Original Article
International Journal of Fuzzy Logic and Intelligent Systems
Vol. 14, No. 4, December 2014, pp. 249-255
http://dx.doi.org/10.5391/IJFIS.2014.14.4.249

ISSN(Print) 1598-2645
ISSN(Online) 2093-744X

Discrete Wavelet Transform for Watermarking Three-Dimensional Triangular Meshes from a Kinect Sensor

Suryo Adhi Wibowo, Eun Kyeong Kim, and Sungshin Kim
Department of Electrical Engineering, Pusan National University, Busan, Korea

Ijfis

Abstract

We present a simple method to watermark three-dimensional (3D) triangular meshes that have been generated from the depth data of the Kinect sensor. In contrast to previous methods, which maintain the shape of 3D triangular meshes and decide the embedding place, requiring calculations of vertices and their neighbors, our method is based on selecting one of the coordinate axes. To maintain shape, we use discrete wavelet transform and constant regularization. We know that the watermarking system needs the information to be embedded; we used a text to provide that information. We used geometry attacks such as rotation, scales, and translation, to test the performance of this watermarking system. Performance parameters in this paper include the vertices error rate (VER) and bit error rate (BER). The results from the VER and BER indicate that using a correction term before the extraction process makes our system robust to geometry attacks.

Keywords: Three-dimensional triangular mesh, Wavelet, Constant regularization, Correction term, Kinect sensor

1. Introduction

Research on geometric processing, three-dimensional (3D) animation, and 3D reconstruction has been growing fast due to rapid technological development. One impact from this situation is the increase in data that can be created, especially 3D data. Because data is one of the intellectual property rights, its protection is of utmost importance and watermarking is one way to protect the data. The watermarking system concept is embedded from the watermark data to the source data. Watermark data can be in a text, audio, image, or video, or in a 3D file, and source data can also be in one of these formats.

Research in watermarking has been performed for many decades. For two decades, 3D watermarking especially has been performed and rapidly developed. Kanai et al. [1] used multiresolution wavelet decomposition to perform digital watermarking in 3D polygons. To perform multiresolution wavelet decomposition, a lazy wavelet has been proposed. This method uses complex computation because the ratio of the wavelet coefficient vector and edges should match. If it does not match, the shape of the 3D polygon will be changed. Other research about watermarking in 3D triangular meshes includes 3D irregular meshes based on wavelet multiresolution analysis [2], identification of a watermarking technique that can be performed on 3D triangular mesh [3], and blind watermarking using oblate spheroidal harmonics [4].

Received: Nov. 16, 2014
Revised : Dec. 11, 2014
Accepted: Dec. 15, 2014

Correspondence to: Sungshin Kim
(sskim@pusan.ac.kr)

Neviere et al. have proposed a new watermarking method for 3D triangular mesh, a constraint optimization framework to perform the watermarking [5]. This method also uses complex computation because it needs the input that has been previously optimized and this method has not been implemented to the data from the Kinect sensor. In this study, we proposed 3D triangular mesh watermarking based on discrete wavelet transform (DWT) and selecting one of the axes from a 3D coordinate system. Although DWT has previously been used by many researchers in 3D triangular mesh watermarking, we perform another approach in order to use the DWT without performing complex computations. Our approach uses constant regularization and a correction term. We also implement this watermarking to the 3D triangular mesh from the Kinect sensor, which can be created from depth data.

This research makes the following contributions:

1) providing a simple method for watermarking 3D triangular mesh,
2) introducing constant regularization and a correction term for 3D triangular mesh,
3) implementing watermarking to the data generated from the Kinect sensor.

The organization of this paper is as follows. Section 1 introduces the watermarking of 3D triangular meshes. In Section 2, 3D triangular mesh data and text data are presented in detail. In Section 3, we describe the insertion and extraction process. Section 4 introduces a correction term for correcting the 3D triangular meshes watermarked before the extraction process. Section 5 presents the experiments and results. Finally, we present our conclusions in Section 6.

