scheme. This pipelined structure enables exploitation of the locality of the data on the memory, and enables performance of FDTD computation with fewer DMA transfers. Consequently, the bottlenecks of DMA transfers described in [9] are solvable and are expected to accelerate the FDTD method. The proposed algorithm also solves the problem in the paper [7] it makes it possible to compute a numerical model that exceeds the size of the local store. As an evaluation of the algorithm, we compared the execution time of the proposed algorithm and conventional region segmentation.

The paper is organized as follows. Section 2 introduces main features of Cell B.E. and describes a novel algorithm used for this study. Section 3 presents some experimentally obtained results for evaluating the performance of the proposed algorithm. In Sect. 4, we end this paper by explaining some salient conclusions.

2. Implementation of the FDTD Method Using a Novel Algorithm on the Cell B.E.

2.1 FDTD Method

As described in this paper, we have considered 2D processing. An electric field E and the magnetic field H are defined as $E = (0, 0, E_z)$ and $H = (H_x, H_y, 0)$. The Maxwell equations can be written as follows:

$$\nabla \times E = -\mu \frac{\partial H}{\partial t} \tag{1}$$

^{††}The author is with the Department of Intelligent and Electronic Systems, Sendai National College of Technology, Sendaishi, 989–3128 Japan.

^{†††}The author is with Center for Northeast Asian Studies, Tohoku University, Sendai-shi, 980–8576 Japan.

b) E-mail: sonoda@sendai-nct.ac.jp

DOI: 10.1587/transinf.E94.D.2338

$$\nabla \times \boldsymbol{H} = \varepsilon \frac{\partial \boldsymbol{E}}{\partial t} + \sigma \boldsymbol{E} \tag{2}$$

The update equations can be written as the following equations by application of central differences to time and space derivatives.

$$E_{z}|_{i,j}^{n+1} = C_{a}E_{z}|_{i,j}^{n} + C_{b}\left\{-H_{x}|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} + H_{x}|_{i,j-\frac{1}{2}}^{n+\frac{1}{2}} + H_{y}|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - H_{y}|_{i-\frac{1}{2},j}^{n+\frac{1}{2}}\right\}$$
(3)

$$H_{x}\Big|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} = H_{x}\Big|_{i,j+\frac{1}{2}}^{n-\frac{1}{2}} + C_{c}\left\{E_{z}\Big|_{i,j+1}^{n} - E_{z}\Big|_{i,j}^{n}\right\}$$
(4)
$$H_{z}\Big|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} = H_{z}\Big|_{i,j+\frac{1}{2}}^{n-\frac{1}{2}} - C_{z}\left\{E_{z}\Big|_{i,j+1}^{n} - E_{z}\Big|_{i,j}^{n}\right\}$$
(5)

$$H_{y}\Big|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} = H_{y}\Big|_{i+\frac{1}{2},j}^{n-\frac{1}{2}} - C_{c}\left\{E_{z}\Big|_{i+1,j}^{n} - E_{z}\Big|_{i,j}^{n}\right\}$$
(5)

Here, the standard Yee notation has been adopted: the superscripts represent the time iteration, whereas subscripts represent the spatial location of the field components. Coefficients C_a , C_b , and C_c are described as follows.

$$C_a = \frac{1 - \frac{\sigma \Delta t}{2\varepsilon}}{1 + \frac{\sigma \Delta t}{2\varepsilon}}, \quad C_b = \frac{\frac{\Delta t}{\varepsilon \Delta}}{1 + \frac{\sigma \Delta t}{2\varepsilon}}, \quad C_c = \frac{\Delta}{\mu \Delta t} \tag{6}$$

In those equations, Δ is the cell size. Coefficients C_a , C_b , and C_c are calculated only once at the beginning of analysis. Therefore, the number of operations per cell is given as the number of operations in Eq. (3), (4), and (5).

