

Fractal Terrain Generator

Budi Hartanto
University of Surabaya
Jl. Raya Kalirungkut
Surabaya
(031) 2981395
budi@ubaya.ac.id

Monica Widiarsi
University of Surabaya
Jl. Raya Kalirungkut
Surabaya
(031) 2981395
monica@ubaya.ac.id

Gunawan Widjaja
University of Surabaya
Jl. Raya Kalirungkut
Surabaya
(031) 2981395

ABSTRACT

Simulated terrains have been extensively used in many flight simulators or games. Unfortunately, creating large terrains can be considered tedious and complex job. Therefore some flight game makers only created a limited amount of terrain and reused the same terrain for another level of the game. Though it seems solving the problem, increasing the number of terrains used in the program may make the game becomes more interesting. Since terrain actually is a fractal, the terrain can be generated using the concept of fractal. The resulted terrains can be saved in a text file to be utilized by any other programs. This research has overcome the terrain creation problem by generating user-defined terrain automatically.

Keywords

Terrain, Fractal, Flight Simulator, Flight Game.

1. INTRODUCTION

In many flight game programs, users have to fly an airplane through a sequence of certain terrain. The terrain usually undulates and it is one of the users' tasks to fly the plane through the hill and valley and to avoid collision with any of them.

In its simplest implementation, the game can create several hill and valley and store the data in a certain media. When the game needs to render the terrain, these stored hill and valley are read back from the media and rendered to the game environment. Due to the difficulty in hill and valley generation, there are only few hill and valley stored in the media. Because there are not many hill and valley can be taken from the media, the game will usually reuse the first available part of hill and valley and put them as the next hill and valley to be rendered on the game.

Though this approach can be used for several kind of game without loosing the users' interest, the availability of big supplies of hill and valley data can make the game more interesting and challenging. The problem is how can we provide affluent data of hill and valley easily?

One proposed solution to create affluent data of hill and valley is by generating them automatically. These hill and valley will build the terrain that can be used by other flight simulator or game program. A fractal method will be used to create the hill and valley in the terrain.

2. FRACTAL

Fractal is a pattern where partial or whole parts of the patterns have self similarity to the general pattern itself [4]. This fact will enable

us to generate fractal recursively and heuristically. One example of fractal object is the Koch curve. The curve can be created by dividing a certain line into three parts. The middle part of the line then can be changed to a right triangle without its bottom part. If we recursively perform the same process to its sub-line, then we will get the Koch curve. Figure 1 shows several Koch curves created from a line, triangle, and polygon.

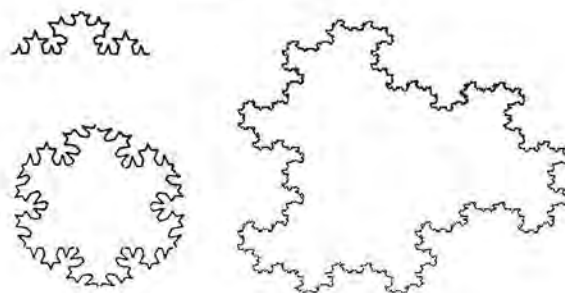


Figure 1. Several kinds of Koch curves.

If we magnify certain part of any Koch curves, we will get the similar pattern of the curve to its bigger part. This property is called self similarity and it becomes the major property of fractal object. As a matter of fact, many nature objects in our everyday life have this property. As an example a branch of a pine tree has a self similarity to the whole tree itself. A certain part of coast line has self similarity to its bigger part, etc. This phenomenon applies as well to terrain with its hills and valleys [6, 7]. A magnifier of a certain hill will give us several smaller hills and valleys as well.

Because of the terrain-self-similarity property it can be concluded that a terrain actually is a fractal in itself. Therefore it should be possible to generate terrain using any fractal method available.

3. GENERATING TERRAIN

3.1 Mid Point Displacement

Midpoint displacement is one of the fractal methods that can be used to create a terrain [1]. At the beginning set up four points that build up a square. Assign a certain height value to each point of the square and perform the following processes recursively:

- Divide the square to another four equal-size squares.
- Assign the height value at the middle of the big square. The value can be found by averaging the height value from all points in the big square corners. Add a modified

Gaussian-distribution-based random value to this average.

- Assign the height value at the middle point of each big square side. The value can be found by averaging the height value from its three adjacent points. Add a modified Gaussian-distribution-based random value to this average.
- Repeat the process to all newly generated squares until it reaches a certain set-up threshold.

The depth of the recursive process will determine the number of vertices that compose the grid. The relation of the number of vertices to the number of recursive level can be specified as:

$$\text{Number of Vertices} = (2^{(n-1)} + 1)^2$$

Where n is equal to the number of recursive level. The number of vertices will determine the smoothness of the terrain. Increasing the number of recursive level will increase the number of vertices generated and as a consequence will increase the smoothness of the rendered terrain. However increasing the number of recursive level will increase the computation time and the memory space to store the on going processing data.

