Journal of Multimedia Information System VOL. 4, NO. 4, December 2017(pp. 249-254): ISSN 2383-7632(Online)
http://dx.doi.org/10.9717/JMIS.2017.4.4.249

Vector Map Simplification Using Poyline Curvature

Ngoc-Giao Pham,[1] Suk-Hwan Lee,[2,*] and Ki-Ryong Kwon[1]

Abstract

Digital vector maps must be compressed effectively for transmission or storage in Web GIS (geographic information system) and mobile GIS applications. This paper presents a polyline compression method that consists of polyline feature-based hybrid simplification and second derivative-based data compression. Experimental results verify that our method has higher simplification and compression efficiency than conventional methods and produces good quality compressed maps.

Key Words: Vector map simplification, Polyline curvature, GIS Vector map, Vector map transmission, Vector map storage.

I. INTRODUCTION

The development of IT based GIS service and cartography technique enables the GIS vector map to be detailed and enriched by more detailed geographic information. However, since the detailed vector map needs huge amount of data, the problem is rearing how to process, store, and transmit them. Especially, web GIS or mobile GIS have the susceptibility for huge data because of limits of transmission bandwidth and mobile device for output, processing, and analysis of geographic information in real-time. Universal lossless compression of zlib and 7-zip has been widely used for the vector map compression based on entropy coding. However, they cannot do the random access for specific region or object in compressed vector map. In this case, they should recover all data of vector map for the random access. Furthermore, they have the limit of the compression efficiency because they do not consider the correlation of objects. Many researchers have worked for the effective vector map compression [1-6].

This paper addresses a combined method of simplification and lossless compression to reduce the data quantity while preserving the map quality and to cope with different GIS applications. Main features of our method are as follows. First, our method improves the compression efficiency with the shape preservation by poyline feature-based simplification and vector derivative-based compression. The former is a loss compression and the latter is a lossless compression. Second, polyline feature-based simplification enables precision and compression ratio to control or vary by a weighting factor. Third, vector derivative-based compression converts vector data to integer part and fractional part of second derivative vector in order to compact the spatial energy of vector data. The integer part that indicates the broad position of object is compressed by the correlation of neighbor vertices. The fractional part that indicates the close position of object is compressed progressively by varying the fraction precision.

Our experiment evaluated the performance of simplification and compression of our method by comparing with Park's simplification method [4] and Jang's compression method [5],[6], which are latest methods. From experimental results, we verified that the simplification rate of our method is about 4% higher than that of Park's method while the distance error and area error of our method is about 0.4[m] and 0.6 [m^2] less than those of Park's method. Furthermore, we verified that the

Manuscript received December 13, 2017; Revised December, 18, 2017; Accepted December 18, 2017. (ID No. JMIS-0050)

Corresponding Author (*): Suk-Hwan Lee, 428, Shinseonno, Namgu, Busan, Tongmyong Univ. E-mail.skylee@tu.ac.kr
[1]Department of IT Convergence and Application Eng., Pukyong National University, Busan, Korea, E-mail: krkwon@pknu.ac.kr
[2]Department of Information Security, Tongmyong University, Busan, Korea, E-mail: skylee@tu.ac.kr

compression ratio of our method is about 10% higher than that of Jang's method.

II. PROPOSED VECTOR MAP SIMPLIFICATION

We address a hybrid compression for vector map of polyline feature-based simplification and vector derivative-based data compression. Our method has two aims as follows; 1) reduce vertices while preserving the global shape of object and 2) enable to vary different precisions and control compression ratio. The process of our hybrid compression for vector map is shown in Figure 1. Polyline feature-based simplification detects feature points of polyline using DP, SF, and TF algorithms and applies an algorithm with minimum area error to sections. Vector derivative-based data compression converts all data of vertices in simplified polyline to integer part and fractional part of second vector derivative and compresses two parts respectively.