2. Data

Building a watermarking system requires two main types of data: source and watermark. We used depth data that has been generated from the Kinect sensor and processed it to become 3D triangular mesh. Our watermark data was the text data because it is easy for the user. The user simply enters the text directly rather than provide an image, voice, or video that represents him or her.

2.1 3D Triangular Mesh Data

We used the kinect sensor as equipment for data acquisition. It produces good depth data when the distance between the object and the sensor are in the range 1.2–3.5 m [6]. To implement this watermarking system, we used a face as the object, following the previously mentioned procedures to obtain the data.

Assume we have coarse depth data that can be represented as a scalar function $d(x, y)$, with x and y as mapping coordinates. Unfortunatelly, the depth data is still two-dimensional (2D). In order to transform 2D to 3D, we can use the following equation

$$d(x, y) \rightarrow v(x, y, z) = \begin{bmatrix} x \\ y \\ d(x, y) \end{bmatrix}. \tag{1}$$

By following [7], we can obtain the smoothness of the 3D face point cloud. Since the 3D face point cloud is not connected at each vertex, the object cannot be realized. The object can be realized if each vertex is connected by an edge to another. This connectivity is computed automatically by creating an edge between two vertices that influence each other. Finally, the object can be represented as $V(x, y, z)$ and has a size $3 \times m$, where m indicates the number of vertices.

2.2 Text Data

In this research, text data can be created by using a mixed set of characters, numbers, or symbols. Because it is created from these components, the text data has a string data type. In order to use text data in the watermark system, it should be converted to binary data.

Assume we have text data t that has a string data type. The text data t will be converted into decimal data by using an ASCII table. After we have decimal data on each component of the text data, we can calculate binary data directly by transforming the decimals to binary. This result can be defined as T, which is a column vector that has n length binary data. Figure 1 shows this part of the process.

3. 3D Triangular Mesh Watermarking

Given the object as source data and the text data as watermark data, the goal of 3D triangular mesh watermarking is to embed watermark data into the source data. Thereby, it is important to adapt watermark data to the characteristics of the source data.

There are two main steps in the 3D triangular mesh watermarking procedure: The first is the insertion process, in which the text data embeds into the object as shown in Figure 1. The second is the extraction process, in which the text data is extracted from the watermarked object as shown in Figure 2.

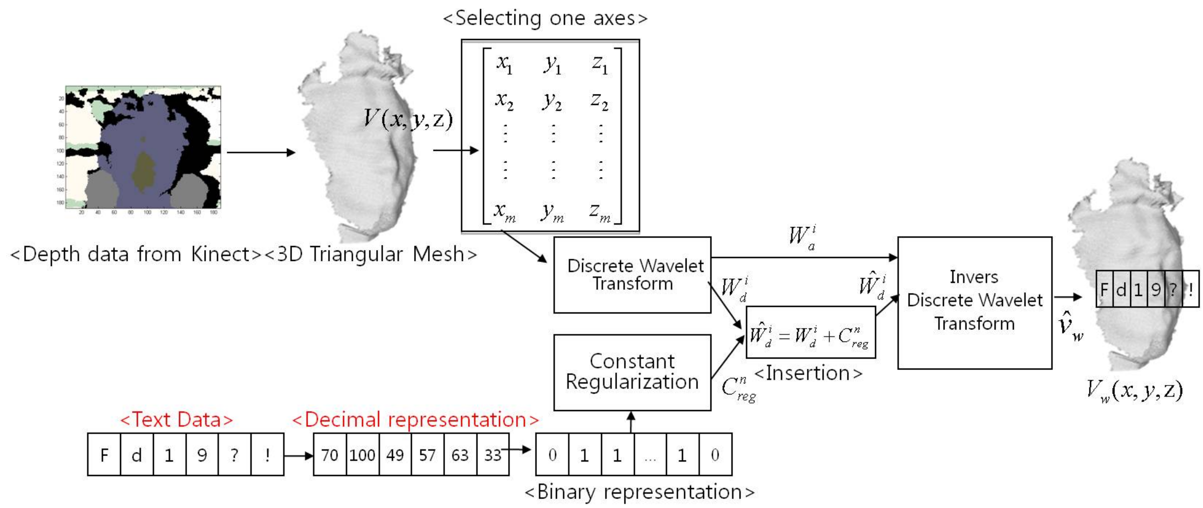

Figure 1. Insertion process of the 3D triangular mesh watermarking.

