Representing Realistic Pavement Blocks with Crack Manipulation

SungChul Jung, Chi-Hyoung Rhee and Chang Ha Lee*

School of Computer Science and Engineering, Chung-Ang University, Seoul 156-756, Korea

jcandaniel@gmail.com, kabapoo@cau.ac.kr, chlee@cau.ac.kr
** Corresponding Author: Chang Ha Lee, Ph.D. (chlee@cau.ac.kr)*

Abstract

The depiction of the irregularity in nature is crucial for the realistic representation of the virtual world. For example, real pavement blocks are not always paved evenly, but partially sink over time. Many games or virtual reality systems however represent the pavement too neatly with perfectly flattened surfaces and clean texture images. In this paper, we suggest a method for representing realistic pavement blocks by adding irregularity to blocks. We create uneven surfaces by placing pavement blocks on the randomly generated terrain. We also represent worn surfaces by adding crack patterns to the original texture images along the user-controlled spline curves with Poisson equations. Our method can generate diverse new crack patterns from images of sample patterns and it would be useful to generate random patterns of other linearly-shaped textures as well. Our method is applicable to real-time rendering systems such as games and virtual reality systems since the cracking effects are incorporated into the texture images.

***Key Words**: Pavement, Perlin noise, Crack, Spline curve, Poisson equation, Irregularity*

1. Introduction

One of the main purposes in computer graphics is to represent highly realistic scenes. The unreality in the computer graphics mostly comes from the clean and regular models. There has been a significant progress in realistic rendering, but many of models shown in games or virtual reality systems are still too clean and regularly modeled. For example, the pavement blocks are generally rendered with two large triangles textured with neat block images. The real life pavement blocks are uneven, soiled, and weathered as shown in Figure 1(a) and (c). Representing the irregularity in nature is crucial for realistic representation of the real world such as the pavement blocks. Since manually representing the irregularity is tedious and time-consuming, there has been some research creating irregular natural phenomena. For representing pavement, techniques for generating pavement textures have been proposed including [1] and [2] Also others suggested methods for representing weathering effects such as scratches or cracks [3, 4]. Most of the methods expressing weathering effects simulate physical phenomena, and are limited to quickly represent those effects in real-time rendering systems.

In this paper, we propose a method that models and renders weathered pavement blocks representing the irregularity in the nature. We present two aspects of irregularity in this paper. First, we represent uneven pavement blocks by changing the geometry of blocks. Second, we present weathered blocks cracked over time. We perform image synthesis for generating cracking features on the blocks and provide a convenient interface to generate diverse crack patterns. The suggested image synthesis method is intuitive and easy to implement with a reasonable result. In addition, it would be useful to generate other linear-shaped texture patterns as well. Our method is useful for real-time rendering systems such as games and virtual reality systems since the cracking effects are incorporated into the texture images

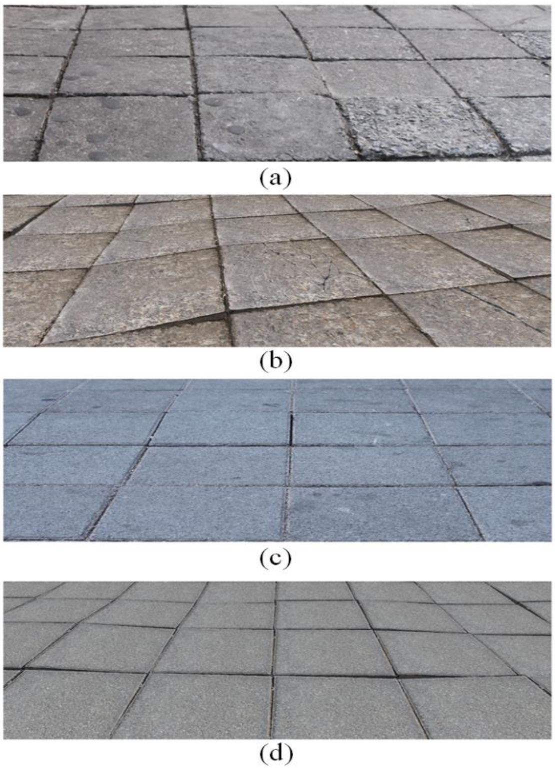


Figure 1. (a) and (c) are Photographs of Actual Pavement Blocks in Real World and (b) and (d) are Pavement Blocks rendered by our Method. Both Real and rendered Blocks the which includes uneven Surface Levels and Worn Blocks which are Partially Cracked Over Time

