

PARALLEL PROCESSING IN COMPUTE UNIFIED DEVICE ARCHITECTURE  
(CUDA) FOR ENERGY SAVING GLASS (E-GLASS)

KHOO WEN XIN

UNIVERSITY TEKNIKAL MALAYSIA MELAKA 2014

**LAMPIRAN B: BORANG PENGESAHAN STATUS TESIS**  
**BORANG PENGESAHAN STATUS TESIS\***

JUDUL: \_\_\_\_\_

SESI PENGAJIAN: \_\_\_\_\_

Saya                     KHOO WEN XIN                      
(HURUF BESAR)

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.

4. \*\* Sila tandakan (/)

\_\_\_\_\_ SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

\_\_\_\_\_ TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

\_\_\_\_\_ TIDAK TERHAD

\_\_\_\_\_  
(TANDATANGAN PENULIS)

Alamat tetap: \_\_\_\_\_

\_\_\_\_\_

Tarikh: \_\_\_\_\_

\_\_\_\_\_  
(TANDATANGAN PENYELIA)

Dr Abdul Samad Shibghatullah

Nama Penyelia

Tarikh: \_\_\_\_\_

CATATAN: \* Tesis dimaksudkan sebagai Laporan Akhir Projek Sarjana Muda (PSM)

\*\* Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

PARALLEL PROCESSING IN COMPUTE UNIFIED DEVICE ARCHITECTURE  
(CUDA) FOR ENERGY SAVING GLASS (E-GLASS)

KHOO WEN XIN

This report is submitted in partial fulfillment of the requirements for the  
Bachelor of Computer Science (Computer Networking)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY  
UNIVERSITY TEKNIKAL MALAYSIA MELAKA 2014

**DECLARATION**

I hereby declare that this project report entitled

**PARALLEL PROCESSING IN COMPUTE CUDA FOR ENERGY SAVING  
GLASS (E-GLASS)**

Is written by me and is my own effort and that no part has been plagiarized without  
citation

STUDENT : \_\_\_\_\_ DATE: \_\_\_\_\_

(KHOO WEN XIN)

SUPERVISOR : \_\_\_\_\_ DATE: \_\_\_\_\_

(DR ABDUL SAMAD BIN SHIBGATULLAH)

## **DEDICATION**

To my beloved parents Khoo Yu Boon and Ooi Chye Kee for supporting me in completing my final year project. Encouragement will never be less throughout my study and I will like to dedicate this to them.

## ACKNOWLEDGEMENTS

I would like thank my parents and my sister for giving me support and encouragement for completing my final year project. Besides I would like to thank Teoh Khai Chien for his generosity of lending me his own personal laptop in order for me to develop my whole system during the duration of my final year project 1 and 2 development. Besides, I would like to thank Dr Samad bin Shibgatullah my supervisor for helping me and guide me throughout my final year project and the consultation advices that he gave. Additional of it, I would like to thank Farah Ayuni for being my best tutor in my testing phase of designing the coating structure and the simulation software that has been used in the testing phase.

## ABSTRACT

Energy saving glass is used to keep the warmth and temperature of the building instead of using thermal radiator or machine to generate the heat all the time during winter. Yet the coating structure of the glass is mostly in regular shape such as tripoles and circular with very small size that limits the useful signal such as wireless signal and radio frequency to pass through. In order to develop a complex coating structure genetic algorithm technique is used. Yet genetic algorithm requires a fast processing speed in order to cope with the process of creating new chromosomes from the population and undergoes the selection, crossover and mutation operation processes. Hence parallel processing is needed to overcome this problem with the use of both CPU and GPU to eliminate the need of purchasing high performance CPU and the need of adding additional repeaters to increase the wireless signals. The coating structure will be presented in binary bits in a text file that shows the best chromosome. The result will then be analyzed in a simulation tool that uses to check for the signal transmission efficiency and rate loss. Besides, speed tests will be done to determine whether parallel processing can increase the speed in running the program code.