2.2 Organization of the Cell Broadband Engine

As described in this paper, we accelerate the FDTD method using the Cell B.E. Figure 1 presents the organization of the Cell B.E. The Cell B.E. has a computational capability of 204.8 G-FLOPS for single-precision operations and 14.6 G-FLOPS for double-precision. The Cell B.E. consists of one power processor element (PPE), eight synergistic processor elements (SPEs), I/O Module, and memory controller. Figure 1 shows that the PPE, SPEs, memory controller, and I/O Modules are connected to the ring bus called the element interconnect bus (EIB), which has a bandwidth of 204.8 Gigabytes per second (GB/s). The PPE, SPEs, and Main memory controller mutually communicate through direct memory access (DMA) transfers via the EIB. The Cell B.E. presents the benefit that it is able to calculate quickly through parallel computing using eight SPEs. It also has a wide-band bus. However, as with other processors, the time for access to the main memory is a bottleneck of its execution time.

2.3 Parallel Computing on the Cell B.E. Using Region Segmentation

Parallel computing in previous works [8] is a method which assigns the analysis region to each SPEs and performs parallel computing with multiple SPEs. In the ideal condition, parallel computing using region segmentation can reduce the calculation time in proportion to an increase in the number of SPEs.



Fig. 1 Organization of the Cell Broadband Engine.

Figure 2 (b) presents the computation method on the large-scale analysis. On large-scale analysis, the whole analysis region cannot be allocated to local storages on SPEs. Therefore, it is allocated to main memory. The analysis region is divided into small blocks. They are loaded sequentially onto the SPE and processed. Each SPE stores four blocks on its local store. The calculation of FDTD method consists of the following steps:

- 1. A block is loaded from the main memory.
- 2. The SPE waits until the adjacent block is loaded.
- 3. The SPE performs a calculation of the block.
- 4. A block is stored in the main memory.

Computation of the electromagnetic field on the boundary of the FDTD cell requires the field on the adjacent block. Therefore, step 2 is included. Large-scale analysis reveals a problem: DMA transfers increase along with rise in the number of SPEs. Then they become a bottleneck.

2.4 Acceleration of the FDTD Method Using the Proposed Algorithm

As described in this paper, we have implemented the FDTD method using the proposed algorithm. Moreover, we have tried to resolve the bottleneck of DMA transfers. The novel method is the method in which the SPEs are used as a pipeline. It can reduce DMA transfers to the main memory.

Figure 3 is revealed in the sequence of calculation on the proposed and conventional method. For conventional region segmentation, shown in Fig. 3 (a), blocks on the analysis region are assigned to each SPE and are calculated simultaneously. Therein, blocks that are calculated simultaneously belong to the same time step. Here, the term "time step" is the discretized time in the FDTD method, not the cycle in the processor. On the other hand, blocks belonging to different time steps are calculated simultaneously using the proposed algorithm. At the SPE of the first stage, time steps of blocks are updated from n to n + 1. They are updated at the next stage from n + 1 to n + 2. This computation sequence enables the SPEs to use blocks from the SPE of the prior stage.

Figure 4 shows the flow of data that occurs with the novel algorithm. The values of FDTD cells are stored in the main memory. The major difference between the conventional and the proposed method is that each SPE is assigned a time step that is assigned a part the of analysis region in



Fig. 2 Parallel computing using region segmentation.



Fig.3 Sequence of calculation on the parallel computing with region segmentation and the proposed algorithm.

the conventional method. As presented in Fig. 4, at the first stage, the blocks are loaded from the main memory, then its time step is updated from n to n + 1. On the second stage, the blocks are loaded from the first stage. Then its time step is updated from n + 1 to n + 2. Blocks are stored in the main memory from the final stage. The blocks are exchanged between the SPEs using DMA transfer. The DMA transfers between two SPEs do not become a bottleneck of performance because the EIB is sufficiently fast. Using the proposed algorithm, the SPEs access the main memory only twice: at the first stage and the last stage. For that reason, access to the main memory does not increase concomitantly with the increase of SPEs. Therefore, the bottleneck of memory access is solvable using the proposed algorithm.