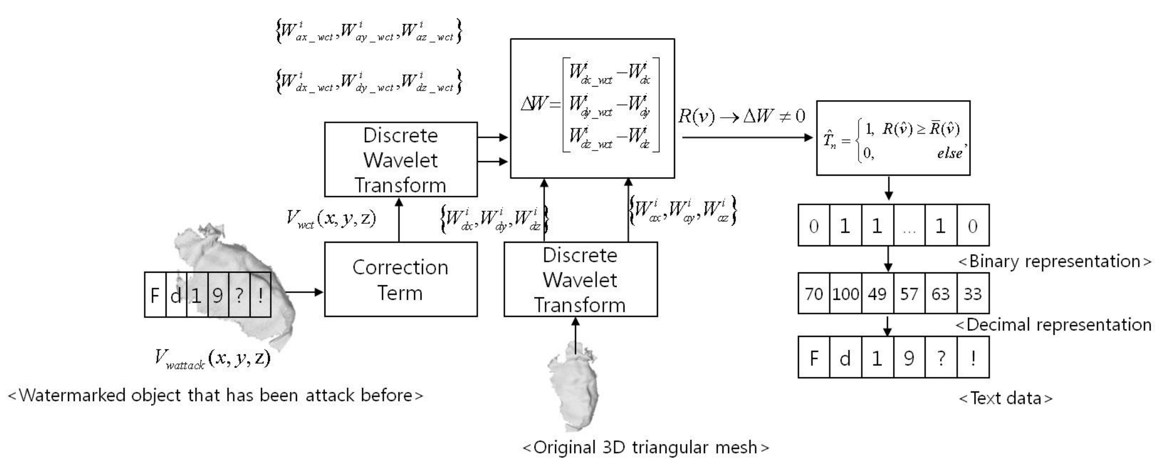

Figure 2. Extraction process of the 3D triangular mesh watermarking.

3.1 Insertion Process

To simplify the explanation, we mention again the object that can be represented as $V(x, y, z)$ and the binary value of the text data, represented as T. Because 3D triangular meshes are very susceptible to changes in the value of vertices as small as possible, we need a method that can maintain the shape even though the binary value of text data T has been successfully embedded. We introduce DWT, selecting one of the coordinate axes and constant regularization to solve this problem.

Defined the column vector $\hat{v}$ is one of the coordinate axes that has been selected and has a size $1 \times m$. Following [8], we compute the approximation coefficients W_a^i and the detail coefficients W_d^i from $\hat{v}$ using DWT one level decomposition. A detailed explanation is as follows: W_d^i can be produced from the downsampling result of the convolution between $\hat{v}$ and a high pass filter coefficient with the downsampling factor is two. W_a^i can be produced from the downsampling result of the convolution between $\hat{v}$ and the low pass filter coefficient with the downsampling factor is two.

DWT has an impact in that where the decomposition level is greater, it will reduce the insertion space. Because of this reason, we used one level decomposition of DWT in our research. Before inserting the text data into the object, there are two main steps that should be required. The first step is that the size of T should match the insertion space. In the other words, $n \leq m/2$. The second step is constant regularization. Constant regularization C_{reg}^n is needed to maintain the shape of the object. The following equation describes the constant regularization

$$C_{reg}^{n} = \begin{cases} C_1, & T_n = 1 \\ C_2, & T_n = 0 \end{cases}, \tag{2}$$

where T_n is binary value in T at index n. In our experiment, we used C_1 and C_2, which equal 0.005 and 0.001, respectively.

