Selective Encryption Algorithm Using Hybrid Transform for GIS Vector Map
Bang Nguyen Van*, Suk-Hwan Lee**, and Ki-Ryong Kwon**

Abstract
Nowadays, geographic information system (GIS) is developed and implemented in many areas. A huge volume of vector map data has been accessed unlawfully by hackers, pirates, or unauthorized users. For this reason, we need the methods that help to protect GIS data for storage, multimedia applications, and transmission. In our paper, a selective encryption method is presented based on vertex randomization and hybrid transform in the GIS vector map. In the proposed algorithm, polylines and polygons are focused as the targets for encryption. Objects are classified in each layer, and all coordinates of the significant objects are encrypted by the key sets generated by using chaotic map before changing them in DWT, DFT domain. Experimental results verify the high efficiency visualization by low complexity, high security performance by random processes.

Keywords
Chaotic Map, GIS Vector Map, Selective Encryption, Vector Map Security

1. Introduction

A geographical information system (GIS) [1-3] is a large system invented to manipulate, analyze, store, capture, and control the geographic. In a GIS, vectors often express geographical features and illustrate features as geometrical shapes. Many varying phenomena can be represented by vector data. Vector map manages all kinds of the geographic information data as geometric factor, topology and metadata by vector data. Vector data reflect features of the real object in the GIS environment, we only use a small size for storing data; graphic represent the spatial data that quite similar handed map; projection is made easily and coordinates transformation [4-6]. For these reasons, vector map is applied in many areas and we also use it in GIS application that help users easily access to services through high technology devices or internet.

However, the detailed vector maps require huge amounts of data, processing, storing, transmitting them poses a challenge and we need human resources, servers, etc., to save a digital map. But, any company can use it, create illegal copies of the map and sell or distribute them easily many times without taking any permission from the original GIS data provider. Besides, GIS data is used in military
fields that require the high level security and must be protect away from attackers, also unauthorized
users. So vector map is necessary to be protected and prevent illegal duplication and distribution of it.

In vector map security, conventional approaches encrypt whole data, so the cryptography of data files
and profiles increase complexity, and these methods take a long time. Furthermore, the decryption data
often occurs loss data and it takes a long time to processing time, because authors use the complicated
mathematical models on big digital data. It is not also flexible for various data types.

For meeting above requirements, a selective encryption scheme is proposed for GIS map security for
multimedia applications, storage and transmission. In proposed algorithm, vector map is separated to
select polyline/polygon layer. We select and encrypt the significant objects in each layer by key sets
generated by using Chaotic map before changing them in DWT, DFT domain. In DWT domain, we
select some coefficients and encrypt them by changing first coefficient in DFT domain. Summary, our
proposed method focus on polylines/polygon objects as target for selective encryption, and change
values of vertices of polylines/polygons by key sets before transforming them by using hybrid
transform. Main advantages of the proposed algorithm are simple computing and transforming
processes but it still meets requirements of security by random processes, and it can be applied to many
types of vector map data. Experimental results show the high efficiency visualization by low complexity,
high security performance by random processes and cryptography. In addition, experiments also show
unique performance, decrypting error approximate zero and computation time be very short.

The remaining parts of paper are organized as follows: Section 2 gives a brief description about vector
map security watermarking, full encryption and selective encryption. Section 3 explains the proposed
algorithm in the detail. And we show experimental results and analytics. Finally, Section 5 gives a
conclusion of the paper.

2. Related Works

Until now, we can separate GIS data into two categories: vector and raster forms. Vector data is a
collection of layers in which the layer includes many geographical objects. These objects reflect
geographical features and topography of real objects or location. A layer includes vector data which
represent the world using points, polygons, and lines (polylines) [7,8], as shown Fig. 1.

According to the recent growth of network digital media, data are needed to protect from distribution
and illegal copying. Many approaches are researched for this issues; these include authentication,
encryption, and time stamping.