Data generated from the mid point displacement can be seen visually by simply connecting adjacent vertices with line. The row and column position of the squares can be used as the coordinate of the terrain and the height value of each point in the square can be used as the height of the terrain. In this simplest visualization, you can see the generated data of the terrain in a wireframe style visualization.

3.2 Gaussian Distribution

Gaussian distribution is commonly called normal distribution [8]. In this distribution, most data will cluster around the mean. That is why the curve of this distribution will appear as a bell. The function of the Gaussian distribution can be seen as follow:

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \text{Eq. 1}$$

where:

μ = mean

σ = standar deviation

x = continuous random value at $-\infty < x < +\infty$

Figure 2 shows two Gaussian distributions. From the figure, it can be said that the mean in the Gaussian distribution a is less than the mean in the Gaussian distribution b . The mean of the Gaussian distribution is taken place at the peak of the shape. Meanwhile, from the overall shape of the shape, it can be said that the standard deviation of Gaussian distribution a is – again – less then the standard deviation of Gaussian distribution b . A Gaussian distribution with smaller standard deviation will have higher bell shape than the one with higher standard deviation.

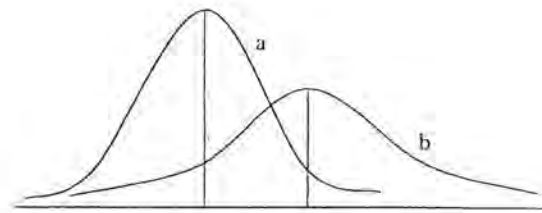


Figure 2. Gaussian bell-shape distribution

One property of the Gaussian distribution is:

$$\int_{-\infty}^{\infty} p(X)dx = 1 \quad \text{Eq. 2}$$

therefore:

$$\int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 1 \quad \text{Eq. 3}$$

from this property it can be concluded that the value of μ and σ will always be in the range $0 \leq \mu, \sigma \leq 1$ because the total area of the curve is 1. To simplify the usage of Gaussian distribution, we need to convert the distribution to a standard Gaussian distribution. A standard Gaussian distribution is a Gaussian distribution with value 0 and 1 for the μ, σ respectively [3, 8].

Random numbers that should be used in generating the terrain must confirm with this standard Gaussian distribution. Unfortunately many computer programming language does not provide standard Gaussian distribution. Instead, most computer programming language provide a uniformly distribution random number.

In order to solve this problem, Goerge Edward Box and Mervin Muller introduced a Box-Muller transform that can be used to translate uniformly distributed random numbers to standard Gaussian distribution random numbers [9, 10]. The method takes two uniformly distributed random numbers and maps them to two independent standard Gaussian distributed random numbers.

Figure 3 shows the illustration of random numbers mapping from the uniform to Gaussian. In Box-Muller transform, two uniformly random number x_1 and x_2 which are distributed between 0 and 1 will map to two Gaussian numbers z_1 and z_2 by the following functions:

$$z_1 = \sqrt{-2 \ln x_1} \cdot \cos(2\pi x_2) \quad \text{Eq.4}$$

$$z_2 = \sqrt{-2 \ln x_1} \cdot \sin(2\pi x_2) \quad \text{Eq.5}$$

Unfortunately, calculating sine and cosine for many repeated processes can be considered costly. Therefore it is necessary to modify the functions and solve the equation in polar form. Given x_1 and x_2 in the range -1 to 1, set the value of s as $x_1^2 + x_2^2$. If $s=0$ or $s \geq 1$ discard x_1 and x_2 and find another x_1 and x_2 pair that generate s in the open interval 0 to 1.

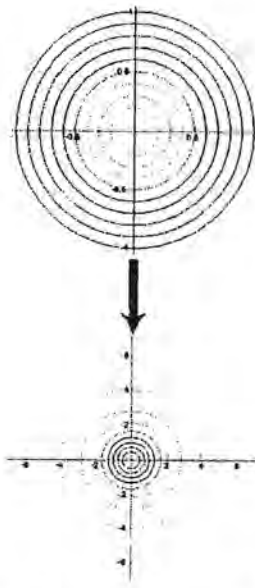


Figure 3. Uniform to Gaussian distributed random numbers

In the polar form, the mapping of uniform distribution to Gaussian can be expressed as [5]:

$$z_1 = x_1 \cdot \frac{\sqrt{-2 \ln s}}{s} \quad \text{Eq.6}$$

and

$$z_2 = x_2 \cdot \frac{\sqrt{-2 \ln s}}{s} \quad \text{Eq.7}$$

3.3 Modified Gaussian Random Value

Before the Gaussian distribution random value can be added to the height field, modify this value to include the recursive level into account. The recursive level determines the number of hills and valleys generated by the program.

A scaling factor will be applied to the Gaussian-distribution-based random value. The scaling factor that will be used is specified as:

$$d_n = (0.5)^{nH/2} d \quad \text{Eq.8}$$

where:

d_n = scaling factor at recursive level n

n = level of the recursive process

H = roughness level of the surface

d = initial scaling factor

From the formula, it can be concluded that as the recursive level is getting higher, the scaling factor is getting smaller. This behavior matches perfectly with the natural terrain property. Most points in the terrain will have big height differences at the lower part of the hill and small height differences at the pinnacle.