## ABSTRAK

Parallel Processing in CUDA in Energy saving glass merupakan satu system yang menggunakan teknik pemrosesan selari bagi NVIDIA kad graphic platform. Sistem ini akan menghasilkan chromosome yang kemudian akan digunakan untuk simulasi testing. Keputusan simulasi ini akan menentukan sama ada salutan design bagi saving glass ini membolehkan frequency radio untuk masuk ke dalam bangunan.



## TABLE OF CONTENTS

CHAPTER	SUBJECT	PAGE
	DECLARATION	i
	DEDICATION	ii
	ACKNOWLEDGEMENT	iii
	ABSTRACT	iv
	TABLE OF CONTENTS	vi
	LIST OF TABLES	xi
	LIST OF FIGURES	xii
CHAPTER I	INTRODUCTION	
	1.1 Project Background	1
	1.2 Problem Statement	2
	1.3 Objectives	3
	1.4 Scopes	4
	1.5 Project Significance	4
	1.6 Expected Output	5
	1.7 Conclusion	6
CHAPTER II	LITERATURE REVIEW AND PROJECT METHODOLOGY	

<b>2.1 Introduction</b>	<b>7</b>
<b>2.2 Facts and Finding</b>	<b>8</b>
<b>2.2.1 Introduction</b>	<b>8</b>
<b>2.2.2 Current Approaches with Energy Saving Glass</b>	<b>9</b>
<b>2.2.3 Genetic Algorithm as Optimization Method</b>	<b>13</b>
<b>2.2.3.1 Overview of the Genetic Algorithm</b>	<b>14</b>
<b>Techniques</b>	
<b>2.2.4 Parallel Genetic Algorithm</b>	<b>18</b>
<b>2.2.5 Graphic Processing Unit as Processing</b>	<b>22</b>
<b>Platform (Parallel Processing)</b>	
<b>2.3 Project Methodology</b>	<b>25</b>
<b>2.4 Project Requirements</b>	<b>27</b>
<b>2.4.1 Software Requirement</b>	<b>27</b>
<b>2.4.2 Hardware Requirement</b>	<b>27</b>
<b>2.5 Project Schedule and Milestone</b>	<b>27</b>
<b>2.5.1 Stages of activities in each process</b>	<b>27</b>
<b>2.5.1.1 Planning stage</b>	<b>28</b>
<b>2.5.1.2 Analyzing stage</b>	<b>28</b>
<b>2.5.1.3 Implementing stage</b>	<b>29</b>
<b>2.5.1.4 Testing stage</b>	<b>29</b>
<b>2.5.1.5 Closing stage</b>	<b>29</b>

	<b>2.6 Conclusion</b>	<b>33</b>
<b>CHAPTER III</b>	<b>ANALYSIS</b>	
	<b>3.1 Introduction</b>	<b>34</b>
	<b>3.2 Problem Analysis</b>	<b>35</b>
	<b>3.2.1 Identified Problem</b>	<b>38</b>
	<b>3.3 Requirement Analysis</b>	<b>39</b>
	<b>3.3.1 Software Requirement</b>	<b>39</b>
	<b>3.3.2 Hardware Requirement</b>	<b>40</b>
	<b>3.4 Conclusion</b>	<b>41</b>
<b>CHAPTER IV</b>	<b>DESIGN</b>	
	<b>4.1 Introduction</b>	<b>42</b>
	<b>4.2 High Level Design</b>	<b>43</b>
	<b>4.2.1 System Architecture</b>	<b>55</b>
	<b>4.2.2 Output Design</b>	<b>56</b>
	<b>4.3 Detailed Design</b>	<b>57</b>
	<b>4.3.1 Software Specification</b>	<b>58</b>
	<b>4.4 Conclusion</b>	<b>62</b>
<b>CHAPTER V</b>	<b>IMPLEMENTATION</b>	
	<b>5.1 Introduction</b>	<b>63</b>