Digital watermarking has been researched to solve the issues in vector map since 2000s. The
traditional method embed secret information in some locations of vector map: [9-11] proposes certain
rules, they help to select a set of coordinates of vertices, and editing them by using a certain range of
precision; [12,13] modify coefficients in the frequency domain to complete watermarking hiding that is
one of the important solution of watermarking algorithm, but the disturb to the vector map content is
also existing, the resistance performance to data fitting, interpolation, scaling is poor. In summary, the
existing methods of geospatial watermarking can be distributed as follows: algorithm uses DCT, DWT,
and DFT domain, the algorithms in spatial domain, and algorithms inherited in 3D watermarking. Not
similar with general multimedia data types, watermarking in vector map has its distinct features due to
the application environments and special data structures of vector data. However, watermarking isn’t a
Selective Encryption Algorithm Using Hybrid Transform for GIS Vector Map

A foolproof way to protect them, it only identifies the rightful owner of the work, so it is the final step in security policy.

The full encryption algorithms usually encrypt all components of original data to change whole data. Wu et al. [14] considered characteristics of the storage, parameters and initial values of chaotic map. After that, he proposed a new compound encryption method, this process is not available to any type formats of data and object indexing. Li [15] selected the vector dataset in external Oracle DBMS for encryption, and he used standard cryptography DES combining with an R-tree spatial index. This algorithm encrypts the spatial index when the GIS dataset is transmitted to the client and designs the key management of public and private keys on a PKI system. In this process, the key length is short so it cannot keep data on the DBMS with high security. Dakroury et al. [16] described better the encrypting algorithm, AES and RSA cryptography are combined along with watermarking method that used in internet online service. This algorithm encrypts all parts of a shape-file using 256 bit for private key of a block cipher AES. Jang et al. [17] encrypted perceptually all polylines and polygons by lossless minimum coding object (MCO) units. But, these algorithms use whole shape-files for encryption and they do not consider important features, it is taking a long time to handle.

![Fig. 1. An example GIS vector map. (a) Area layer, (b) road layer, (c) water layer, and (d) area + road + water layers.](image-url)
In digital data protection topic, selective encryption methods have been researched since 2000s. The simple concept of selective encryption is showed in Fig. 2. Nowadays, many selective encryption algorithms are proposed to video and image. However, all algorithms used DES, AES or other symmetric-key algorithms become quite simple algebraic structure and not high security. Therefore, our proposed method only select polyline and polygon objects in GIS map for encryption. They are considered as protected part and user need authority to access this part. Firstly, we randomize these objects by randomization values generated from SHA-512 and chaotic map. After that, we continued to encrypt them in DWT, DFT domain. The step by step of our method is explained in detail in Section 3.

3. Proposed Algorithm

3.1 Encryption Process

Fig. 3 shows the illustration diagram of our proposed algorithm, and we also explain the step-by-step in detail, as follows:

![Diagram](image)

**Fig. 3.** Illustration diagram for proposed encryption technique.
• A vector map $M$ is a set of layers: $M = \{L_i | i \in [1, |M|]\}$ with $|M|$; the cardinality in map $M$ (the cardinality of a set is a measure of the "number of elements of the set").
• A layer $L_i$ include many polyline/polygon object: $L_i = \{O_{ij}|j \in [1, |L_i|]\}$ with $|L_i|$ is the total number of objects in layer $L_i$.
• An object $O_{ij}$ is a set of vertices $O_{ij} = \{v_{ijk}|k \in [1, |O_{ij}|]\}$ and $O_{ij}$ has two attributes: the total number of points and the area of the bounding box. We find the area threshold $(B_{L,th})$ and the point threshold $(A_{L,th})$ in layer $L_i$.
• All objects are classified into two groups by comparing their attributes with threshold values.
• The proposed method leave the insignificant objects without encryption.
• With the significant objects, all vertices of these objects are encrypted by key sets generating from hybrid transform and chaotic map.

The detail of each step is illustrated in continuous sections.

3.1.1 Object definition

In the proposed method, polyline and polygon objects are selected for encryption. Nevertheless, we have many objects in a layer, so if all of them are encrypted, it also take a long processing time. An object has two characteristics such as the area of the bounding box $(A)$ and the number of points $(P)$, as shown in Fig. 4(a). These characteristics are different when comparing each other. Some objects are created by a few vertices and the value of area is very small. These objects are very simple and they are marked as the insignificant objects, as shown in Fig. 4(b). In the same way, we have some objects created from many vertices and the area of the bounding box is larger than, it also complex than and mark it as a significant object, as shown in Fig. 4(c).