3. Results and Discussion

3.1 Conditions

Figure 5 portrays the 2-D free-space model we used. The electric field *E* and the magnetic field *H* are defined as *E* = $(0, 0, E_z)$ and *H* = $(H_x, H_y, 0)$, where the medium of the analysis model is vacuum and non-dispersive. In Fig. 5, the electric hard source as the sinusoidal wave $E_z(t) = \sin \omega t$ is set on the line $x = 42.0\lambda$. Therein, ω is $2\pi \times 10^9$ rad/s. We used a PlayStation 3 (Sony Corp.) with Cell B.E. (3.2 GHz) and main memory of 256 MB. We used gcc 4.1.1 for compilation, and Yellow Dog Linux 6.0 as the operating system (OS). On the PlayStation 3, the SPEs we can use are limited to six because one SPE is disabled at the hardware level, the OS occupies an SPE. We also have compared the execu-



Fig. 4 Dataflow for the proposed algorithm.



Fig. 5 Numerical model used for analysis.

tion time of the FDTD method on the Cell B.E. with those obtained using general processors. We have also used an Apple Mac Pro Workstation, which has two Xeon 2.8 GHz processors and main memory of 6 GB. To compare the execution time on a processor level, one of two processors was used. The FDTD program was compiled with gcc 4.3.2, and was run on Ubuntu 7.10. Both on the Cell B.E. and workstation, double-precision floating-point operations and variables were used. The cells constituting the numerical model are $4200 \times 4200 = 17,640,000$ cells. The data size of the model is $17,640,000 \times 8$ B = 135 MB. The data size of model is limited by the main memory size. On the PlayStation 3, which has 256 MB main memory, the number of cells is limited to 5792×5792 cells.

3.2 Comparison of Execution Time of the Proposed Algorithm with that of Parallel Computing with Region Segmentation

First, we demonstrated the correctness of the proposed algorithm through the experiment of calculating FDTD method



Fig. 6 Calculated E_v value at the observation point.

and showed that there is no violation of data dependence on the proposed algorithm. Figure 6 shows the electric field E_z obtained with the FDTD method using the conventional and the proposed algorithm. The exact value is calculated from the solution of the Maxwell equation under the condition in Fig. 5.

$$E_z = \sin(\omega t - ky) = \sin(\omega t - 2\pi \times 4.2) \tag{7}$$

The cell size is $\Delta = \lambda/100$ and the time step is $\Delta t = 7.07 \times 10^{-12}$. Here, the symbol λ denotes the wavelength. In the FDTD method, the cell size is usually less than $\lambda/20$. Here we used $\Delta = \lambda/100$. The time step is determined to meet the CFL condition as $\Delta t \leq \Delta/(c\sqrt{2})$. In Fig. 6, the electric field using the proposed algorithm shows good agreement with the exact value. For the proposed algorithm, equations of the FDTD method are the same as those on the conventional FDTD method. Therefore, the result is equal to that of the conventional FDTD method.



Fig. 7 Execution time on the proposed and conventional algorithm.

Figure 7 presents the execution time for the twodimensional FDTD method on the Cell B.E. The proposed algorithm shows a shorter calculation time than that obtained using the conventional method. This difference results from the bottleneck of the DMA transfers. In fact, the proposed method gives linear speedup because it has no DMA transfer bottleneck. However, the decrease of the execution time on the conventional method saturates at four SPEs because of the DMA transfer bottleneck.

Compared with the workstation, the proposed algorithm with 6 SPEs shows speed that is 1.7 times greater.

3.3 Theoretical Comparison of Novel and Methods of Parallel Computing with Region Segmentation

In this section, we have determined the theoretical expression of the execution time for both methods. We have calculated the number of SPEs in which the increase of performance is saturated and have compared the experimentally obtained results presented in the Sect. 3.2. Here, the get is the DMA transfer from the main memory to SPEs, and the put is that from SPEs to the main memory.