<b>5.2 Software Configuration Management</b>	<b>63</b>
<b>5.2.1 Configuration Setup</b>	<b>63</b>
<b>5.2.2 Version Control Procedure</b>	<b>72</b>
<b>5.2.2.1 Formal Control</b>	<b>72</b>
<b>5.2.2.2 Informal Control</b>	<b>73</b>
<b>5.3 Implementation Status</b>	<b>74</b>
<b>5.4 Conclusion</b>	<b>77</b>
<b>CHAPTER VI TESTING</b>	
<b>6.1 Introduction</b>	<b>78</b>
<b>6.2 Test Plan</b>	<b>78</b>
<b>6.2.1 Test Organization</b>	<b>78</b>
<b>6.2.2 Test Environment</b>	<b>79</b>
<b>6.2.2.1 Location /Environment</b>	<b>79</b>
<b>6.2.2.2 Hardware/Software Configuration</b>	<b>79</b>
<b>6.2.2.2.1 Configuration and Preparation</b>	<b>79</b>
<b>6.2.2.2.2 Training Prior to Testing</b>	<b>80</b>
<b>6.2.3 Test Schedule</b>	<b>80</b>
<b>6.3 Test Strategy</b>	<b>80</b>
<b>6.3.1 Classes of Tests</b>	<b>80</b>
<b>6.4 Test Design</b>	<b>81</b>

<b>6.4.1 Test Description</b>	<b>81</b>
<b>6.4.2 Test Data</b>	<b>81</b>
<b>6.4.2.1 Speed Test</b>	<b>81</b>
<b>6.4.2.2 White box and Black box Testing</b>	<b>85</b>
<b>6.4.2.2.1 Black box Testing</b>	<b>85</b>
<b>6.4.2.2.2 White box Testing</b>	<b>88</b>
<b>6.4.2.3 CST Testing</b>	<b>89</b>
<b>6.5 Test Results and Analysis</b>	<b>95</b>
<b>6.6 Conclusion</b>	<b>98</b>
<b>CHAPTER VII</b>	<b>PROJECT CONCLUSION</b>
<b>7.1 Observation on Weakness and Strength</b>	<b>99</b>
<b>7.2 Proposition for Improvement</b>	<b>100</b>
<b>7.3 Contribution</b>	<b>100</b>
<b>7.4 Conclusion</b>	<b>101</b>
<b>REFERENCES</b>	<b>102</b>
<b>APPENDICES</b>	<b>104</b>

**LIST OF TABLES**

<b>TABLE</b>	<b>TITLE</b>	<b>PAGE</b>
<b>1</b>	<b>Speed Test</b>	<b>92</b>
<b>2</b>	<b>Black box Testing</b>	<b>93</b>
<b>3</b>	<b>White box Testing</b>	<b>93</b>
<b>4</b>	<b>CST Testing</b>	<b>94</b>

## LIST OF FIGURES

<b>DIAGRAM</b>	<b>TITLE</b>	<b>PAGE</b>
1.1	<b>Irregular Shapes Of Existing Coating</b>	<b>9</b>
1.2	<b>The Double Regular Shapes</b>	<b>10</b>
1.3	<b>Double Rectangular Design Shape</b>	<b>10</b>
1.4	<b>A Cross Dipole Shape Dimension</b>	<b>11</b>
1.5	<b>A Lowpass And Highpass Rectangular Coated Design Shape</b>	<b>12</b>
1.6	<b>Pre- Existing FSS Design Generated By Periodic Method Of Moment</b>	<b>12</b>
1.7	<b>The Double Regular Shapes</b>	<b>13</b>
1.8	<b>Various Popular Crossover Diagrams</b>	<b>17</b>
1.9	<b>Mutation Operations after Crossover Operation</b>	<b>18</b>
1.10	<b>A Schematic of A Master-Slave Parallel GA</b>	<b>20</b>
1.11	<b>Coarse-grained Parallel Genetic Algorithms</b>	<b>21</b>
1.12	<b>Architecture of GPU</b>	<b>24</b>
1.13	<b>Architecture of CUDA</b>	<b>25</b>
1.14	<b>Top-Down Approach Process and Activities</b>	<b>30</b>
1.15	<b>Gantt Chart Activities</b>	<b>31</b>
1.16	<b>Milestones base on Gantt Chart Activities</b>	<b>32</b>