For this reason, the probability distribution is used to define thresholds in one layer. By this way, we classify an insignificant object or a significant object by comparing object’s attributes with thresholds. In a layer $L_i$, two threshold values: $A_{L,th}$ and $B_{L,th}$ are calculated as following:
• A layer $L_i : L_i = \{O_{ij}|j \in [1, |L_i|]\}$ with $|L_i|$ is the cardinality of a layer $L_i$.
• An object $O_{ij}$ has two attributes: The total number of points $(B_{ij})$ and the area of bounding box $(A_{ij})$.

Fig. 4. Example about objects in a layer.
• So, in layer $L_i$ we have: $A_i = \{A_{ij} | j \in [1, |A_i|]\}$, $B_i = \{B_{ij} | j \in [1, |B_i|]\}$ and: $|A_i| = |B_i| = |L_i|$. 

• The discrete probability distribution is used to find threshold in set $A_i, B_i$ with thresholds are the smallest values as possible by condition (1), (2):

$$A_{i,th} \in A_i \text{ and } F(A) = \sum P(X = A_{ij} & A_{ij} \geq A_{i,th}) \geq 0.5 \tag{1}$$

$$B_{i,th} \in B_i \text{ and } F(B) = \sum P(X = B_{ij} & B_{ij} \geq B_{i,th}) \geq 0.5 \tag{2}$$

With $X$ is called a discrete random variable and $\sum_{j=1}^{|A_i|} P(X = A_{ij}) = \sum_{j=1}^{|B_i|} P(X = B_{ij}) = 1$. And if an object have $A_{ij} \geq A_{i,th}$ and $B_{ij} \geq B_{i,th}$, mark it as a significant object, else mark it as an insignificant object.

3.1.2 Chaotic map

The logistic map is the classic example of a non-linear recursion relation whose iterative values (trajectory) can exhibit deterministic chaos. The classical chaos system in one-dimension is a logistic map, which can be defined by following:

$$x_{k+1} = \mu x_k (1 - x_k) \tag{3}$$

$\mu$ is a positive constant sometimes known as the "biotic potential" gives the so-called logistic map $0 \leq \mu \leq 4$, $k=0,1,2,\cdots$, and all the values of $\{x_i\}$ appear in the range $[0,1]$ for the initial value $x_0 \in [0,1]$. It is noted that Eq. (3) has the chaotic behavior [27,28] when $\mu$ appears in the range $[3.57,4]$, and especially the chaotic behavior called Pomeau–Manneville scenario [18] when $\mu$ appears in the range $[3.57,3.82]$.

![Fig. 5. Proposed encryption for vector map.](image)

![Fig. 6. Selective encryption process in polyline/polygon layer.](image)
### 3.1.3 Encryption block

After all objects are classified, the significant objects are encrypted, as illustrated in Fig. 5 and shown in Fig. 6.

- **Step 1:** X, Y coordinates of the significant objects are selected and arranged into two 1D-arrays and the length of one array is $N$. Define length of the segment is $n$, with $n = 2^k$ and $k$ is determined by using the floor function ($\lfloor x \rfloor$ is the largest integer less than or equal to $x$) and $N_s$, as follows:

$$k = \lfloor \log_2 \sqrt{N_s} \rfloor$$

The number of segments is determined by equation:

$$m = \left\lfloor \frac{N_s}{n} \right\rfloor$$

- **Step 2:** The user key $K_1$ is used to create two key sets for layer $L_i$. It is generated randomly the first key in each key set by SHA-512 algorithm from user key [19] with key length is 512 bits for each key. And other keys are generated by using chaotic map as Eq. (3). Fig. 7 shows a key process for the key sets generation and DFT encryption value. And, we have two key set: $a = \{a_i | i \in [1, n]\}, b = \{b_i | i \in [1, n]\}$ and DFT encryption value: $E_{DFT} = n * a_i * b_i$.