2. Related Works

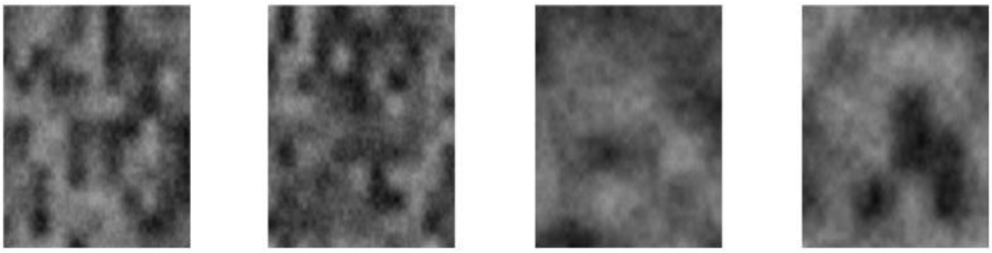
There have been several studies for synthesizing realistic pavement texture patterns. [5] addressed a method for generating pavement texture images by using a square packing technique. The square cells are tightly packed inside an arbitrary pavement shape by using the particle simulation method with a proximity control. After square cells are packed, each square cell is substituted by an individual stone shape. Cellular texture method suggested by [6] generates textures representing cellular patterns such as masonry, tiling, and shingles. Their method produces the feature-based resulting texture derived from the patterns of cells while considering the underlying geometric information. Another approach by [1] proposed an algorithm for generating graphical materials such as wood patterns, stones, plants, and ground materials. His method initially produces regular patterns, and then generates random disturbances on them.

Representing cracks is one of the most important factors in realistic aging and weathering effects. In the study of [2], they surveyed many researches about aging and weathering effects. Generally, there are two types of methods for creating cracks. One is the method using physically-based simulations and the other is the image-based method. The physically-based method can generate realistic cracks which are physically correct. However, it takes too much computation time and it is relatively difficult to control. On the other hand, the image-based method provides easier manipulation and implementation. [7] proposed a technique to simulate the crack propagation on three-dimensional surfaces. Their method suggests a semi-physical solution by considering the interaction between cracks and spectrum stresses. [3] proposed a procedural method for creating worn edges of exposed objects by using an example-based approach. [4] presented a physically-based method for generating surface cracks by applying a stress field defined heuristically over triangles of the model. [8]

proposed an image-based method to represent a weathering effect on the surface materials by using iterative manipulations on the single input image. [9] introduced a method that stores two-dimensional fracture patterns and profile curves in an atlas according to the material types. Their method then uses a procedural automatic mapping of fracture patterns on the object surface. [10, 11] suggested image synthesis methods which are effective for the composition of multiple images. Their methods use the gradients between the target image and the source image. Similarly, a gradient domain painting system was introduced by [12]. Their system can generate a crack through the image synthesis in gradient domain along the user's manual strokes.

3. Representing Uneven Pavement

For generating a naturally curved terrain, we generate a random noise by using the Perlin's method [13, 14] which generates gradient noises. Based on the underneath ground properties and oldness of the pavement, level of terrain unevenness would be various. The terrain underneath an older pavement on the softer ground would be more highly curved, and the terrain underneath a newer pavement on the firmer ground would be less curved. To express the level of terrain unevenness, we generate a two dimensional noise in various frequencies and amplitudes by adjusting the frequency and the amplitude of the noise function. Figure 2 shows height maps for uneven surfaces in various levels. The generated noise is stored in a two dimensional array and used as a height map representing the terrain underneath of the pavement blocks.



(a) (b) (c) (d)

Figure 2. Height Maps in various Frequencies. Each Map is created by Controlling the Frequency and the Amplitude of the Noise Function

After generating a height map representing the underneath terrain, we place pavement blocks on the curved terrain. This procedure resembles the real world procedure to pave blocks. First, we uniformly pack the pavement blocks in the rectangle containing the height map, which is the horizontal bounding box of the terrain. Then, we vertically reposition the blocks so that they lie at the same height with the underlying terrain. Finally, we rotate each block so that it is aligned with the orientation of the underlying terrain area covered by it. Figure 3 illustrates the overall procedure to pave blocks on the curved terrain.