Because human eyes are sensitive to dents, it is important to maintain the shape of watermarked object so that it looks similar to the original object. In addition, much information from $\hat{v}$ is contained in W_a^i. Because of these reasons, we select W_d^i as an insertion area; besides, there is no important information contained in W_d^i.

To perform the insertion process [9], we sum each of the detailed coefficients W_d^i with each of the constant regularizations C_{reg}^{n}

$$\hat{W}_d^i = W_d^i + C_{reg}^{n}, \tag{3}$$

where $\hat{W}_d^i$ is the watermarked detail coefficient of DWT.

After the insertion process finishes, the next step is watermarked object reconstruction. To perform this process, we should do inverse DWT using $\hat{W}_d^i$ and W_a^i. This result can be represented as $\hat{v}_w$ and we can use $\hat{v}_w$ to the following equation

$$V(x,y,z) \overset{\hat{v}_w}{\rightarrow} V_w(x,y,z), \tag{4}$$

where $V_w(x,y,z)$ is the watermarked object.

3.2 Extraction Process

Assume we have a watermarked object that has been repaired using a correction term $\{W_{ax\_wct}^i, W_{ay\_wct}^i, W_{az\_wct}^i\}$.The goal from the extraction process is to extract the text data from the watermarked data. Because our watermarking system is non-blind, we need the original object $V(x,y,z)$ to perform it. The first step in the extraction process performs DWT to $V(x,y,z)$ and $\{W_{ax\_wct}^i, W_{ay\_wct}^i, W_{az\_wct}^i\}$. From this process, we have an approximation coefficient $\{W_{ax}^i, W_{ay}^i, W_{az}^i\}$ and the detail coefficient $\left\{W_{dx}^i, W_{dy}^i, W_{dz}^i\right\}$ from $V(x,y,z)$ is also an approximation coefficient $\{W_{ax\_wct}^i, W_{ay\_wct}^i, W_{az\_wct}^i\}$ and detail coefficient $\left\{W_{dx\_wct}^i, W_{dy\_wct}^i, W_{dz\_wct}^i\right\}$ from $V_{wct}(x,y,z)$. By using this information and also following Eqs. (5) and (6), we can determine the values at the insertion area $R(\hat{v})$ and the binary data $\hat{T}_n$ of the text data.

$$\Delta W = \begin{bmatrix} W_{dx\_wct}^i - W_{dx}^i \\ W_{dy\_wct}^i - W_{dy}^i \\ W_{dz\_wct}^i - W_{dz}^i \end{bmatrix}, \tag{5}$$

$$\hat{T}_n = \begin{cases} 1, & R(\hat{v}) \geq \bar{R}(\hat{v}) \\ 0, & otherwise \end{cases}, \tag{6}$$

where $R(\hat{v})$ is equal to $\Delta W \neq 0$, and $\bar{R}(\hat{v})$ is the mean value of $R(\hat{v})$.

Now, we have a set binary value of $\hat{T}_n$. A goal of the next step is text data reconstruction. In a similar way to the previous method, we will first convert the binary value, $\hat{T}_n$ into a decimal value and we can then determine the text data, $\hat{t}$, by converting the decimal value to the character (string data type) using the ASCII table.

4. Correction Term

Attacks on a watermarked object is a reasonable condition. This is possible if a user wants to cheat on a person's work. In order to solve this problem, the watermarking system should have a robust performance. Consider this problem: we introduce the correction term before the extraction process. This correction term satisfies each type of geometric attack. The types of geometric attacks that are proposed in this research are rotation, scale, and translation attacks.

The type of geometric attack should be known first before the correction term process. The procedure of this process is as follows. First, we investigate the rotation attack. We perform a comparison between the vertices from the original object and the vertices from the watermarked object that has been previously attacked. We also investigate the length of the result from each vertex comparison. If we can detect the result of a different length from each vertices comparison, we can determine a rotation attack.