Meanwhile the parameter H in the formula can be used to alter the roughness of a surface. Smaller value will create a rougher surface,

and greater value will create smoother surface. Figure 4 show the difference of result using different values of H .

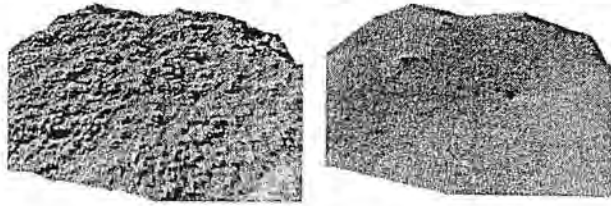


Figure 4. Surface roughness as determined by the value of H

3.4 Building and Storing the Terrain

In order to be able to be used by any other program, the generated terrain data must be stored in a file with general format that can easily be opened and read by any other program. In this research, the file format that is used for storing the terrain is a text file.

Three dimensional objects can be built from a collection of mesh triangles. The triangle shape is chosen because triangle is the simplest convex object. Therefore this object can not be divided anymore to another object especially to a non convex object. Guaranteeing convexity of an object is compulsory in object's rendering process.

Since the result of mid point displacement will generate mesh squares (see figure 5) then it is necessary to change the mesh squares to mesh triangles. Figure 5 shows the modification that must be made to change the generated mesh squares to mesh triangles.

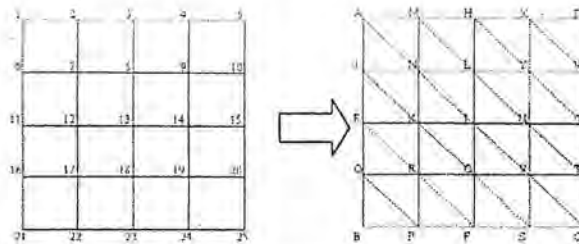


Figure 5. Surface roughness as determined by the value of H

One final process that must be performed after having all the required data is definitely storing the data itself. Two kinds of data that is compulsory to be stored are the vertices and faces. Figure 5 actually have already given clues about the vertices and faces relation to the indices of data. Numbers at the left picture of figure 5 shows the indices used for the vertices, while letters at the right picture of figure 5 shows the relation of indices to the faces of the mesh triangles.

Therefore, if figure 5 represents a certain generated terrain then there will be 25 indices that will be used to store all vertices. Meanwhile the number of faces to be generated will follow the following formula:

$$faces = 2 \cdot (\sqrt{vertices} - 1)^2 \quad \text{Eq.9}$$

In this case, the generated terrain in figure 5 will have 32 faces.

To avoid lighting problem in rendering process, all faces must be specified in either clockwise or counter clockwise manner. In this program, all faces will be stored in counter clockwise manner. Therefore the first and the second faces for terrain such as shown in figure 5 will be AJN and ANM respectively.

In general, data that will be stored in the text file are:

- The number of vertices and faces
- All vertices that build up the terrain. Each vertex will contain its three dimensional position (x, y, z)
- All faces that build up the terrain. Each face will refer to vertices index that build the face. Therefore instead of storing A, J, N, as the first face of terrain in figure 5, the application will store 1, 6, 7 as face's vertex indices.

3.5 Rendering

In order to prove the correctness of the terrain data generated by the program, a rendering facility will be built to show the terrain data visually. An OpenGL graphics library [2] will be used in the program to assist the process of terrain visualization. By this library, one can set up a camera-like viewing to see the terrain three dimensionally.

To increase realism of the terrain, a certain color scheme will be set up and used for coloring the terrain. Several certain heights will be used as a threshold for certain colors. Therefore, hill will have different color from valley. A texture mapping is used as well in the program to improve the realism of the terrain.

4. RESULT AND DISCUSSION

Figure 6 to 9 show several terrains that have been built using the application. Several parameters that influence the shape of the generated terrain are:

- The maximum number of recursive level (Iteration)
- The number of squares used (Tile)
- The roughness of terrain (Rough)
- Initial scaling factor (Scale)

Figure 6, shows terrain with many low hills. This kind of terrain can be generated using lower value of scale. In this example, the value of the scale is 50. On the other hand, increasing the scale value will yield terrain with many high hills. Figure 7 shows this kind of terrain. In this example, 75 is chosen as the value of the scale.

The roughness of the terrain can be set up as well in the program. Figure 6 shows the roughness of the terrain with value 3, while figure 7 shows the roughness of the terrain with value 2. As explained in sub chapter 3.3., smaller value of this parameter will make the terrain rougher.