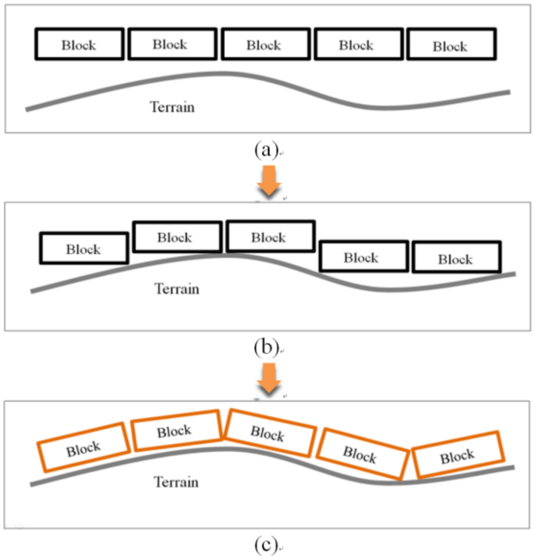


Figure 3. The procedure to create uneven pavement blocks. (a) First, the terrain is generated and blocks are uniformly packed. (b) Next, each block is vertically translated by the height of the underlying height map. (c) Finally, each block is rotated by analyzing the orientation of the underlying terrain using principal component analysis

To compute the orientation of the underlying terrain, we crop the height map covered by e ach block, and generate a point set by converting pixels of the cropped height map into points , and compute three orthogonal directions by using the principal component analysis (PCA)[1 5,16]. The PCA is done by computing the eigenvectors of a covariance matrix of the data poi nts on the underlying terrain. Figure 4 (a) shows an axis-aligned pavement block, and Figure 4 (b) shows a set of points on the underlying terrain and the pavement block aligned with it.

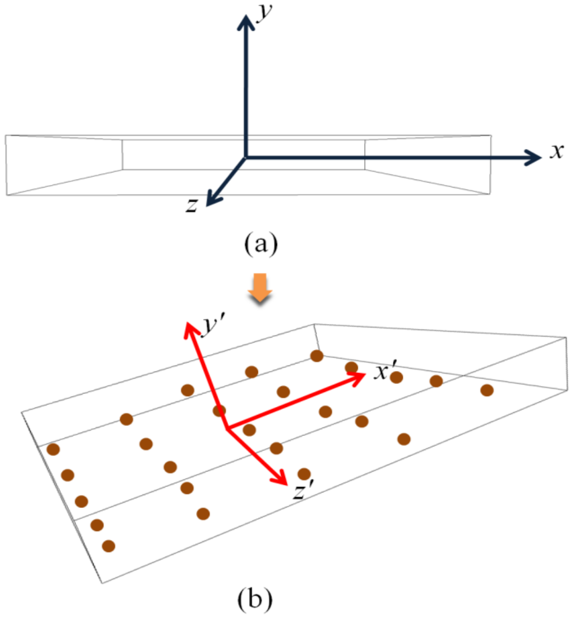


Figure 4. Each block is aligned with the underlying terrain orientation. (a) The axis-aligned pavement block is vertically positioned on the terrain. (b) The block is rotated so that it is aligned with the underlying terrain orientation computed by the principle component analysis.

In this paper, we create a pavement realistically by addressing an irregularity of unevennes s. Representing uneven pavement blocks which are partially raised or sunk is crucial for mod eling the irregularity in the paved block. Figure 4 shows the resulting uneven pavement block s.