<b>2.1</b>	<b>General Process Flow Chart</b>	<b>36</b>
<b>2.2</b>	<b>Main Flow Chart</b>	<b>37</b>
<b>3.1</b>	<b>High Level Design</b>	<b>43</b>
<b>3.2</b>	<b>General Flow Chart</b>	<b>44</b>
<b>3.3</b>	<b>Generate Chromosome Flow Chart</b>	<b>45</b>
<b>3.4</b>	<b>Generate Crossover Flow Chart</b>	<b>46</b>
<b>3.5</b>	<b>Mutation Flow Chart</b>	<b>47</b>
<b>3.6</b>	<b>Data Flow Diagram Level 0</b>	<b>48</b>
<b>3.7</b>	<b>Data Flow Diagram Level 1</b>	<b>49</b>
<b>3.8</b>	<b>Data Flow Diagram Level 2</b>	<b>50</b>
<b>3.9</b>	<b>Data Flow Diagram Level 3</b>	<b>50</b>
<b>3.10</b>	<b>Data Flow Diagram Level 4</b>	<b>51</b>
<b>3.11</b>	<b>Data Flow Diagram Level 5</b>	<b>52</b>
<b>3.12</b>	<b>Data Flow Diagram Level 6</b>	<b>53</b>
<b>3.13</b>	<b>Data Flow Diagram Level 7</b>	<b>54</b>
<b>3.14</b>	<b>System Architecture</b>	<b>55</b>
<b>3.15</b>	<b>Example of Output (Text File)</b>	<b>56</b>
<b>3.16</b>	<b>Detailed Design</b>	<b>57</b>
<b>3.17</b>	<b>Software Specification</b>	<b>58</b>
<b>3.18</b>	<b>Example of Output</b>	<b>59</b>
<b>3.19</b>	<b>Example of Output in 2GA.txt</b>	<b>60</b>



<b>3.20</b>	<b>Example of Output in Best Chromosome.txt</b>	<b>61</b>
<b>4.1</b>	<b>Verify Installation and Version Model</b>	<b>64</b>
<b>4.2</b>	<b>Verify Bandwidth Test</b>	<b>65</b>
<b>4.3</b>	<b>Copy Important Program File</b>	<b>65</b>
<b>4.4</b>	<b>Edit CUDA 5.5.props</b>	<b>66</b>
<b>4.5</b>	<b>Edit CUDA 5.5.targets</b>	<b>67</b>
<b>4.6</b>	<b>Edit CUDA 5.5.target</b>	<b>67</b>
<b>4.7</b>	<b>Edit CUDA 5.5.target</b>	<b>68</b>
<b>4.8</b>	<b>Edit CUDA 5.5.target</b>	<b>68</b>
<b>4.9</b>	<b>Place the Copy Files in MSBUILD Folder</b>	<b>69</b>
<b>4.10</b>	<b>Add cu Extension</b>	<b>70</b>
<b>4.11</b>	<b>Change Supported Version to 1700</b>	<b>70</b>
<b>4.12</b>	<b>Select Console Application</b>	<b>71</b>
<b>4.13</b>	<b>Select Build Dependencies</b>	<b>71</b>
<b>4.14</b>	<b>Select CUDA 5.5(targets, .props)</b>	<b>72</b>
<b>4.15</b>	<b>Calculate Fitness Function</b>	<b>74</b>
<b>4.16</b>	<b>Crossover Function</b>	<b>75</b>
<b>4.17</b>	<b>Mutation Function</b>	<b>76</b>
<b>4.18</b>	<b>Selection Function</b>	<b>76</b>
<b>5.1</b>	<b>Speed Test with 5000 Population Sizes</b>	<b>82</b>
<b>5.2</b>	<b>Speed test with 5000 Population Sizes Timeline</b>	<b>82</b>