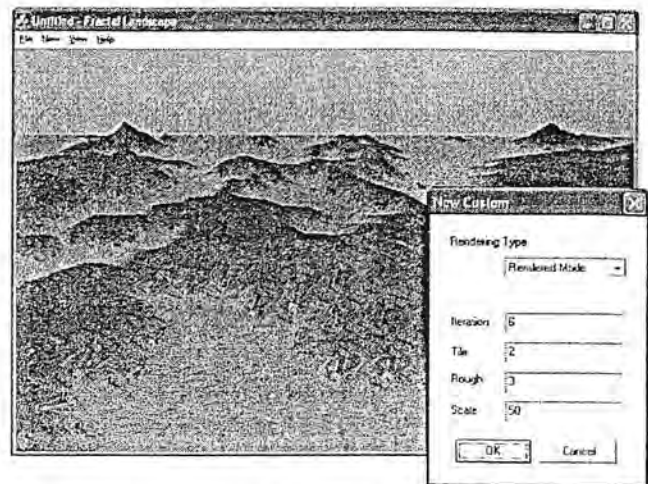


Figure 6. Lowland terrain

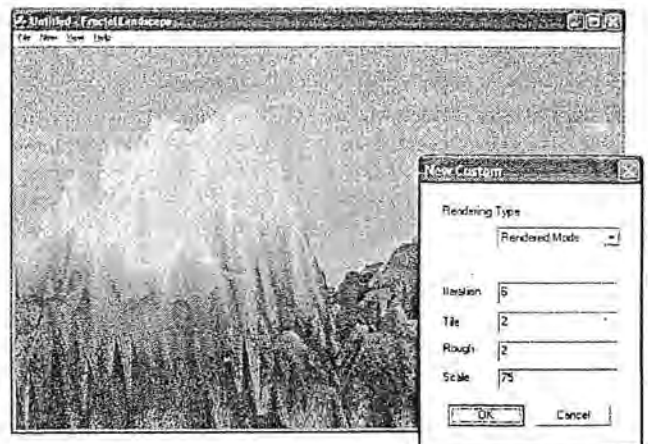


Figure 7. Highland terrain

With the assistance from OpenGL graphics library, one can view the terrain with a camera-like viewing. Therefore we can wander the terrain such as a flying bird. Figure 8 actually shows the same terrain such as shown in figure 7, after we fly closer to the terrain.

Two last figures, i.e. figure 9 and figure 10, show the genuine data of the terrain. Figure 9 shows the data in visualized meshed triangle, while figure 10 shows the raw data in text file. Each vertex and face data in the text file is always preceded with the keyword "v" and "f". This keyword will ease other programmers to read and interpret the data for their own use.

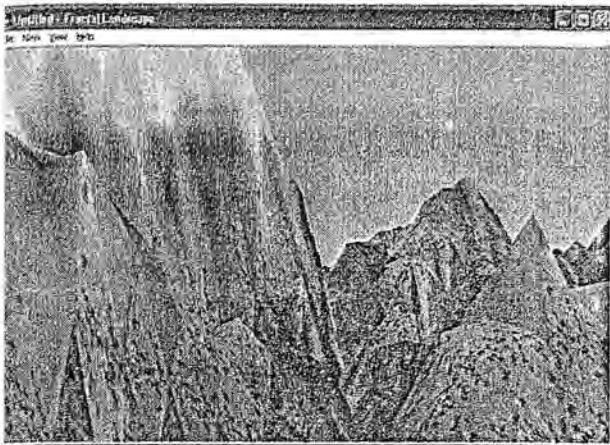


Figure 8. Walkthrough the terrain

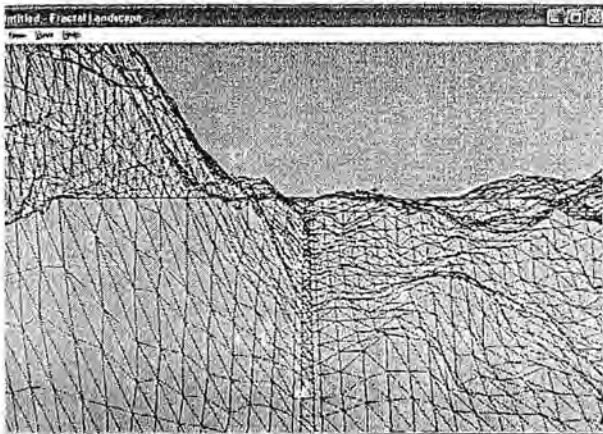


Figure 9. Mesh triangles of the terrain

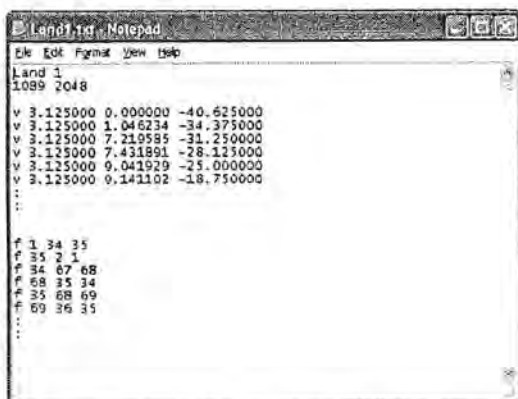


Figure 10. Data stored for a certain terrain

5. CONCLUSION

In general, this research can be said have already achieved its goal. Generating data to create terrain can be performed easily using this terrain generator. Users can create as many terrains as they need and like. The data of the terrains that are saved in a text file make it accessible and usable to any other programs.