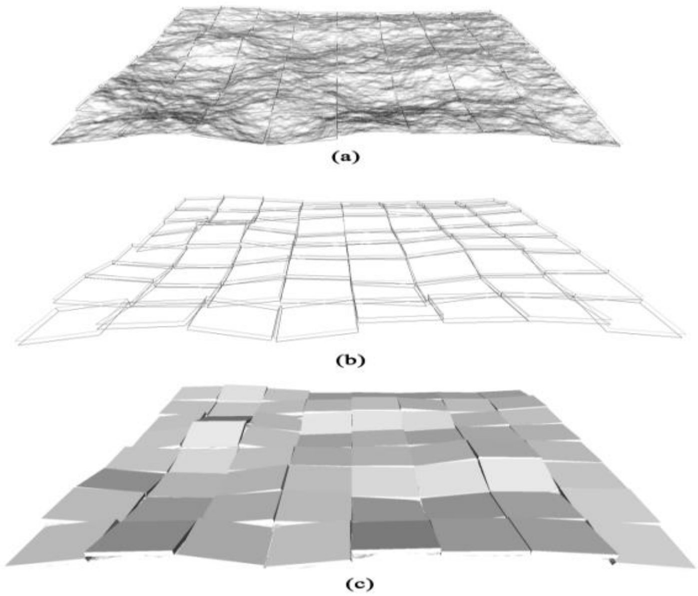


Figure 5. (a) The Underlying Terrain generated by a Perlin Noise. (b) The Pavement Blocks placed on the Curved Terrain in Wire Frames. (c) The Blocks Paved on the curved Terrain in Solid Rendering

In this paper, we create a pavement realistically by addressing an irregularity of unevenness. Representing uneven pavement blocks which are partially raised or sunk is crucial for modeling the irregularity in the paved block. Figure 5 shows the resulting uneven pavement blocks.

4. Representing Cracked Pavement

In this section, we introduce a way to represent weathering effects such as cracks of the block. Our method can produce various patterns of cracks and composite them to diverse uncracked texture images. Therefore, our suggestion generates various types of cracked texture images from a sample crack by the user manipulation with less effort. There are image based methods [17, 18] and physically-based modeling methods for generating crack patterns. Physically-based methods generally take more time and complexity than other methods, and gradient-based methods show more natural composition than image-based methods. Therefore, in this paper, we present a gradient-based method for generating diverse new cracked textures by synthesizing cracks, which are sampled from examples and tweaked by users, upon the pavement textures. Our method is intuitive and easy to implement. There are two stages for creating cracks. The first is the crack manipulation step, and the second step is the synthesis between the target image and the crack image by solving a Poisson equation using Gauss-Seidel iteration [19].

To create various natural crack patterns effectively, we use spline curves to tweak the original crack pattern. First, users can select a crack pattern with a concerned region on the sample crack image. Then, we apply spline curves for manipulate the original crack pattern and generating diverse new patterns. Users can easily re-design the crack shape by tweaking the control points of spline curves, and get a new crack shape. To transform the original pattern into a desirable shape, we composite the patterns in gradient domain [10].

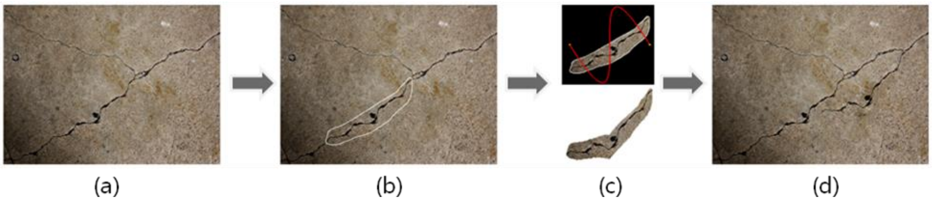


Figure 6. Crack Composition Procedure: (a) Original Image, (b) Drawing an Interested Crack Region, (c) Manipulating Crack Patterns using a Spline Curve, (d) the Synthesize. (a) and (c) Use the Poisson Equation

Next, for a realistic crack composition, we use the Poisson partial differential equation with Dirichlet boundary conditions.

$$\nabla^2 \varphi = \mathcal{F} \tag{1}$$

Where f is real valued functions,

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) = f(x, y) \tag{2}$$

For solving this equation, we used a multi-grid method, Gauss-Seidel iterations. Finally we can generate a reasonably synthesized image. Figure 6 shows cracked texture images generated by our method and Figure 7 shows the final weathered pavement block images by our method.