The second procedure investigates a scale attack. To investigate this attack, we divide the vertices of the each axis from the watermaked object with the original one; we then compare the results to each other. We can determine a scale's factor if we can detect these comparison results. If the two investigations previously cannot be known, we assume that the attack is a translation attack. A translation factor can be determined by using vertices subtraction between $V_{wattack}(x,y,z)$ and $V(x,y,z)$.

After we detect the type of attack, we can perform the correction term. The following equation describes the correction term of the rotation attack. The goal of the correction term in the rotation attack is to find the angle that influenced the watermarked object. This problem can be solved by using the least squared method. Assume that the watermarked object that has been attacked is $V_{wattack}(x, y, z)$, and the axis rotation that has been detected before is $x - axes$. Therefore, the equation to perform the correction term for the rotation attack is

$$\min_{\theta} \left\| V(y) - [\cos(\theta) \sin(\theta)] \begin{bmatrix} V_{wattack}(y) \\ V_{wattack}(x) \end{bmatrix} \right\|_2^2, \quad (7)$$

where $V_{wattack}(y)$, $V_{wattack}(z)$, $V(y)$, and θ are the vertices in the $y - axes$ of the watermarked object, vertices in the $z - axes$ of the watermarked object, vertices in the $y - axes$ of the original object, and the angle of the rotation attack, respectively.

In addition, there are three basic rotation matrices $[R_x(\theta), R_y(\theta), R_z(\theta)]$ in the 3D coordinate system and we can use the rotation matrix to repair the watermarked object from the attack.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix},$$
$$R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \quad (8)$$
$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & \sin(theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The remaining correction terms are for the scale and translation attacks. We can use the division between $V_{wattack}(x, y, z)$ and a scale that has been previously known as a correction term for the scale's attack. For the translation attack, we can use the adding operation between the $V_{wattack}(x, y, z)$ and the translation factor. The result from the correction term can be symbolized as $V_{wct}(x, y, z)$.

5. Experiment

To represent characters, symbols, and numbers in text data, we used "Fd19?!" as text data. In order to determine the performance of the system, we performed a testing scheme using three scenario. The first scenario is a testing of the system using a rotation attack; the second scenario is a testing of the system using the scale's attack. The last one is a testing system using a translation attack. The processes were carried out on a 2.4 GHz Intel core i3 processor with 2 GB memory.

5.1 Performance Parameters

Performance of the system was evaluated in terms of the vertices error rate (VER) and bit error rate (BER), defined as follows:

$$VER = \frac{\sum_i |V_i(x, y, z) - V_{w_i}(x, y, z)|}{number\, of\, vertices} \times 100\%, \quad (9)$$

$$BER = \frac{\sum_n \left| T_n - \hat{T}_n \right|}{number\, of\, bit} \times 100\%, \quad (10)$$

where $i = 1, \cdots,$ number of vertices and $n = 1, \cdots,$ number of bit. The number of vertices that we used in this system are 57786 for Obj1 and 60183 for Obj2. The number of bit is 42 bits. We compared the results based on the type of the DWT. We used Haar, Daubechies, and Symlets one level decomposition as the type of DWT and also we compared the result by using correction term and does not use correction term.

5.2 Performance Evaluation

In this experiment, we used a single attack, not a blending attack, to perform the attack. The sample of the angle that influence in $x - axes$ of the 3D coordinate system for a rotation attack are a 30, 90, and 180. For the sample of source data, we used two kinds of 3D triangular mesh, Obj1 and Obj2. The scale factors that we used in this experiment for a scale's attack are 0.5, 0.7, and 2, respectively. The translation attack was also performed in our system. For our translation attack, the translation coefficients are 0.001, 0.01, and 0.1, respectively. However, there is no limitation on angle selection, scale factor, and the translation coefficient, because the value of each factor is in the normal value of the attack.