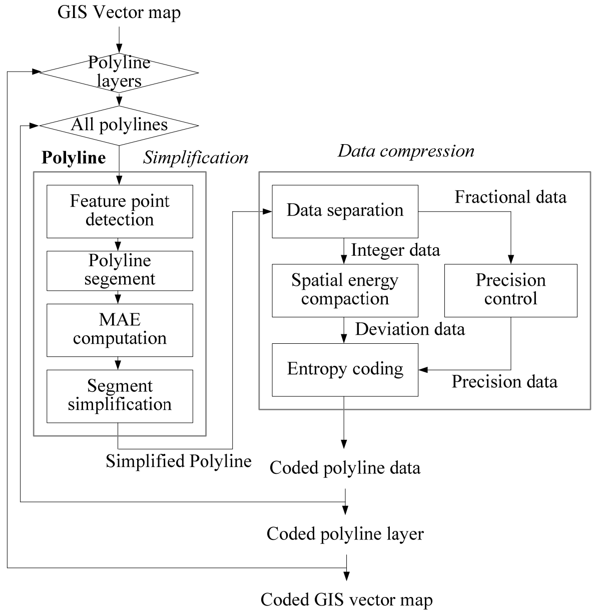


Fig. 1. Process of our hybrid compression for vector map.

2.1 Polyline Feature-Based Simplification

Our method performs on all polylines in a vector map and segment polylines by feature points that reflects the object shape. Then it simplifies the segments so that the area difference is minimized. The detail process of our simplification is as follows.

We extract feature points using DP, SF, and TF algorithms. Figure 2 shows parameters for three algorithms; the maximum perpendicular distance $d_{\max}$ for DP algorithm, the angle tolerance θ for sector bound for SF algorithm, and the length threshold l_{th} and orientation angle threshold $\Delta\theta_{th}$ of a segment line for TF algorithm. Given a polyine $\boldsymbol{L}$ in any layer, let us consider that $\boldsymbol{L}$ consists of N vertices, $\boldsymbol{L} = \{\boldsymbol{v}_0, \boldsymbol{v}_1, \cdots, \boldsymbol{v}_{N-1}\}$, and a vertex has two coordinate values $\boldsymbol{v}_i = \{x_i, y_i\}$. We generate three simplified polylines of a polyline $\boldsymbol{L}$ using DP, SF, TF algorithms.

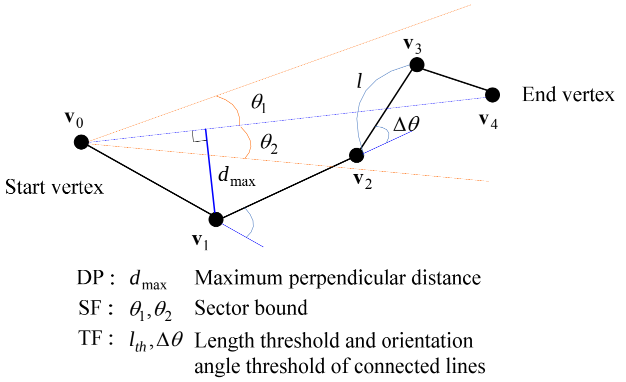

$\mathbf{v}_3$
l
θ_1
End vertex
$\mathbf{v}_0$
$\Delta\theta$
$\mathbf{v}_4$
θ_2
Start vertex
$d_{\max}$
$\mathbf{v}_2$
$\mathbf{v}_1$
DP : $d_{\max}$ Maximum perpendicular distance
SF : θ_1, θ_2 Sector bound
TF : $l_{th}, \Delta\theta$ Length threshold and orientation angle threshold of connected lines

Fig. 2. Parameters for DP, SF, and TF algorithms.

We define feature points $\boldsymbol{F}_L$ for a polyline $\boldsymbol{L}$ as vertices that exist in three simplified polylines $\boldsymbol{L}_{DP}$, $\boldsymbol{L}_{SF}$, $\boldsymbol{L}_{TF}$; $\boldsymbol{F}_L = \{\dot{\boldsymbol{v}}_k | k \in [1, N_F]\}$ for $\dot{\boldsymbol{v}}_k \neq \boldsymbol{v}_0, \boldsymbol{v}_{N-1}$ and $\dot{\boldsymbol{v}}_k \in \boldsymbol{L}_{DP}$ and $\dot{\boldsymbol{v}}_k \in \boldsymbol{L}_{SF}$ and $\dot{\boldsymbol{v}}_k \in \boldsymbol{L}_{TF}$. We use feature points as break points for segmenting polyline. Thus, a polyline $\boldsymbol{L}$ is segmented to N_F sections $\mathbf{S} = \{\boldsymbol{S}_k | k \in [0, N_F - 1]\}$ using N_F feature points. A section $\boldsymbol{S}_k$ consists of $k-1$th feature point $\dot{\boldsymbol{v}}_{k-1}$ and kth feature point $\dot{\boldsymbol{v}}_k$ and middle vertices between two feature points;

Our method simplifies a polyline $\boldsymbol{L}$ as a unit of segmented section $\boldsymbol{S}_k$ using DP, SF, and TF algorithms. Thus, we assign one among three algorithms to a section $\boldsymbol{S}_k$ using the minimum area difference (MAD) and then simplify a section $\boldsymbol{S}_k$ by an assigned algorithm. The minimum area difference (MAD) measures how different the simplified polyline is with the original polyline. We explain how to calculate MAD for a section in next sub-section.