From the visualization of the terrain, one can see that based on parameters provided, many variations of the terrains can be generated by the program. By applying the color and texture to the generated terrains, one can see the pleasing result of the terrains.

For further work, it could be better if the users can modify the generated terrains. Therefore the users can create the terrain that meets exactly to their preferences. Furthermore, the application can be made to allow users select their own preferred color or texture to be applied to the hills or valleys generated. All of this information then can be stored in a text file for further use in other applications.

6. REFERENCES

- [1] Daniel, A. N.D. Random Midpoint Fractals. <http://eldar.mathstat.uoguelph.ca/dashlock/ftax/RMP.html>
- [2] Donald, U. 2009. OpenGL Tutorial. http://www.swiftless.com/tutorials/opengl/opengl_tuts.html
- [3] Eric. W. 1999. Gaussian Distribution. http://bbs.sachina.pku.edu.cn/Stat/Math_World/math/g/g084.htm
- [4] Heinz, O., Hartmut, J., Dietmar, S. 2004. Chaos and Fractals: New Frontiers of Science. Springer.
- [5] Taygeta. N.D. Generating Gaussian Random Number. <http://www.taygeta.com/random/gaussian.html>
- [6] William, W., Paul, M., Forrest, M. 1997. The Fractal Landscape Realizer. <http://research.esd.ornl.gov/realizer>
- [7] Wikipedia. 2009. Fractal Landscape. http://en.wikipedia.org/wiki/Fractal_landscape
- [8] Wikipedia. 2010. Normal Distribution. http://en.wikipedia.org/wiki/Normal_distribution
- [9] Wikipedia. 2010. Box-Muller Transform. http://en.wikipedia.org/wiki/Box-Muller_transform
- [10] Wolfram. N.D. Box-Muller Transformation. <http://mathworld.wolfram.com/Box-MullerTransformation.html>

Proceedings of **2nd ICSIIT 2010**

International Conference on
Soft Computing, Intelligent System
and Information Technology

1-2 July 2010, Bali, Indonesia



ICSIIT

MERATUS 



APTIKOM
ASOSIASI PENGURUAN TINGGI INFORMATIKA DAN KOMPUTER

Supported by



Proceedings

ICSIIT 2010

International Conference on
Soft Computing, Intelligent System and Information Technology

1-2 July 2010

Bali, Indonesia

Editors:
Leo Willyanto Santoso
Andreas Handojo



Informatics Engineering Department
Petra Christian University

Center of Soft Computing and
Intelligent System Studies

Proceedings

International Conference on Soft Computing, Intelligent System and Information Technology 2010

Copyright © 2010 by Informatics Engineering Department, Petra Christian University

All rights reserved. Abstracting is permitted with credit to the source. Library may photocopy the articles for private use of patrons in this proceedings publication. Copying of individual articles for non-commercial purposes is permitted without fee, provided that credit to the source is given. For other copying, reproduction, republication or translation of any part of the proceedings without permission in writing from the publisher is not permitted. The content of the papers in the proceedings reflects the authors' opinions and not the responsibilities of the editors.

Publisher:

Informatics Engineering Department
Petra Christian University

ISBN: 978-602-97124-0-7

Additional copies may be ordered from:

Informatics Engineering Department
Petra Christian University, Siwalankerto 121-131, Surabaya 60236, Indonesia

Cover Art production by Adi Wibowo/Informatics Engineering Department

ICSIIT 2010

Table of Contents

Preface.....	xi
Organizing Committee.....	xii
Program Committee.....	xiii
Human Language Technology: The Philippine Context	1
<i>Rachel Edita Roxas, Allan Borra</i>	
Hybrid-Multidimensional Fuzzy Association Rules from a Normalized Database.....	10
<i>Rolly Intan</i>	
Fuzzy Systems & Neural Networks	
A Context-Based Fuzzy Model for a Generator Bidding System.....	18
<i>Moeljono Widjaja</i>	
Neural Networks for Air-Conditioning Objects Recognition in Industrial Environments	24
<i>Enrique Dominguez, J.J. Carmona</i>	
Pattern Recognition Using Discrete Wavelet Transformation and Fuzzy Adaptive Resonance Theory.....	29
<i>Arnold Aribowo, Samuel Lukas, Joannes Franciscus</i>	
Resolving Occlusion in Multi-Object Tracking using Fuzzy Similarity Measure	33
<i>Rahmatri Mardiko, M. Rahmat Widyanto</i>	
Search Engine Application using Fuzzy Relation Method for e-Journal of Informatics Department Petra Christian University.....	39
<i>Leo Willyanto Santoso, Rolly Intan, Prayogo Probo Susanto</i>	
The Use of Gabor Filter and Back-Propogation Neural Network for the Automobile Types Recognition.....	45
<i>Gregorius Satia Budhi, Rudy Adipranata, Fransisco Jimmy Hartono</i>	
Genetic Algorithm & Applications	
A Linear Graph and Genetic Algorithm Approach for Evolving Manipulator Modelling.....	51
<i>Kok Kiong Tan.</i>	