![Fig. 7. Shows a key process for key sets generation and DFT coefficient encryption.](image)

- **Step 3:** All coordinates in one segment are multiplied with values of two key sets $a, b$ and we make the complex numbers as equation: $Z_i = X_i * a_i + j Y_i * b_i, i \in [1, n]$.
- **Step 4:** This segment is applied DWT-$k$ level to get a set of DWT coefficients. Select all first coefficients of DWT coefficients and continue to apply DFT to get a set of DFT coefficients.
- **Step 5:** In DFT domain, the first coefficient is multiplied with DFT encryption value by Eq. (6).

$$DFT^* = DFT * E_{DFT}$$

Then, we perform IDFT to get a set of encrypted values of second transformed values.
- **Step 6:** All first coefficients in DWT-segments are replaced by encrypted values in step 5 and IDWT-$k$ level to get a set of encrypted values of first transformed values. And, the real and image parts of the encrypted complex values also are the encrypted coordinates of objects.
3.2 Decryption Process

To perform decryption, after threshold values are defined, the decryption block is used to decrypt the selected vertices of the significant objects. If the user has the correct key when he decrypts the encrypted map, and the decoded data should be like a transcript of the input map. The output map is very various if the user provides an incorrect key. Results will verify the high efficiency visualization by low complexity, high security performance in the next section. In addition, experiments also show a unique performance, decryption error approximate zero and computation time faster than the existing algorithms.

4. Experimental Results

4.1 Object Definition

In our algorithm, the significant objects and the insignificant objects are defined based on threshold values in layers. After that, we only select the significant objects for encryption. When we change two values (that are user-defined) in Eqs. (1)–(2), threshold values are also changed and percentage of the encrypted vertices change according to this values, as shown in Fig. 8.

Fig. 8. Percentages of the significant objects (a) and the encrypted points (b).
4.2 Visualization

For performance evaluation of the proposed algorithm, many different scaled maps are used that contain layers, including rivers, roads, lakes, and countries in the world. Simulation results prove that the encrypted map change absolutely perception of whole maps, as shown in Figs. 9–11. The proposed method changes the whole visual image of map because randomly some vertices are encrypted lead to scrambling shape of all objects. In addition, the proposed method changes only vertices position of objects in the layer. The size of input and output map are the same, so it is not a lose data.

![Images of maps](image1)

**Fig. 9.** Experimental results with scaling layers 1:5,000. (a) Original polyline layer, (b) encrypted polyline layer, (c) original polygon layer, and (d) encrypted polygon layer.

![Images of maps](image2)

**Fig. 10.** Experimental results with scaling layers 1:10,000, 1:100,000. (a) Original polyline layer, (b) encrypted polyline layer, (c) original polygon layer, and (d) encrypted polygon layer.
Fig. 11. Experimental result with full scaling layers 1:5,000. (a) Original map and (b) encrypted map.

4.3 Decryption Error

We used 1:5,000, 1:10,000 and 1:100,000 scaling maps in our experiments. Firstly, we experimented with each polyline layer and polygon layer. Then, we tested the full map of 1:5,000 scale. Experimental results show that all maps are changed as given by Figs. 9–11. Unauthorized users cannot see anything on the map because we changed polyline/polygon through many processes: multiplication, DWT, IDWT, DFT and IDFT.

Table 1. The error between original coordinates and decrypted coordinates

<table>
<thead>
<tr>
<th>Original coordinates</th>
<th>Decryption coordinates</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>144.13246</td>
<td>144.132460000128</td>
<td>1.27982957565109E-10</td>
</tr>
<tr>
<td>-37.32808</td>
<td>-37.3280800000347</td>
<td>3.47029072145233E-11</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>-118.749289</td>
<td>-118.749289001048</td>
<td>2.83009171653248E-10</td>
</tr>
<tr>
<td>34.817734</td>
<td>34.8177340000613</td>
<td>6.12985218140238E-11</td>
</tr>
</tbody>
</table>