From Table 1, the Haar DWT by using correction term produces the VER, which has a smaller result than the other type of DWT. Because the Haar DWT is the simplest type of DWT, it does not need a complex computation to perform it. The result of the extraction process, which is extremely good, is caused by a correction term, which is performed before extraction process and without correction term, which is affected by the attacks vertex will contain a value that can distort the shape of the object. The BER values show this result.

Table 1. Performance comparisons

Variable		Type of discrete wavelet transform (%)							
		Using correction term						Does not use correction term	
		Haar		Daubechies		Symlets		Haar	
		VER	BER	VER	BER	VER	BER	VER	BER
Rotation attack(°)									
Obj1	30	2.67×10^{-13}	0	3.06×10^{-6}	0	6.19×10^{-6}	0	3.96×10^{4}	100
	90	2.67×10^{-13}	0	3.06×10^{-6}	0	6.19×10^{-6}	0	8.05×10^{4}	100
	180	2.67×10^{-13}	0	3.06×10^{-6}	0	6.19×10^{-6}	0	3.77×10^{3}	100
Obj2	30	7.35×10^{-13}	0	2.94×10^{-6}	0	5.87×10^{-6}	0	4.03×10^{4}	100
	90	7.35×10^{-13}	0	2.94×10^{-6}	0	5.87×10^{-6}	0	8.44×10^{4}	100
	180	7.35×10^{-13}	0	2.94×10^{-6}	0	5.87×10^{-6}	0	1.06×10^{4}	100
Scale Attack (sacle)									
Obj1	0.5	2.67×10^{-13}	0	3.06×10^{-6}	0	6.19×10^{-6}	0	943.2303	100
	0.7	2.67×10^{-13}	0	3.06×10^{-6}	0	6.19×10^{-6}	0	565.9382	100
	2	2.67×10^{-13}	0	3.06×10^{-6}	0	6.19×10^{-6}	0	1.88×10^{3}	100
Obj2	0.5	7.35×10^{-13}	0	2.94×10^{-6}	0	5.87×10^{-6}	0	2.63×10^{3}	100
	0.7	7.35×10^{-13}	0	2.94×10^{-6}	0	5.87×10^{-6}	0	1.58×10^{3}	100
	2	7.35×10^{-13}	0	2.94×10^{-6}	0	5.87×10^{-6}	0	5.26×10^{3}	100
Translation attack(coef.)									
Obj1	0.001	2.67×10^{-13}	0	3.06×10^{-6}	0	6.19×10^{-6}	0	0.1	0
	0.01	2.67×10^{-13}	0	3.06×10^{-6}	0	6.19×10^{-6}	0	1	0
	0.1	2.67×10^{-13}	0	3.06×10^{-6}	0	6.19×10^{-6}	0	10	0
Obj2	0.001	7.35×10^{-13}	0	2.94×10^{-6}	0	5.87×10^{-6}	0	0.1	0
	0.01	7.35×10^{-13}	0	2.94×10^{-6}	0	5.87×10^{-6}	0	1	0
	0.1	7.35×10^{-13}	0	2.94×10^{-6}	0	5.87×10^{-6}	0	10	0

6. Conclusions

This paper presents a simple method for 3D triangular mesh watermarking produced from a Kinect sensor. The key idea of this research is how to implement the DWT to perform 3D triangular mesh watermarking that does not have complex computation. By performing constant regularization and a correction term, the watermarking can be successful. It also produced a significant result for the VER and BER. Haar DWT is a type of DWT that produces the smallest value for the VER.

Because this research performed a single attack, future research should perform a blending attack. The simple and robust method of blind watermarking also can be performed for future research because in our research, we performed non-blind watermarking.

Acknowledgments

This work was supported by BK21PLUS, Creative Human Resource Development Program for IT Convergence and was supported by the Ministry of Trade, Industry & Energy (MOTIE), Korea, under the Industry Convergence Liaison Robotics Creative Graduates Education Program supervised by the KIAT (N0001126).