Comparing Genetic and Ant System Algorithm in Course Timetabling Problem	56
<i>Djasli Djamarus</i>	
Gas Distribution Network Optimization with Genetic Algorithm.....	62
<i>K.A. Sidarto, L.S. Riza, C.K. Widita, F. Haryadi</i>	
Hybrid Genetic Algorithm for Solving Strimko Puzzle	68
<i>Samuel Lukas, Arnold Aribowo, James Nagajaya Dyalim</i>	
Optimal Design of Hydrogen Based Stand-Alone Wind/Microhydro System Using Genetic Algorithm	71
<i>Soedibyo, Heri Suryoatmojo, Imam Robandi, Mochamad Ashari, Takashi Hiyama</i>	
Optimization of Steel Structure by Combining Evolutionary Algorithm and SAP2000.....	76
<i>Mohammad Ghazi, Pujo Aji, Priyo Suprobo</i>	
The Hydrophobic-Polar Model Approach to Protein Structure Prediction	82
<i>Tigor Nauli</i>	
University Course Scheduling Using the Evolutionary Algorithm	86
<i>Ade Jamal</i>	
 Artificial Intelligence & Applications	
Adaptive Appearance Learning Method using Simulated Annealing	91
<i>Du Yong Kim, Ehwa Yang, Moongu Jeon, Vladimir Shin</i>	
Bayesian Network and Minimax Algorithm in Big2 Card Game.....	96
<i>Nur Ulfa Maulidevi, Hengky Budiman</i>	
Cell Formation Using Particle Swarm Optimization (PSO) Considering Machine Capacity, Processing Time, and Demand Rate Constraints.....	102
<i>Dedy Suryadi, Ferry Putra, Cynthia Juwono</i>	
Computer Aided Learning for List Implementation in Data Structure.....	108
<i>Ng Melissa Angga, Susana Limanto</i>	
Development Weightless Neural Network on Programmable Chips to Intelligent Mobile Robot.....	112
<i>Siti Nurmaini, Bambang Tutuko</i>	
If-Statement Modification for Single Path Transformation: Case Study on Bubble Sort and Selection Sort Algorithms.....	116
<i>Rahmadi Trimananda</i>	

Implementation of Particle Swarm Optimization Method in K-Harmonic Means Method for Data Clustering.....	120
<i>Ahmad Saikhu, Yoke Okta</i>	
Implementation of Starfruit Maturity Classification Algorithm.....	127
<i>R. Amirulah, M.M. Mokji, Z. Ibrahim</i>	
Improving Choquet Integral Agent Network Performance by using Competitive Learning Algorithms	132
<i>Handri Santoso, Shusaku Nomura, Kazuo Nakamura</i>	
Improving Food Resilience with Effective Cropping Pattern Planning using Spatial Temporal-Based Updated Pranata Mangsa.....	138
<i>Kristoko Dwi Hartomo, Sri Yulianto J.P., Krismiyati</i>	
Knowledge Based System in Defining Human Gender Based On Syllable Pattern Recognition	143
<i>Muhammad Fachrurrozi</i>	
Maintaining Visibility of a Moving Target: The Case of an Adaptive Collision Risk Function.....	146
<i>Ashraf Elnagar, Ibrahim Al-Bluwi</i>	
Measuring Interesting Rules in Characteristic Rule	152
<i>Spits Warnars</i>	
MIDI Composition Tools using JFugue Java API.....	157
<i>Kartika Gunadi, Liliana, Hendra Kurnia Wijaya</i>	
Mobile-based Interaction using Djikstra's Algorithm for Decision Making in Traffic Jam System ...	159
<i>Puji Sularsih, Egy Wisnu Moyo, Fitria H. Siburian, Sigit Widiyanto, Dewi Agushinta R.</i>	
Model and Boarding Simulation for Reducing Seat and Aisle Interferences Between Passenger	164
<i>Bilqis Amaliah, Victor Hariadi, Antonius Malem Barus</i>	
Optimizing Rijndael Cipher using Selected Variants of GF Arithmetic Operators.....	170
<i>Petrus Mursanto</i>	
PCR Primer Design using Particle Swarm Optimization Combined with Piecewise Linear Chaotic Map	176
<i>Cheng-Hong Yang, Yu-Huei Cheng, Li-Yeh Chuang</i>	
Performance Analysis of Heterogeneous Computer Cluster	182
<i>Abdusy Syarif, Saiful Ikhwan, Muhammad Risky</i>	
Reduced Space Classification using Kernel Dimensionality Reduction for Question Classification in Public Health Question-Answering	187
<i>Hapnes Toba, Ito Wasito</i>	