We use the weighting factor α for controlling the simplification intensity and for generating vector maps with different precisions. Thus, we set parameters for three algorithms by the weighting factor and initial parameters.

$$\delta = \delta_0 \times (1 + \alpha) \text{ for DP} \tag{1}$$

$$\varepsilon = \varepsilon_0 \times (1 + \alpha) \text{ for SF} \tag{2}$$

$$l_{th} = l_{th,0} \times (1 + \alpha) \text{ and } \Delta\theta = \Delta\theta_0 \times (1 + \alpha) \text{ for TF} \tag{3}$$

where $\alpha \geq -1$

Let us look at how to compute MAD for a polyline, which is used for simplifying vertices while preserving the shape of original polyline. A section $\boldsymbol{S}_k =$

$\{\boldsymbol{v}_l, \boldsymbol{v}_{l+1}, \cdots, \boldsymbol{v}_{l+N_k}\}$ likes a sub-polyline with the start vertex and end vertex of feature points. The process for computing MAD, MAD_k, of a section is as follows.

1. Consider $\boldsymbol{S}_k$ as sub-polygon type by connecting the start vertex to the end vertex.
2. Compute area A_k of sections $\boldsymbol{S}_k$ with N_k vertices.

$$A_k = \frac{1}{2}\sum_{i=0}^{N_k}(x_{k+i}y_{k+i+1} - x_{k+i+1}y_{k+i}) \tag{4}$$

And compute areas A_k^{DP}, A_k^{SF}, A_k^{TF} of sections that correspond to simplified polylines $\boldsymbol{L}_{DP}$, $\boldsymbol{L}_{SF}$, $\boldsymbol{L}_{TF}$ using the above equation. Since $\boldsymbol{S}_k$ uses two feature points as endpoints, simplified polylines $\boldsymbol{L}_{DP}$, $\boldsymbol{L}_{SF}$, $\boldsymbol{L}_{TF}$ are segmented to the same sections to original polyline. However, these sections include different vertices and different number of vertices.

3. Compute the differences between an area A_k and three areas A_k^{DP}, A_k^{SF}, A_k^{TF}. Denote these differences as e_k^{DP}, e_k^{SF}, e_k^{TF}.

$$e_k^{DP} = A_k - A_k^{DP}, \quad e_k^{SF} = A_k - A_k^{SF}, \quad e_k^{TF} = A_k - A_k^{TF} \tag{5}$$

for all $k \in [1, N_k]$

Since $\mathrm{A_k}$ for an original polyline is always higher than $\mathrm{A_k^{DP}}$, $\mathrm{A_k^{SF}}$, $\mathrm{A_k^{TF}}$ for simplified polylines, the differences $\mathrm{e_k^{DP}}$, $\mathrm{e_k^{SF}}$, $\mathrm{e_k^{TF}}$ are over zero. Define MAD_k for $\boldsymbol{S}_k$ as the minimum among e_k^{DP},e_k^{SF},e_k^{TF}.

$$MAD_k = \min(e_k^{DP}, e_k^{SF}, e_k^{TF}) \tag{6}$$

δ_0 and ε_0 are initial distance threshold for DP and SF and $l_{th,0}$ and $\Delta\theta_0$ are initial length threshold and initial angle threshold for TF. In this paper, we set δ_0, ε_0, $l_{th,0}$ to 1[m] and set $\Delta\theta_0$ to 5°.