First, we figure out the time for operation of a cell as t_{op} , which can be expressed as

$$t_{\rm op} = \frac{n_{\rm cycle}}{f_{\rm clock} n_{\rm SIMD}},\tag{8}$$

where n_{cycle} represents the processor cycles for the calculation of a cell, f_{clock} is a frequency of CPU clock (3.2 GHz), and n_{SIMD} denotes the number of operations performed simultaneously by SIMD operations (For the double-precision arithmetic instruction on the Cell B.E., $n_{\text{SIMD}} = 2$). As shown in Table 1, we can estimate the number of the Cell B.E. instructions required for updating a FDTD cell from the Eq. (3), (4), and (5). In Table 1, we counted load instructions by counting the variables in the equation. We can update a FDTD cell with 8 addition/subtraction, 4 multiplication, 15 load, and 3 store instructions. On the Cell B.E., a DFMA instruction is available, which performs the operation $X = Y \times Z + W$ (*X*, *Y*, *Z*, *W* are the floating point

 Table 1
 Number of instructions required for each value of the field.



Fig. 8 Data flow graph for Eq. (3).

number) in an instruction. Figure 8 portrays the data flow graph for Eq. (3). We can perform operations D and F in one DFMA instruction according to Fig. 8. For that reason, we can process a value of an electric field with three addition/subtraction instruction, one multiplication, and one dfma instruction. Similarly, we can process a value of magnetic field with one addition/subtraction instruction and one dfma instruction. Double-precision arithmetic (Addition/subtraction and multiplication) instructions cost seven cycles; a load/store instruction costs one cycle. Therefore, we can calculate the number of cycles n_{cycle} as

$$n_{\text{cycle}} = \{3 + 1 + 1 + 2 \times (1 + 1)\} \times 7 + (15 + 3) \times 1$$

= 9 × 7 + 18 = 81. (9)

Next, we work out the time of DMA transfers per cell. The DMA transfer time is the amount of data divided by the bandwidth of the main memory. On the parallel computing with region segmentation, t_{DMAp} is given as shown below.

$$t_{\text{DMA}p} = l_d \left(\frac{n_g}{BW_g} + \frac{n_p}{BW_p} \right) \tag{10}$$

Therein, BW_g and BW_p respectively show the get and put bandwidth. On the Cell B.E. at 3.2 GHz, $BW_g = BW_p =$ 25.6 GB/s [10]. n_g and n_p respectively stand for the amounts of data per cell that are gotten and put. In addition, l_d signifies the length of each datum (For double-precision floating point number, $l_d = 8$). Using the proposed algorithm, only two DMA transfers are necessary for the calculation: get on the SPE of first stage and put on the last stage. The calculation on the SPE of an intermediate stage requires no access to the main memory. Therefore, DMA transfers are reduced when SPEs increase. Consequently, the DMA transfer time t_{DMAn} is expressed as presented below.





$$t_{\text{DMA}n} = l_d \left(\frac{n_g}{BW_g} + \frac{n_p}{BW_p} \right) \frac{1}{n_{\text{SPE}}}$$
(11)

Next we calculate the number of SPEs where the increase of performance is saturated. Figure 9 presents the theoretical execution time shown in Eq. (8), (10), and (11). The Fig. 9 is shown on a log–log scale. The execution time is the time that is the longer one of Eq. (8) and (10). Therefore, n_{sat} is the number of SPEs for which the speedup is saturated in the conventional method.