The Developing of Interactive Software for Supporting the Kinematics Study on Linear Motion and Swing Pendulum.....	193
<i>Liliana, Kartika Gunadi, Yonathan Rindayanto Ongko</i>	
University Timetabling Problems with Customizable Constraints using Particle Swarm Optimization Method.....	197
<i>Paulus Mudjihartono, Wahyu Triadi Gunawan, The Jin Ai</i>	
Knowledge & Data Engineering	
A Design of Multidimensional Database for Content-based Television Video Commercial Mining.....	201
<i>Yaya Heryadi, Yudho Giri Sucahyo, Aniat Murni Arymurthy</i>	
Applying Sound to Enhance the Comprehension of Sorting Algorithms.....	206
<i>Lisana, Edwin Pramana</i>	
Data Mining to Build a Pattern of Knowledge from Psychological Consultations.....	211
<i>Sri Mulyana, Sri Hartati, Retantyo Wardoyo, Edi Winarko</i>	
Data Warehouse Information Management System RSUD Dr. Soetomo for Supporting Decision Making.....	215
<i>Silvia Rostianingsih, Oviliani Yenti Yuliana, Gregorius Satia Budhi, Denny Irawan</i>	
Development of an Electronic Medical Record (EMR) in Stayed Nursing Installation.....	220
<i>Eko Handoyo, Aghus Sofwan, Mohammad Muttaqin</i>	
Development of Supporting Sales Analysis Application using Frequent Closed Constraint Gradient Mining Algorithm (FCCGM)	224
<i>Susana Limanto, Dhiani Tresna Absari</i>	
Implementation of KMS to Integrate Knowledge Management and Supply Chain Management Process	229
<i>Vivine Nurcahyawati, Retno Aulia Vinarti, Mudjahidin</i>	
Indonesian WordNet Sense Disambiguation using Cosine Similarity and Singular Value Decomposition.....	234
<i>Syandra Sari, Ruli Manurung, Mirna Adriani</i>	
Influence of Electronic Media and External Reward Towards Knowledge Sharing Management to Learning Process in Higher Education Institution	240
<i>Alexander Setiawan</i>	

Information and Technology Outsourcing Vendor Selection: An Integrative Literature Review.....	245
<i>Jimmy</i>	
Information Retrieval on MARC Metadata	251
<i>Adi Wibowo, Rolly Intan, Irawan Arifin</i>	
Learning Management Systems' Integration.....	256
<i>N.S Linawati, Putra Sastra, P.K. Sudiarta</i>	
Mining Sequential Pattern on Sequential Data of Paint Sales Transaction Flow	260
<i>Agustinus Noertjahyana, Gregorius Satia Budhi, Henny Kusumawati Wibowo</i>	
Modeling School Bus for Needy Student Using Geographic Information System.	265
<i>Daniel Hary Prasetyo, Jamilah Muhamad, Rosmadi Fauzi</i>	
Optimization SQL Server 2005 Query using Cost Model and Statistic	272
<i>Ibnu Gunawan</i>	
Spatial Autocorrelation Modelling for Determining High Risk Dengue Fever Transmission Area in Salatiga, Central Java, Indonesia	277
<i>Sri Yulianto J.P., Kristoko Dwi Hartomo, Krismiyati</i>	
Supply Chain Improvement with Design Structure Matrix Method and Clustering Analysis (A Case Study).....	281
<i>Tanti Octavia, Siana Halim, Stefanus Anugraha Lukmanto, Harvey Sutopo</i>	
The Comparation of Similarity Detection Method on Indonesian Language Document	285
<i>Anna Kurniawati, Lily Wulandari, I Wayan Simri Wicaksana</i>	
The Effects of Training Documents, Stemming, and Query Expansion in Automated Essay Scoring for Indonesian Language with VSM and LSA Methods.....	290
<i>Heninggar Septiantri, Indra Budi</i>	
The Impact of Object Ordering in Memory on Java Application Performance	296
<i>Amil A. Ilham, Kazuaki Murakami</i>	
Using Data Mining to Improve Prediction of 'No Show' Passenger on an Airline Reservation System.....	302
<i>Johan Setiawan, Bobby Limantara</i>	
Using Frequent Max Substring Technique for Thai Keyword Extraction used in Thai Text Mining	309
<i>Todsanai Chumwatana, Kok Wai Wong, Hong Xie</i>	

Using the End-User Computing Satisfaction Instrument to Measure Satisfaction with Web-Based Information Systems	315
<i>Dedi Rianto Rahadi</i>	

Imaging Technology

Batik Image Classification using Log-Gabor and Generalized Hough Transform Features	320
<i>Laksmi Rahadiani, Hadaq R. Sanabila, Ruli Manurung, Aniati Murni</i>	
Burrows Wheeler Compression Algorithm (BWCA) in Lossless Image Compression	326
<i>Elfitri Syahrul, Julien Dubois, Vincent Vajnovszki, Asep Juarna</i>	
Comparison of Random Gaussian and Partial Random Fourier Measurement in Compressive Sensing Using Iteratively Reweighted Least Squares Reconstruction	332
<i>Endra</i>	
Developing a Video Player Application for Phillips File Standard for Pictorial Data Format (NXPP): A Project View Approach	335
<i>Eko Handoyo, Restiono Djati Kusumo</i>	
Development Edge Detection Using Adhi Method, Case Study: Batik Sidomukti Motif.....	340
<i>Adhi Pranoto, Suyoto</i>	
Discriminating Cystic and Non Cystic Mass Using GLCM and GLRM-based Texture Features	346
<i>Hari Wibawanto, Adhi Susanto, Thomas Sri Widodo, S Maesadji Tjokronegoro</i>	
Fractal Terrain Generator.....	351
<i>Budi Hartanto, Monica Widiastri, Gunawan Widjaja</i>	
From Taiwan Puppet Show to Augmented Reality	356
<i>Yang Wang, Bo Ruei Huang, Zih Huei Wang</i>	
Generating Iriscode using Gabor Filter.....	362
<i>I Ketut Gede Darma Putra, Lie Jasa</i>	
Interpolation Technique to Improve Unsupervised Motion Vector Learning of Wyner-Ziv Video Coding.....	366
<i>I. M. Oka Widyantara, N.P. Sastra, D.M. Wiharta, Wirawan, G. Hendranto</i>	
Iris Segmentation and Normalization	371
<i>I Ketut Gede Darma Putra, I Nyoman Piarsa, Nazer Jawas</i>	
NEATS: A New Method for Edge Detection	377
<i>Maria Yumike, Suyoto</i>	