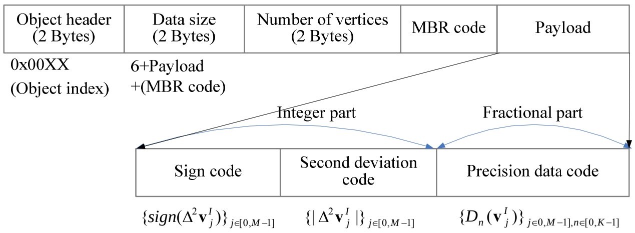
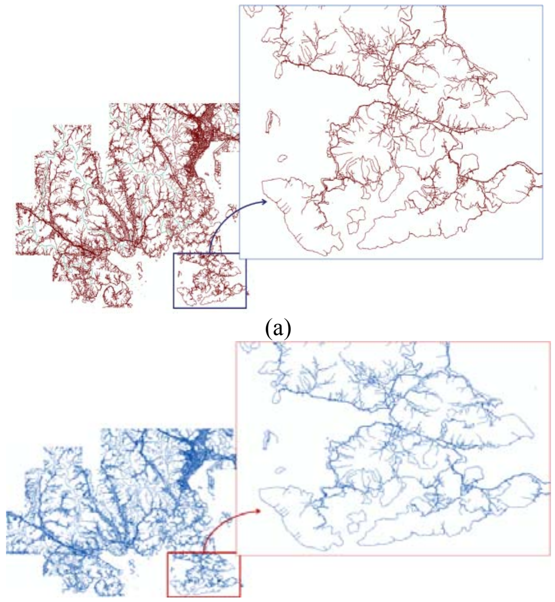


Fig. 3. Structure of coded polyline data.

The weighting factor α can control the compression ratio and the number of simplified vertices. If $\alpha = -1$, a polyline do not be simplified. If α increases from -1, the simplified vertices increase while the data capacity decrease. However, the increased α produces the degradation of shape.



(a)

(b)

Fig. 4. Structure of coded polyline data.

2.2 Vector Derivative-based Data Compression

We separate the data of vertices in a simplified polyline $\boldsymbol{L}'$ to two parts; integer parts $\boldsymbol{V}^I$ and fractional parts $\boldsymbol{V}^R$. Fractional parts are converted to integer types by the reciprocal number of the precision c that indicates the precision level of map data. Thus, two data of integer part and fractional part of a vertex $\boldsymbol{v}_j'$ can be defined by

$$\boldsymbol{v}_j^I = \lfloor \boldsymbol{v}_j' \rfloor \text{ and } \boldsymbol{v}_j^R = (\boldsymbol{v}_j' - \boldsymbol{v}_j^I) \times 10^{-c} \tag{7}$$

for all $j \in [0, M-1]$.

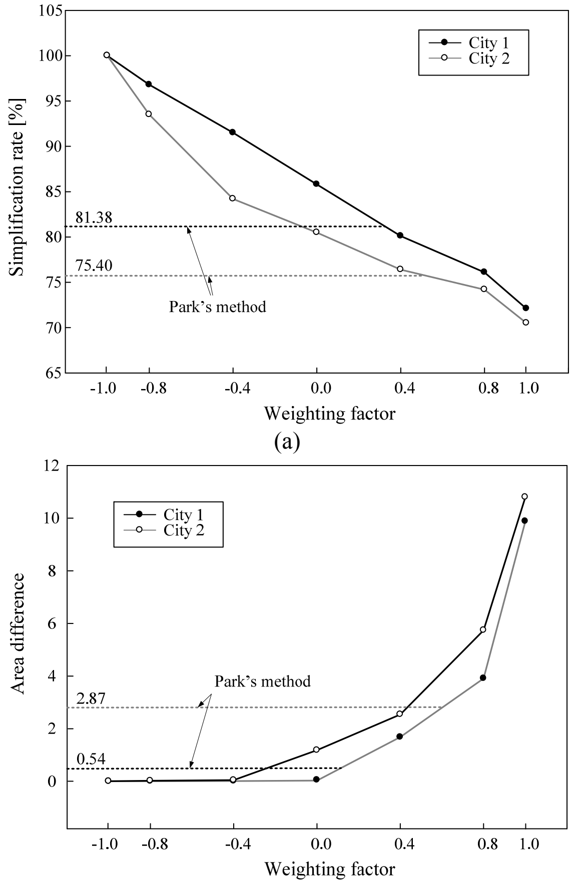
The integer part of vertex data represents the primary position of vector map. Therefore, our method compresses $\boldsymbol{V}^I$ with lossless using the bounding box of object MBR. We firstly compute the first-order derivative vector of integer part on the left-top point MBR_{min} of MBR

$$\Delta\boldsymbol{v}_j^I = \boldsymbol{v}_j^I - MBR_{min} \text{ for all } j \in [0, M-1] \tag{8}$$

and the average derivative vector $\overline{\Delta\boldsymbol{v}^I}$. Then we compute the second-order derivative vector $\Delta^2\boldsymbol{v}_j^I$ as follows.