<b>5.3</b>	<b>Speed test with 1000 Population Sizes</b>	<b>83</b>
<b>5.4</b>	<b>Speed test with 100 Population Sizes</b>	<b>83</b>
<b>5.5</b>	<b>Speed test with 20 Population Sizes</b>	<b>84</b>
<b>5.6</b>	<b>Speed test with 20 Population Sizes Timeline</b>	<b>84</b>
<b>5.7</b>	<b>Input or Predefine value for CHRO_LENGTH</b>	<b>85</b>
<b>5.8</b>	<b>Output Value for CHRO_LENGTH with 100 Binary Bits</b>	<b>85</b>
<b>5.9</b>	<b>Predefine Input as Binary Number Using Bool</b>	<b>86</b>
<b>5.10</b>	<b>Output in Binary Bits</b>	<b>86</b>
<b>5.11</b>	<b>Output in Binary Bits</b>	<b>87</b>
<b>5.12</b>	<b>Output of Population Size</b>	<b>87</b>
<b>5.13</b>	<b>Correct Fitness Value</b>	<b>88</b>
<b>5.14</b>	<b>Fitness Value of 46 Instead of 47</b>	<b>89</b>
<b>5.15</b>	<b>Fitness Value of 40 Instead of 42</b>	<b>89</b>
<b>5.16</b>	<b>Fitness Value of 53 Instead of 50</b>	<b>89</b>
<b>5.17</b>	<b>Best Chromosome with Fitness Value 53</b>	<b>89</b>
<b>5.18</b>	<b>Coating Structure of 100Bits with 0.5 mm Of Glass Width</b>	<b>90</b>
<b>5.19</b>	<b>Simulation Result of 100Bits with 0.5 mm Of Glass Width</b>	<b>90</b>
<b>5.20</b>	<b>Coating Structure of 100Bits with 2 mm Of Glass Width</b>	<b>91</b>
<b>5.21</b>	<b>Simulation Result of 100Bits with 2 mm</b>	<b>91</b>

	<b>Of Glass Width</b>	
<b>5.22</b>	<b>Coating Structure of 400Bits with 0.5 mm Of Glass Width</b>	<b>92</b>
<b>5.23</b>	<b>Simulation Result of 400Bits with 0.5 mm Of Glass Width</b>	<b>92</b>
<b>5.24</b>	<b>Coating Structure of 400Bits with 0.1 mm Of Glass Width</b>	<b>93</b>
<b>5.25</b>	<b>Simulation Result of 400Bits with 0.1 mm Of Glass Width</b>	<b>93</b>
<b>5.26</b>	<b>Coating Structure of 400Bits with 2 mm Of Glass Width</b>	<b>94</b>
<b>5.27</b>	<b>Simulation Result of 400Bits with 2 mm Of Glass Width</b>	<b>94</b>
<b>6.1</b>	<b>Installation of NVIDIA latest driver, Microsoft Visual Studio Professional 2012 and CUDA Document Toolkit 5.5</b>	<b>104</b>
<b>6.2</b>	<b>Create New CUDA Project</b>	<b>105</b>
<b>6.3</b>	<b>Compile Program</b>	<b>106</b>
<b>6.4</b>	<b>Debug Program</b>	<b>107</b>
<b>6.5</b>	<b>Result after Debug</b>	<b>107</b>
<b>6.6</b>	<b>Result in Text File</b>	<b>108</b>
<b>6.7</b>	<b>Result in Text File 111GA.txt</b>	<b>108</b>
<b>6.8</b>	<b>Result in Text File Best Chromosome.txt</b>	<b>109</b>

## CHAPTER I

### INTRODUCTION

#### 1.1 Project Background

E-saving glass is benefit in keeping the temperature of the room or it keeps the warmth of certain place. It has it special coating structure that can prevent the heat energy from being release out. Besides, it reflects the incoming UV ray from being penetrate into the building or vehicles, which more or less act like a filter but in a form of fabrication instead of plastic.

The problem occur when the coating of the glass actually limits the signals such as infrared, radio wave (WIFI) from getting into the room due to the coating of the glass, as most of the useful signal cannot penetrate into the glass that causes the loss of useful signal in the room which can limits the bandwidth of the signal. Besides, the coating structure on the glass is varies and mostly are in regular shapes.

In order to generate complex and irregular coating structure, it may be a difficulty in implementation as the techniques that have been use require reaching an ideal result. By

using the parallel Genetic algorithm technique, it is able to create new chromosome, and develop new population by using the chromosome that have created.

