A Support Vector Machine Classification of Computational Capabilities of 3D Map on Mobile Device for Navigation Aid

ARTICLE *in* INTERNATIONAL JOURNAL OF INTERACTIVE MOBILE TECHNOLOGIES (IJIM) · NOVEMBER 2015

DOI: 10.3991/2315.1111.333

9 AUTHORS, INCLUDING:

Teddy Mantoro

Sampoerna University

Ahmad Waqas

Sukkur Institute of Business Administration

Shafi'i M Abdulhamid

Federal University of Technology Minna

Abdulsalam Ya'u Gital

Association for Computing Machinery

ACCEPTED FOR PUBLICATION IN INTERNATIONAL JOURNAL OF INTERACTIVE MOBILE TECHNOLOGIES

A Support Vector Machine Classification of Computational Capabilities of 3D Map on Mobile Device for Navigation Aid

doi:10.3991/ijxx.vxnx.xxx

Adamu Abubakar[1], Teddy Mantoro[2], Sardjoeni Moedjiono[3], Haruna Chiroma[4], Ahmad Waqas[1], Shafi'i Muhammad Abdulhamid[5], Mukhtar Fatihu Hamza[6], Abdulsalam Ya'u Gital[7]
[1]Department of Information Systems, International Islamic University Malaysia, Kuala Lumpur
[2]Faculty of Science and Technology, Sampoerna University, Jakarta, Indonesia
[3]Budi Luhur University, Computer Science Post Graduate Program, Jakarta
[4]Department of Artificial Intelligence, University of Malaya, 50603 Pantai Valley, Kuala Lumpur
[5]Faculty of Computing, Universiti Teknologi Malaysia, Johor Bahru Malaysia.
[6]Department of Mechatronics Engineering, Bayero University, 3011, Kano, Nigeria
[7] Mathematical Science Department, Abubakar Tafawa Balewa University Bauchi, Nigeria

***Abstract*—3D map for mobile devices provide more realistic view of an environment and serves as better navigation aid. Previous research studies shows differences in 3D maps effect on acquiring of spatial knowledge. This is attributed to the differences in mobile device computational capabilities. Crucial to this is the time it takes for 3D map dataset to be rendered for a required complete navigation task. Different findings suggest different approaches on solving the problem of time required for both in-core (inside mobile) and out-core (remote) rendering of 3D dataset. Unfortunately, studies on analytical techniques required to show the impact of computational resources required for the use of 3D map on mobile devices were neglected by the research communities. This paper uses Support Vector Machine (SVM) to analytically classified mobile device computational capabilities required for 3D map that will be suitable for use as navigation aid. Fifty different Smart phones were categorized on the bases of their Graphical Processing Unit (GPU), display resolution, memory and size. The result of the proposed classification shows high accuracy.**

***Index Terms*—3D Map, Graphical Processing Unit, Support Vector Machine. In-core rendering, out-of-core rendering.**

I. INTRODUCTION

Mobile devices, especially smart phones are now able to render 3D map which provide more realistic view of an environment and serves as better navigation aid [1]. The main advantage of 3D map is enhancing the visualization quality and when use as a navigation aid improves navigation practices [2]. The current mobile devices resources combined with an increasing wireless networking capabilities and Global Positioning System (GPS) receiver, offers an opportunity for navigation [3]. There is a need for computing communities to take advantage of these properties in order to improve mobile services and enhance special knowledge.

There are many research studies which shows the positive impact of 3D maps on acquiring of spatial knowledge [4-5]. Some studies also reveals that 3D map for navigation aid is directly associated with mobile device computational capabilities, mobile device physical structures and the practices of navigation task in both physical and virtual world [4, 6-9]. Crucial to this, is the time it takes for 3D map dataset to be rendered for a required complete navigation task [4]. Different findings suggest different approaches on solving the problem of time required for both in-core (inside mobile) and out-core (remote) rendering of 3D dataset [4, 7-8]. The majority also added that visualization is a major control variable needed to solve navigation tasks with the aid of 3D map [1-9].