Finally, we determined n_{SPE} from t_{op} and $t_{\text{DMA}p}$. The value of t_{op} is calculable as shown in Fig. 9.

$$t_{\rm op} = \frac{n_{\rm cycle}}{f_{\rm clock} n_{\rm SIMD} n_{\rm SPE}} = \frac{81}{3.2 \times 10^9 \cdot 2 \cdot n_{\rm SPE}} = \frac{1.27 \times 10^{-8}}{n_{\rm SPE}} {\rm s}$$
(12)

The computation of $t_{\text{DMA}p}$ requires the value of n_g and n_p . The calculation of the electric field requires five variables $(E_z, H_x, H_y, C_a \text{ and } C_b)$. That of the magnetic field uses four variables $(E_z, H_x, H_y \text{ and } C_c)$. Therefore, we can calculate $t_{\text{DMA}p}$ as shown in (13).

$$t_{\text{DMA}p} = l_d \left(\frac{n_g}{BW_g} + \frac{n_p}{BW_p} \right)$$

= $8 \left(\frac{5+3}{25.6 \times 10^9} + \frac{1+2}{25.6 \times 10^9} \right)$
= 3.43×10^{-9} s (13)

 n_{sat} can be figured as the n_{SPE} when $t_{\text{op}} = t_{\text{DMA}}$. Consequently, we obtain $n_{\text{sat}} = 3.70$, which suggests that the speedup is saturated when the number of SPEs is four or greater. Therefore, the number of SPEs in which the speedup is saturated is expected to be four, which corresponds with the experimentally obtained result presented in Sect. 3.2.

4. Conclusion

Implementation of the FDTD method on the Cell B.E. presents the problem that the execution time does not decrease in proportion to the number of SPEs because of a bottleneck of DMA transfers during large-scale analysis. To alleviate this problem, we employed a novel algorithm that is able to resolve the bottleneck. Using this algorithm, the electric field and magnetic fields which belong to different time steps are calculated simultaneously. This sequence of calculation reduces the DMA transfers to the main memory and solves the DMA transfer bottleneck. Results show that the FDTD method using the proposed algorithm has a linear decrease of execution time along with increase of SPEs. The proposed algorithm using six SPEs exhibits performance that is 1.8 times faster than that of conventional parallel computing with region segmentation. The performance is 1.7 times faster than that of a workstation using four cores.

References

- K.S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," IEEE Trans. Antennas Propag., vol.14, no.3, pp.302–307, May 1966.
- [2] C. Guiffaut and K. Mahdjoubi, "A parallel FDTD algorithm using the MPI library," IEEE Antennas Propag. Mag., vol.43, no.2, pp.94– 103, April 2001.
- [3] W. Walendziuk and J. Forene, "Verification of computations of a parallel FDTD algorithm," Proc. IEEE International Conference on Parallel Computing in Electrical Engineering (PARELEC'02), p.1, Warsaw, Poland, Jan. 2002.
- [4] P. Placidi, L. Verducci, G. Matrella, L. Roselli, and P. Ciampolini, "A custom VLSI architecture for the solution of FDTD equations," IEICE Trans. Electron., vol.E85-C, no.3, pp.572–577, March 2002.
- [5] J.P. Durbano, F.E. Ortiz, J.R. Humphrey, M.S. Mirotznik, and D.W. Prather, "Hardware implementation of a three-dimensional finite-difference time-domain algorithm," IEEE Antennas Wireless Propag. Lett., vol.2, pp.54–57, Jan. 2003.
- [6] T. Ciamulski, M. Hjelm, and M. Sypniewski, "Parallel FDTD calculation efficiency on computers with shared memory architecture," IEEE Workshop on Computational Electromagnetics in Time-Domain (CEM-TD 2007), pp.1–4, Oct. 2007.
- [7] B. Li, H. Jin, and Z. Shao, "Two-level parallel implementation of FDTD algorithm on CBE," Proc. IEEE International Conference on Networking, Sensing and Control (ICNSC'08), pp.1812–1817, Hainan, China, April 2008.
- [8] M. Xu and P. Thulasiraman, "Finite-difference time-domain on the Cell/B.E. processor," Proc. IEEE International Parallel and Distributed Processing Symposium (IPDPS'08), pp.1–8, Florida, USA, April 2008.
- [9] M. Xu and P. Thulasiraman, "Parallel algorithm design and performance evaluation of FDTD on 3 different architectures: Cluster, homogeneous multicore and Cell/B.E.," Proc. Tenth IEEE International Conference on High Performance Computing and Communications (HPCC'08), pp.174–181, Dalian, China, Sept. 2008.
- [10] M. Kister, M. Perrone, and F. Petrini, "Cell multiprocessor communication network: Built for speed," IEEE Micro, vol.26, no.3, pp.10–23, May/June 2006.