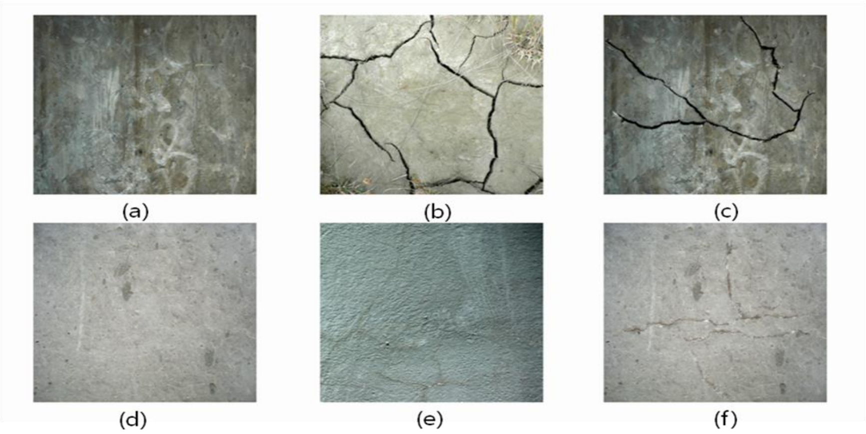


Figure 7. Result Images: (a) Target Image, (b) Crack Image, (c) Result Image by Synthesizing the Target Image and the Crack Patterns generated by Tweaking the Cracks in the Crack Image. (d), (e), and (f) Were Adapted by the Equivalent Procedure

5. Result

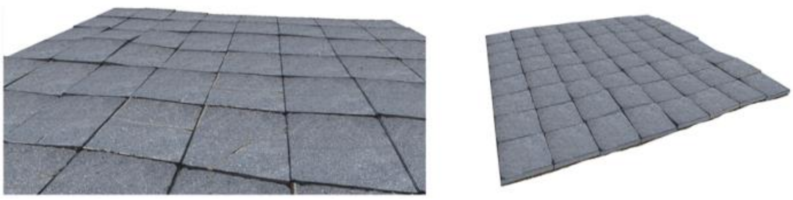
We generate uneven and cracked pavement blocks by combining the two methods described in Section 3 and 4. Figures 8~10 show result images generated by our method with various texture images. Each example shows different levels of irregularity by using different frequencies for generating uneven pavements. Table 1 shows the timing results for the pre-computation and the rendering of our method. Our method takes a several second for pre-computation, but the rendering time is reasonable to be used in a real-time rendering system. We used an Intel core i7 2.66GHz PC system with 8GB memory for this data.

Table. 1. Timing information for the pre-computation and the rendering of our method (milliseconds)

Tile	Pre-compute Time			Rendering Time
	Perlin noise	PCA	Total Time	
2×2	508.09	1180.25	1904.61	0.0003794
4×4	508.09	4791.01	5515.37	0.0007589
8×8	508.09	19067.70	19792.06	0.0015177

6. Conclusion

In this paper, we propose a method for creating a realistic pavement on the terrain by representing the irregularity. Our method is intuitive and easily applicable to most of arbitrary random terrains in games, virtual reality systems, or other interactive graphics systems. In this paper, we generate randomly curved terrains for representing uneven pavements and randomly generated or manually tweaked cracks on the blocks for representing worn surfaces. Our approach for generating cracks can be applied to any surfaces such as walls or buildings. Moreover, it would be useful to generate other weathering effects by compositing selected patterns to texture images. In addition, our suggestion for representing uneven pavements actually transform the object's geometry, so it has a potential to represent physical phenomena such as puddles, wet or soiled surfaces, snow on the pavement, and mosses or grasses between bricks. For real-time systems, generating uneven surfaces and synthesizing textures on the GPU21 would be also worth to try for the future works.



(a) low irregularity

(b) mid irregularity

(c) high irregularity

Figure 8. Uneven and Cracked Pavement Blocks generated by our Method (a) on a Terrain by a Low Frequency Height Map, (b) on a Terrain by a Mid Frequency Height Map, and (c) on a Terrain by a High Frequency Height Map

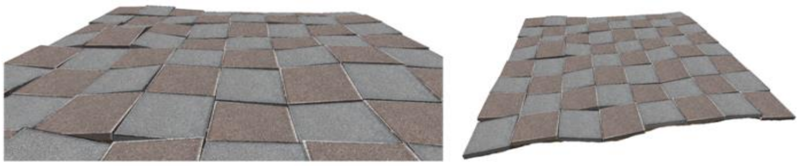
(a) low irregularity

(b) mid irregularity

(c) high irregularity

Figure 9. Results of our Method with Different Texture Images. You can See Cracks on Random Blocks. (a) Shows slightly Sunk Pavement Blocks, (b) Shows more Sinking of the Pavement, and (c) Shows severely Sunk Pavement

(a) low irregularity



(b) mid irregularity

(c) high irregularity

Figure 10. Results of our Method with Checker Patterned Textures. Images from (a) to (c) Show the Different Levels of Terrain Irregularity

Acknowledgements

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(grant number 2010-0011068). This work was also supported by Korea Science & Engineering Foundation through the NRL Program(Grant ROA-2008-000-20060-0).