$$\Delta^2\boldsymbol{v}_j^I = \Delta\boldsymbol{v}_j^I - \overline{\Delta\boldsymbol{v}^I} \text{ for all } j \in [0, M-1] \tag{9}$$

All integer parts $\boldsymbol{V}^I$ are converted to the magnitudes and signs of the second-order derivative vectors $\Delta^2\boldsymbol{v}_j^I$; $\{\mathrm{sign}(\Delta^2\boldsymbol{v}_j^I), |\Delta^2\boldsymbol{v}_j^I|\}$.



(a)

(b)

Fig. 5. Structure of coded polyline data.

The fractional part represents the detail precision of vector map. This part is sensitive to small deviation unlike integer part and is little correlated to each other. We separates fractional parts to precision data as a unit of 8-bit and rearranges precision data by precision levels. Example, the data of nth. precision level for $\boldsymbol{v}_j^R$ can be defined as

$$D_n(\boldsymbol{v}_j^R) = (\boldsymbol{v}_j^R \gg 8n)\&0xFF. \tag{10}$$

This compression enables the vertex data to be transmitted or stored to different precision levels or preferred precision levels.

From the above processes, the integer part $\boldsymbol{V}^I$ and fractional part $\boldsymbol{V}^R$ for a polyline $\boldsymbol{L}$ can be converted to data for compressing. The coded data for a polyline has the structure of a polyline label (object index), a size of coded data, a number of vertices, a MBR coordinate value, and a payload, as shown in Figure 3. A MBR is for easy access to a polyline. A payload is the coded data of integer part and fractional part. Since the compression and recovery are performed as the unit of a polyline object, our method makes the random access of polyline objects easy.

III. Experimental Results

Our experiment used shapefiles (SHP format) of two cities with different layers and applied our method to layers with polyline object in shapefiles. A shapefile format that is a popular geospatial vector data format.

3.1 Map Quality Evaluation

Our experiment evaluated the subject quality between the original map and compressed map. Figures 4(a),4(b) show full layers of original maps and compressed maps. From these figure, we know that though about 29% and 20% vertices are removed in compressed vector maps, there are no visually difference between original vector map and compressed vector map. From these figures, we know that it is very difficult to discriminate visually between original layer and compressed layer. We investigated the degradation of quality in enlarged maps. The quantitative criteria for evaluating quality in the enlargement has not been provided yet. Therefore, we enlarged two of original map and compressed map to arbitrary scales and looked the discrimination of two enlarged maps. From these results, we know that the degradation of compressed map comes into sight from about 2,356x. Thus, it should be enlarged our compressed map to large scale to see the degradation of quality.

3.2 Simplification Evaluation

Our method in $\alpha = 0$ has the same parameters of three simplification algorithms as Park's method. We compared our method in $\alpha = 0$ to Park's method. The simplification rates for layers of our method are lower 0.1%-10.08% than those of Park's method. Thus, our method simplifies vertices of layers more higher 0.1%-10.08% than Park's method. Moreover, our method simplifies vertices in a map higher 4% than Park's method. Layers of facility and administrative boundary are simplified to about 94%-99% because there are a number of simple polylines. However, layers of river and road are simplified to about 63%-74% because there are a number of complex polylines.

We measured simplification rate r and mean area difference $\bar{A}$ of our method on different weighting factors from -1.0 to 1.0. The measurement results are shown in Figures 5(a) and 5(b). As α increases, r decreases while $\bar{A}$ increases. When $\alpha = 1$, r decreases about 57%-65% while $\bar{A}$ increases about 9.88[m^2]-12.14[m^2]. However, when $\alpha = -0.4$, r increases about 92%-94% while $\bar{A}$ increases about 0.25[m^2]-0.40[m^2]. From these results, we verified that our method can control the simplification rate and mean area difference varying the weighting factor. This enables users to receive simplified vector maps on different GIS applications.