The purpose of using parallel genetic algorithm as can provide faster speed in generate and execute the coding which, will then using the GPU(Graphic Processor Unit) together with CPU(Center Processing Unit) to execute the processes; as executing genetic algorithm command and codes requires longer time to process due to the creation of the new chromosome and population. Hence, the project will be focus in using CUDA programming language, the language that use in NVIDIA graphic card. By executing the process using both GPU and CPU will speed up the processing time more than usual as two processes can be ran at the same time which will decrease the opportunity in having bottle neck and reduce the cost of purchasing high processing speed computer or devices. After the development of coating structure, it will then tested by Computer Simulation Tool (CST) to see the end result of the coating structure and decide which will be an ideal coating structure for the glass.

## 1.2 Problem Statement

The problem occurs when the coating of the glass actually limits the signals such as infrared, radio wave (WIFI) from getting into the room due to the coating of the glass, as most of the useful signal cannot penetrate into the glass that causes the loss of useful signal in the room which can limits the bandwidth of the signal. Besides, the current coating structure on the glass is varies and mostly are in regular shapes. In order to generate complex and irregular coating structure, it may be a difficulty in implementation as the techniques that have been use require reaching an ideal result.

Hence, parallel genetic algorithm method is chosen, as it can generate complex structure of the glass coating besides it can select the best coating chromosome to develop new population without affecting other population. Here is the problem again when we are using parallel genetic algorithm, as we know that using parallel GA may require fast processing time and speed for the processor in order to execute and run the processes and commands that generate by the creating of new coating chromosomes and population, unless we have more than one processor also know as central processing unit. In order to have more than one processor, we may need a very high-end technology computer to

process the processes threads. Hence, we came up with using graphic processing unit (GPU) also known as the graphic card that can run the processes together with the CPU that provide faster speed in processing the processes generate by the parallel GA. Besides, different types of GPU have different types of coding method that can be used.

### 1.3 Objective

- i. To develop new coating structure using parallel genetic algorithm

Parallel genetic algorithm enables to develop new chromosome and population by using selection, mutation or crossover method. All these method enable to develop irregular shape of coating structure. Parallel genetic algorithm enable fast processing and executing of the command as it use both CPU and GPU to process the processes that can save time and provide faster speed.

- ii. To decrease the chances of useful signal from being lost

Based on the current coating structure of E-glass, it has filter most of the useful signal from getting into the building by reflecting them. Hence the chances of useful signals that can go into the building have been reflected away or loss that has limits the bandwidth.

- iii. Parallel Processing using CPU and GPU

Since the development of the parallel genetic algorithm require the need to generate new chromosome and population in order to create the complex coating structure before it can be tested as an optimal coating structure. Yet, this step requires high speed processor in order to generate the chromosome and the best fitness function. Hence, with the use of GPU together with CPU can generate a very fast processing time in executing the codes base on our GPU capacity which most of the laptop nowadays already equipped with it. That is one of the ways to reduce the cost in purchasing high speed processor.

## 1.4 Scopes

- i. Parallel processing using GPU and CPU.

Parallel genetic algorithm requires faster speed to execute the process; hence the use of GPU together with CPU to process the processes can speed up the processing time. CUDA (only for NVIDIA graphic card) programming language uses C++ as the coding technique.

- ii. Develop a new coating structure for E-glass saving

To enhance the current coating structure by developing a new coating structure that is complex and irregular in shape by using parallel genetic algorithm that enable to create new coating chromosome and develop new population without affecting the current population.

- iii. Using Computer Simulation tool (CST) to determine most suitable coating structure.

The coating structures need to be tested first before it can be determine to be the most ideal coating structure. By using the CST, we can determine the coating structure suitability base on the result that has been generated.

## 1.5 Project Significance

This project will beneficial the society nowadays as most of the users is demanding on the wireless technology such as WIFI that uses the radio frequency. By allowing more useful signal from getting through the building, the users or employees can increase their work efficiency and will reduce the cost of implementing more base station to have a better signals or bandwidth.

## 1.6 Expected Output

This project will help to overcome the loss of useful signal inside the building by developing the coating structure of the glass that used to overcome the signal that pass through the building. Besides, with the use parallel processing that used to run the genetic algorithm function create a new platform in reducing the cost of buying a high performance Central Processing Unit (CPU) as both CPU and GPU can run the process parallel in order to execute the process which consider to be a win-win solution.