Sho Endo received a B.E. degree in Electronic Engineering from the National Institution for Academic Degrees and University Evaluation in 2010. He is currently working toward an M.S. degree. His research interests include computational electromagnetics, and parallel processing.



Takafumi Aokireceived B.E., M.E., andD.E. degrees in electronic engineering fromTohoku University, Sendai, Japan, in 1988,1990, and 1992, respectively. He is currently aProfessor of the Graduate School of InformationSciences at Tohoku University. For 1997–1999,he also joined the PRESTO project, Japan Science and Technology Corp. (JST). His researchinterests include theoretical aspects of compu-tation, VLSI computing structures for signaland image processing, multiple-valued logic,

and biomolecular computing. Dr. Aoki received the Outstanding Paper Award at the 1990, 2000, 2001 and 2006 IEEE International Symposia on Multiple-Valued Logic, the Outstanding Transactions Paper Award from the Institute of Electronics, Information and Communication Engineers (IEICE) of Japan in 1989 and 1997, the IEE Ambrose Fleming Premium Award in 1994, the IEICE Inose Award in 1997, the IEE Mountbatten Premium Award in 1999, the Best Paper Award at the 1999 IEEE International Symposium on Intelligent Signal Processing and Communication Systems, the IP Award at the Seventh LSI IP Design Award in 2005, and the Best Paper Award at the 14th Workshop on Synthesis and System Integration of Mixed Information Technologies.



Jun Sonoda received B.E. and M.E. degrees in Information Engineering from Toyohashi University of Technology, Toyohashi, Japan, in 1993 and 1995, respectively. He received a Ph.D. degrees in Environment Science from Tohoku University, Sendai, Japan, in 2005. He is currently an Associate Professor of the Department of Intelligent and Electronic Systems, Sendai National College of Technology. His current interests include high-performance and high-accuracy numerical simulation, large-scale

EM analysis, and electromagnetic/optical devices.



Motoyuki Sato received B.E., M.E., and Dr. Eng. degrees in Information Engineering from Tohoku University, Sendai, Japan, in 1980, 1982, and 1985, respectively. During 1988– 1989, he was a Visiting Researcher with the Federal German Institute for Geoscience and Natural Resources, Hannover, Germany. Since 1997, he has been a Professor at Tohoku University, where he has been a Distinguished Professor since 2007 and the Director of the Center for Northeast Asian Studies since 2009. He has

conducted development of ground penetrating radar (GPR) sensors for humanitarian demining. His sensor, the advanced landmine imaging system (ALIS), which is a hand-held dual sensor, has detected more than 30 mines in mine fields in Cambodia since May 2009. He is a Visiting Professor with Jilin University, Changchun, China; Delft University of Technology, Delft, The Netherlands; and Mongolian University of Science and Technology, Ulan Bator, Mongolia. His current interests include transient electromagnetics and antennas, radar polarimetry, GPR, borehole radar, electromagnetic induction sensing, interferometric, and polarimetric synthetic aperture radar. Dr. Sato is a member of the IEEE Geoscience and Remote Sensing Society (GRSS) Administrative Committee (2006, 2007), where he is responsible for specialty symposia and Asian issues. He is an Associate Editor of the IEEE GRSS Newsletter, and a Guest Editor of the Special Issue of GPR2006 in IEEE Transactions on Geoscience and Remote Sensing. He was the Chair of the IEEE GRSS Japan Chapter (2006, 2007). He was appointed as the General Chair of IGARSS2011, to be held in Sendai.