3.3 Compression Evaluation

We compared the compression efficiency of our method to that of Jang's SEC method. Hereby, we experimented two ways of our method for comparing with Jang's method. The first way is to do only the process of lossless compression that preserves data of all precision levels in fractional part. This way passes over the simplification. The second way is to do the simplification with $\alpha = 1$ and lossless compression. The compression efficiency of our lossless compression has slightly higher 0.43%-3.60% than that of Jang's method. Furthermore, the compression efficiency of our simplification and lossless compression has higher about 3.89%-16.58% than that of Jang's method and higher about 3.47%-13.84% than that of only lossless compression. When looking at the results on layers, layers of river and tributary that has a number of polylines are compressed to high rate. However, layer of facility that has a few of polylines is compressed to low rate. It is natural that the polyline based compression method depends on the number of polylines on layers.

III. CONCLUSION

We have addressed a polyline simplification and compression for the effective transmission or storage of vector maps for various GIS applications. The proposed method consists of two processes of polyline feature point-based simplification and the second derivative-based data compression. The first process for simplification detects the feature points of polyline using three algorithms of DP, SF, and TF and divides a polyline to sectors by feature points. The next process for data compression segments the second derivative of simplified polylines to integer parts and fractional parts. Experimental results showed that our method has good subjective quality and also has more simplification efficiency than Park's simplification method and has more compression efficiency than Jang's compression method. Furthermore, our method can provide the closest quality to original map with about 23% vertices of original map and generate vector maps with different precision levels or compression ratios.

Acknowledgement

This work was supported by BB21(Brain Busan) project and also Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2016R1D1A3B03931003 and NRF-2017R1A2B2012456).

References

[1] Z. Xie, H. Wang, and L. Wu, “The improved Douglas-Peucker algorithm based on the contour character,” *Geoinformatics, 19th International Conference on*, pp. 1-5, 2011.

[2] Z. Zhao and A. Saalfeld, “Linear-time sleeve-fitting polyline simplification algorithms,” Autocarto 13, ACSM/ASPRS’97 Technical Papers, Seattle, Washington, vol. 5, pp. 214–223, 1997

[3] J. D. Carvalho, D. Guliato, D. A. Santiago and R. M. Rangayyan, “Polygonal modeling of contours using the turning angle function,” *Proceedings of Canadian Conference on Electrical and Computer Engineering*, pp. 1090-1093, 2007.

[4] W. J. Park and K. Y. Yu, “Hybrid line simplification for cartographic generalization,” *Pattern Recognition Letters*, vol. 32, no 9, pp. 1267-1273, 2011.

[5] B.-J. Jang, G.-S. Moon, S.-H. Lee, and K.-R. Kwon, “Effective Compression Technique for Secure Transmission and Storage of GIS Digital Map,” *Journal of Korea Multimedia Society*, vol. 14, no. 2, pp. 210-218, 2011.

[6] B.-J. Jang, S.-H. Lee, and K.-R. Kwon. “GIS Vector Map Compression using Spatial Energy Compaction based on Bin Classification,” *Journal of IEEK*, vol. 49, no. 3 pp. 15-26, 2012.

Authors

Ngoc-GIao Pham received B.E degree in School of Electronic & Telecommunication in Hanoi University of Science & Technology (HUST) in 2011, and Master degree from Pukyong National University (PKNU), Busan, South Korea in 2014. Currently, he is Ph.D. student in Multimedia Communication & Signal Processing Lab in PKNU. His research interests include digital image processing application, GIS visualization, multimedia data security and smart system.

Suk-Hwan Lee received a B.S., a M.S., and a Ph.D. Degrees in Electrical Engineering from Kyungpook National University, Korea in 1999, 2001, and 2004 respectively. He is currently an associate professor in Department of Information Security at Tongmyong University. His research interests include multimedia security, digital image processing, and computer graphics.

Ki-Ryong Kwon received the B.S., M.S., and Ph.D. degrees in electronics engineering from Kyungpook National University in 1986, 1990, and 1994 respectively. He worked at Hyundai Motor Company from 1986–1988 and at Pusan University of Foreign Language from 1996–2006. He is currently a professor in Dept. of IT Convergence & Application Engineering at the Pukyong National University. He has researched University of Minnesota in USA on 2000–2002 with Post-Doc, and Colorado State University on 2011–2012 with visiting professor. He is currently the General President of Korea Multimedia Society, also is a director of IEEE R10 Changwon section. His research interests are in the area of digital image processing, multimedia security and watermarking, bioinformatics, weather radar information processing.