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

References

[1] S. Kanai, H. Date, and T. Khisinami, “Digital Watermarking for 3D Polygons using Multiresolution Wavelet Decomposition,” in *Proceedings of Sixth IFIP WG 5.2 GEO-6,* Tokyo, Japan, December 7-9, 1998, pp. 296-307.

[2] M. S. Kim, S. Valette, H. Y. Jung, and R. Prost, “Watermarking of 3D Irregular Meshes Based on Wavelet Multiresolution Analysis,” *Digital Watermarking Lecture Notes in Computer Science, vol. 3710*, M. Barni, I. Cox, T. Kalker, and H. J. Kim, Eds. Heidelberg: Springer Berlin, 2005, pp. 313-324. http://dx.doi.org/10.1007/11551492_24

[3] K. Wang, G. Lavoue, F. Denis, and A. Baskurt, “A comprehensive survey on three-dimensional mesh watermarking,” *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1513-1527, Dec. 2008. http://dx.doi.org/10.1109/TMM.2008.2007350

[4] J. M. Konstantinides, A. Mademlis, P. Daras, P. A. Mitkas, and M. G. Strintzis, “ Blind robust 3-D mesh watermarking based on oblate spheroidal harmonics,” *IEEE Transactions on Multimedia*, vol. 11, no.1, pp. 23-37, Dec. 2009. http://dx.doi.org/10.1109/TMM.2008.2008913

[5] X. Rolland-Neviere, G. Doerr, and P. Alliez, “Triangle surface mesh watermarking based on a constrained optimization framework,” *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 9, pp. 1491-1501, Sept. 2014. http://dx.doi.org/10.1109/TIFS.2014.2336376

[6] M. Zollhöfer, M. Martinek, G. Greiner, M. Stamminger, and J. Süßmuth, “Automatic reconstruction of personalized avatars from 3D face scans,” *Computer Animation and Virtual Worlds*, vol. 22, no. 2-3, pp. 195-202, April 2011. http://dx.doi.org/10.1002/cav.405

[7] S. A. Wibowo and S. Kim, “Three-dimensional face point cloud smoothing based on modified anisotropic diffusion method,” *International Journal of Fuzzy Logic and Intelligent System*, vol. 14, no. 2, pp. 84-90, June. 2014. http://dx.doi.org/10.5391/IJFIS.2014.14.2.84

[8] S. G. Mallat, “A theory for multiresolution signal decomposition: The wavelet representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693, Jul. 1989. http://dx.doi.org/10.1109/34.192463

[9] R. F. Bukit, B. Hidayat, and S. A. Wibowo, “Simulasi dan analisis watermarking tiga dimensi dengan metode transformasi wavelet dan metode fuzzy logic,” Available http://digilib.tes.telkomuniversity.ac.id/index.php?option=com_repo&Itemid=34&task=detail&nim=111090083

Suryo Adhi Wibowo received his B.S. engineering degree and M.S. Engineering degree in Telecommunication Engineering from Telkom Institute of Technology, Indonesia, in 2009 and 2012, respectively. He is currently a Ph.D. candidate at Department of Electrical and Computer Engineering, Pusan National University, Korea. His research interests include computer vision, intelligent system, pattern recognition, and computer graphics.
E-mail: suryo@pusan.ac.kr

Eun Kyeong Kim received her B.S. degree at Department of Electronic Electrical Engineering from Pusan National University in 2014. She is currently M.S. candidate at Department of Electrical and Computer Engineering, Pusan National University. Her research interests include robotics and intelligent system.
E-mail: kimeunkyeong@pusan.ac.kr

Sungshin Kim received his B.S. and M.S. degrees in Electrical Engineering from Yonsei University, Korea, in 1984 and 1986, respectively, and his Ph.D. degree in Electrical Engineering from the Georgia Institute of Technology, USA, in 1996. He is currently a professor at the Electrical Engineering Department, Pusan National University. His research interests include fuzzy logic controls, neuro fuzzy systems, neural networks, robotics, signal analysis, and intelligent systems.
E-mail: sskim@pusan.ac.kr