Table 2. The max, min error between original map and decryption map

<table>
<thead>
<tr>
<th>Size (kB)</th>
<th>Total object</th>
<th>Total point</th>
<th>Max error</th>
<th>Min error</th>
<th>Average error</th>
</tr>
</thead>
<tbody>
<tr>
<td>332</td>
<td>76</td>
<td>20920</td>
<td>4.58763E-07</td>
<td>0</td>
<td>1.75211E-08</td>
</tr>
<tr>
<td>449</td>
<td>147</td>
<td>28162</td>
<td>3.70411E-07</td>
<td>0</td>
<td>1.72524E-08</td>
</tr>
<tr>
<td>751</td>
<td>20</td>
<td>47947</td>
<td>2.88627E-07</td>
<td>0</td>
<td>4.00043E-08</td>
</tr>
<tr>
<td>965</td>
<td>7011</td>
<td>37209</td>
<td>1.13562E-07</td>
<td>0</td>
<td>2.92078E-10</td>
</tr>
<tr>
<td>1246</td>
<td>375</td>
<td>79499</td>
<td>6.41549E-07</td>
<td>0</td>
<td>6.00142E-08</td>
</tr>
<tr>
<td>1730</td>
<td>13960</td>
<td>61798</td>
<td>7.83001E-08</td>
<td>0</td>
<td>1.76301E-09</td>
</tr>
</tbody>
</table>

In this algorithm, only coordinates of points (vertices) in objects are changed. The encrypted map still have the same size as the original map. However, Chaotic map is used to generate random number and key values from user’s key hashing SHA-512 bits, it make these values not absolutely similar in encryption and decryption step. Meaning of this issues come from the problem of system calculation.
when it stored real numbers in memory. With storing vertices in double type, it seems be no problem when the decryption errors values are approximately zero, as given by Table 1. Then, many maps are tested to find the maximum error and calculate the average error, as shown in Table 2.

4.4 Distance Measure

We use Eq. (7) to calculate the difference between encrypted map and original map:

\[ D(E', L) = \sum_{i=1}^{N} d(P_{ij}) \]  

where \( L \) is an original map and \( E'(L) \) is corresponding encrypted map and \( N \) is total object in original map. \( d(P_{ij}) \) is distance between corresponding objects in \( E'(L) \) and \( L \), which is computed by

\[ d(P_{ij}) = \sum_{j=1}^{N_{ij}} \sqrt{(x'_j - x_j)^2 + (y'_j - y_j)^2} \]  

where \( N_{ij} \) is the total number of points in object \( P_{ij} \).

We used polyline map, polygon map to experiment with different passwords \( K_1 \) and \( K_2 \). Then we calculate \( D(E', L) \) distance of each experimental time, as show in Table 3.

<table>
<thead>
<tr>
<th>Total number of points</th>
<th>Distance</th>
<th>User key ( K_1 )</th>
<th>User key ( K_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>798</td>
<td>35,682</td>
<td>39,065</td>
<td></td>
</tr>
<tr>
<td>1249</td>
<td>53,018</td>
<td>50,682</td>
<td></td>
</tr>
<tr>
<td>2457</td>
<td>200,133</td>
<td>178,431</td>
<td></td>
</tr>
<tr>
<td>2967</td>
<td>233,311</td>
<td>200,644</td>
<td></td>
</tr>
<tr>
<td>3900</td>
<td>318,270</td>
<td>404,741</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 12. Computation time according to the total number of objects.
4.5 Computation Time

The computation time of proposed methods depends on many factors. Moreover, it also depends on random processes, size of the map. Therefore, to measure the computation time we need to use specific mathematical model. In this section we just show the dependence of computation time and we compared result with Giao’s algorithm [20,21] under same the total number of objects, as shown in Fig. 12. Analyzing the detail results, we verify that the computation time of our method have lower than those of Giao’s method.

4.6 Security

**Cryptographic security:** Our proposed method use threshold values to select the significant objects, this method encrypted 60%-70% of data with the key values and random coefficients created by using Chaotic map and secret key.

**Key sensitivity analysis:** A highly key sensitive encryption algorithm protects the encrypted data against various cryptanalytic attacks. While developing a cryptosystem, it is assumed that an intruder knows the encryption structure and a-priori probability of used key \( k \in K \). As per the Kerckhoff’s principle [22], only secrecy of the used key is required. Even a strong or well-designed cryptosystem can be broken easily if the key is poorly chosen or key space is too small. Thus, a good cryptosystem should satisfy the following two conditions to verify the key sensitivity and key space:

- The key space should be discretized in such a way that two ciphertexts encrypted by two slightly different keys \( k_1, k_2 \in K \) should be completely different.
- With the generated keys, the ciphertext should be responding to slight changes, signals, or influences.