References

[1] C. I. Yessios, “Computer drafting of stones, wood, plant and ground materials”, SIGGRAPH '79 Proceedings of the 6th annual conference on Computer graphics and interactive techniques, Chicago, USA, **(1979)** August 8-10.
[2] S. Merillou and D. Ghazanfarpour, “A survey of aging and weathering phenomena in computer graphics”, Computers & Graphics, vol. 32, no. 159, **(2008)**.
[3] C. Brachmann, B. Walther-Franks and J. Loviscach, “Exemplar-based Worn Edges”, Eurographics Short Presentations, Prague, Czech Republic, **(2007)** September 3-7.
[4] H. N. Iben and J. F. O’brien, “Generating Surface Crack Patterns”, Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, Vienna, Austria, **(2006)** September 2-4.
[5] K. Miyata, T. Itoh and K. Shimada, “A method for generating pavement textures using the square packing technique”, The Visual Computer, vol. 17, no. 475, **(2001)**.
[6] J. Legakis, J. Dorsey and S. Gortler, “Feature-based cellular texturing for architectural models”, SIGGRAPH`01 Proceedings of the 28th annual conference on Computer graphics and interactive techniques, Los Angeles, USA, **(2001)** August 12-17.
[7] S. Gobron and N. Chiba, “Crack pattern simulation based on 3D surface cellular automata”, The Visual Computer, vol. 17, no. 287, **(2001)**.
[8] S. Xue, J. Wang, X. Tong, Q. Dail and B. Guo, “Image-based Material Weathering”, Computer Graphics Forum, vol. 27, no. 617, **(2008)**.
[9] B. Desbenoit, E. Galin and S. Akkouche, “Modeling cracks and fractures”, The Visual Computer, vol. 21, no. 717, **(2005)**.

[10] P. Pérez, M. Gangnet and A. Blake, "Poisson Image Editing", ACM Transactions on Graphics, vol. 22, no. 313, **(2003)**.
[11] A. Levin, A. Zomet, S. Peleg and Y. Weiss, "Seamless Image Stitching in the Gradient Domain", IEEE Transactions on Image Processing, vol. 15, no. 969, **(2006)**.
[12] J. McCann and N. S. Pollard, "Real-Time Gradient-Domain Painting, ACM Transactions on Graphics, vol. 27, no. 3, **(2008)**.
[13] K. Perlin, "Improving noise", SIGGRAPH `02 Proceedings of the 29th annual conference on Computer Graphics and interactive techniques, Texas, USA, **(2002)** July 21-26.
[14] K. Perlin, "An Image Synthesizer", SIGGRAPH '85 Proceedings of the 12th annual conference on Computer graphics and interactive techniques, San Francisco, USA, **(1985)** July 22-26.
[15] C. R. Rao, "The Use and Interpretation of Principal Component Analysis in Applied Research", The Indian Journal of Statistics, vol. 26, no. 329, **(1964)**.
[16] E. Lengyel, "Mathematics for 3D game programming & computer graphics", Charles River Media Publishers, Massachusetts, **(2004)**.
[17] M. David, "Image-Guided Fracture", GI '05 Proceedings of Graphics Interface, Victoria, Canada, **(2005)** May 9-11.
[18] B. Wyvill, K. V. Overveld and S. Carpendale, "Rendering Cracks in Batik", Proceeding NPAR '04 Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering, **(2004)** June 7-9.
[19] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, "Numerical Recipes in C: The Art of Scientific Computing", Cambridge University Press Publishers, New York, **(1992)**.
[20] Y. O. Park and J. W. Park, "A Study on the 3 Dimension Graphics accelerator for Phong Shading Algorithm", The Journal of IWIT, vol. 10, no. 97, **(2010)**.