Online Facial Caricature Generator	383
<i>Rudy Adipranata, Stephanus Surya Jaya, Kartika Gunadi</i>	

Silny Approach to Edge Detection for Central Borneo Batik.....	387
<i>Silvia, Suyoto</i>	

Internet, Web Services & Mobile Applications

Cattle's Cost of Goods Sold System Information at CV Agriranch	392
<i>Lily Puspa Dewi, Yulia, Anita Nathania, Doddy Hartanto</i>	

Compensation Method for Internet Grids using One-to-many Bargaining	396
<i>Andreas Kurniawan, Pujiyanto Yugopuspito, Johan Muliadi Kerta</i>	

Mobile RSS Push Using Jabber Protocol.....	406
<i>Fajar Baskoro, Dwi Ardi Irawan</i>	

Teacher's Community Building Website to Facilitate Networking and Life-Long Learning.....	412
<i>Arlinah Imam Rahardjo, Yulia, Silvia Rostianingsih</i>	

Vision and Mission Educational Foundation (YPVM) Web-Based Project Management System	417
<i>Arlinah Imam Rahardjo, Yulia, Edwin</i>	

Web Based School Administration Information System on LOGOS School.....	421
<i>Djoni Haryadi Setiabudi, Ibnu Gunawan, Handoko Agung Fuandy</i>	

Communication Systems & Networks

Data Visualization of Modulated Laser Beam Communication System	427
<i>Zin May Aye</i>	

Development of Steganography Software with Least Significant Bit and Substitution Monoalphabetic Cipher Methods for Security of Message Through Image.....	432
<i>Iswar Kumbara, Erwin</i>	

Feasibility Analysis of Zigbee Protocol in Wireless Body Area Network	436
<i>Vera Suryani, Achmad Rizal</i>	

Mobile TV with RTSP Streaming Protocol and Helix Mobile Producer	439
<i>Yunianto Purnomo, Andrew Jaya Efendy</i>	

Quantitative Performance Mobile Ad-Hoc Network using Optimized Link State Routing Protocol (OLSR) and Ad-Hoc On-Demand Distance Vector (AODV).....	443
---	-----

Andreas Handojo, Justinus Andjarwirawan, Hiem Hok

Spatial Rain Rate Measurement to Simulation Colour Noise Communication Channel Modeling for Millimeter Wave In Mataram.....	449
<i>Made Sutha Yadnya, Gamantyo Hendrantoro</i>	
The Effect of Maximum Allocation Model in Differentiated Service-Aware MPLS-TE	453
<i>Bayu Erfianto</i>	
User Accounting System of Centralized Computer Networks using RADIUS Protocol	457
<i>Heru Nurwarsito, Raden Arief Setyawan, Handoko D. Fatikno</i>	
Wireless Data Communication with Frequency Hoping Spread Spectrum (FHSS) Technique.....	463
<i>Khin Swe Myint, Zarli Cho</i>	
Wireless LAN User Positioning using Location Fingerprinting and Weighted Distance Inverse	469
<i>Justinus Andjarwirawan, Silvia Rostianingsih, Charlie Anthony</i>	
WLANXCHANGE: A New Approach in Data Transfer for Mobile Phone Environment	474
<i>Ary Mazharuddin Shiddiqi, Bagus Jati Santoso, Rio Indra Maulana</i>	

Control & Automation

Analysis Influence Internal Factors on Fuzzy Type 2 Performance of Swing Phase Gait Restoration	479
<i>Hendi Wicaksono</i>	
Design and Construction of Wind Speed Indicator Based on PIC Microcontroller System	484
<i>Khin Mar Aye, Khi Tar Oo</i>	
Fault Diagnosis in Batch Chemical Process Control System using Intelligent System	489
<i>Syahril Ardi</i>	
Implementation of an Adaptive PID Controller using the SPSA Algorithm with Realistic Target Response.....	493
<i>Sofyan Tan</i>	
Induction Heating Efficiency Analysis Modeling Using COMSOL® Multiphysics Software.....	498
<i>Didi Istardi</i>	
Authors Index.....	504