The original layer is encrypted by using the slightly different keys, and we analysis the difference between the obtained encrypted layers. The different layers are evaluated to verify the condition that, “layer is encrypted with slightly different keys should be completely different”.

For security evaluation, the slightly different keys are generated by modifying the first key in each key set \( a, b \) and modifying coefficient \( \mu \) in Eq. (3). And then, the method change one of them when keeping other values in the modified key. So, when test the key sensitivity, algorithm use the original key with three components. For the original key \( K_1: (3.52,0.34,0.62) \) (three parameters \( (\mu, a_1, b_1) \) represent for key \( K \)), the modified keys are expressed as \( K_2: (3.42,0.34,0.62), K_3: (3.52,0.44,0.62) \) and \( K_4: (3.52,0.34,0.52) \). We use \( K_1 \) to generate an encrypted layer of the original layer (Fig. 9(c)) and Fig. 13(a) show image of this layer. After that, we continue to generate other encrypted layers with the slightly modified keys \( K_2, K_3, \) and \( K_4 \). Fig. 13(b)–(d) indicates the corresponding encrypted layers. It is observed that layers encrypted with slightly different keys are completely incomprehensible. This verifies that, “ciphertexts generated using slightly different keys are completely different from each other”.

With another provision of key sensitivity, the encrypted layer is decoded with key \( K_2, K_3 \) instead of the correct key \( K_1 \), as shown in Fig. 14. The layer is decoded with incorrect keys that completely incomprehensible, and do not leak any information about the original layer.
Fig. 13. Key sensitivity with original layer (Fig. 9(c)) using different keys. (a) Key $K_1$, (b) key $K_2$, (c) key $K_3$, and (d) key $K_4$.

Fig. 14. Key sensitivity analysis for decryption process. (a) Proposed encrypted map, (b) decrypted map with $K_1$, (c) decrypted map with $K_2$, and (d) decrypted map with $K_3$.

4.7 Algorithm Comparison

We compare the properties of the proposed method with the existing encryption method [20,21]—authors use DCT transform for GIS vector map encryption. In these papers, authors select all vertices in a layer and encrypt them by random algorithm in DCT domain, they did not consider important part in each layer. So, this is full encryption methods and still need very long computation time (we proved it in Sections 4.1 and 4.5). Our method only select the significant objects (that is defined by owner based on threshold values) for encryption, it would be very difficult to break the encryption algorithm or try to predict the encrypted part. Furthermore, authors only used a set of random value to encrypt coefficients in DCT domain and they didn’t clearly explain in security evaluation part. We think that it should be broken easily because the key is poorly chosen and key space is too small.

5. Conclusion

In this paper, a new method is proposed which aim to reduce the ratio of encrypted data in GIS vector map but still assure the performance and the high security. This considers how to select significant objects in a layer by using threshold values. After that, the selected vertices are encrypted with random numbers, key values generating from chaotic map and hybrid transforms. We confirm that human perception do not see any information in encrypted map, poor error in decryption step, computation time is less than it in the existing methods, high security and a large amount of GIS vector map data can be protected by this algorithm. The algorithm can be used in many kind of application or standard vector map because it is proposed to encrypt randomly vertices of important/complex objects (polygons and polylines), so it can be applied for any vector map data and GIS database on on/off-lines server.
Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2014R1A1A4A01006663 and NRF-2016R1D1A3B03931003) and the ICT R&D program of MSIP/IITP (R0126-15-1112, Development of Media Application Framework based on Multi-modality which enables Personal Media Reconstruction).

References


Bang Van Nguyen

He received a B.S. degree in School of Electronic & Telecommunication from Hanoi University of Science & Technology (HUST) in 2014. Currently, he is a Master student in Multimedia Communication & Signal Processing Lab in Pukyong National University. His research interests include video processing & application, GIS applications, data security, and smart system.

Suk-Hwan Lee http://orcid.org/0000-0003-4779-2888

He received the B.S., M.S., and 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

He 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 Department of IT Convergence and 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 President of Korea Multimedia Society. His research interests are in the area of digital image processing, multimedia security and watermarking, bioinformatics, weather radar information processing.