Unfortunately, analytical durations for both in-core and out-core rendering with respect to visualizations quality of 3D map for navigation were neglected. Analytical model construction and study of pattern of data required for 3D map visualization on mobile devices for navigation were mostly jump to a conclusion [4, 7-8]. Most studies were agree that the more the computational resources the better for 3D map visualization for navigation aid [4-9].

This paper utilize support vector machine (SVM) algorithm to map mobile devices resources required for navigation practice while using 3D map. The reason of using the algorithm is because it imitates the real life processes of the demarcating two or more elements for optimization in order to have the best solution for understanding required computational resources for 3D map suitable for the mobile device navigation aid.

II. RELATED WORK

Navigations aided devices help people to find wherever they are going. Although there are a lot of ways for providing guides to reach to any location. The most common approach used is sign indicator, indicating names of streets, or an important structure or sometimes even landmarks. This information when combine with paper 2D map helps in identifying places to a reasonable degree. The use of navigation aided devices like in-car GPS navigation devices such as Garmin or TomTom's have greatly gain a huge support [4]. Such tools typically use 2D and 3D projection of upcoming road layout, without substantial pictorial realism, and without freedom for the user to substantially manipulate the viewpoint [10]. There are many other examples of systems that apply different constraints and presentation means for the purpose of map navigation [1-4]. Such as the pictorial realism of satellite imagery or 3D view projections regardless of the means for portrayal and view control [9], others combine 2D map [4]. All these approaches intends to help circumvent the drawback of the paper map that need further description of symbols or legend [4].

Technical problems associated with a navigation aided device having an on-board Global Positioning System (GPS) were addressed in [4, 7-8]. Unfortunately problems with navigation practices are mostly evaluated subjectively, for instance, route choice and crossing behavior of pedestrian. This was evaluated in urban areas were by subjective assessment, where it assumes that pedestrians start from a network traversing to another network by selecting consecutive links of choice, then draws to a conclusion [10]. Similarly, tactical model for route choice behavior of pedestrian's travel time in multidirectional flows are presented in an observational experiment [11]. Navigation practices associated with navigation aided devices require knowledge of spatial orientation and wayfinding [12]. This can be modelled, designed and implemented in any navigation aided system. The steps require acquiring real life data of pedestrian flow to give a proof of concept, similar to the study performed in [13]. Thereafter perform an analysis of rendering speed (for in-core and out-of-core rendering) and download rate or (out-of-core rendering) of 3D map similar to the study performed in [14]. Finally, Implementations of the designed model could go for the testing, similar to the study performed in [4].

Despite that research studies are involved in drawing positive impact of 3D maps to spatial knowledge, mobile device computational capabilities, mobile device physical properties and navigation task [4-9]. Yet, there is a lack of analytical approaches that utilized 3D map dataset for either, to investigate time require for both in-core and out-core rendering of 3D dataset or visualization. In general there is a need for meaningful prediction of an approximate computational resource of 3D map required by mobile device for navigation aid. This is because 3D map view in mobile devices possesses an invaluable potential in providing a higher level of perception of a certain environment, where real-world entities could be recognized easily

III. RESEARCH METHODOLOGY

A. Support Vector Machine

The SVM technique earlier presented to deals with problems in classification, regression analysis and forecasting under linear supervised learning [15]. It involves classifying two or more classes of samples sets within the training set by producing the optimal hyperplane between them. The optimal hyperplane is one that has the maximum distance between the two sample classes in the model [16]. Training the SVM is equivalent to linear constrained quadratic programming problem which translate to exceptional and global optimum. The optimum solution obtained by SVM for any given problem is highly associated on support vectors [17]. SVM has demonstrated acceptable performance when compared with other methods, especially when used for classification [18]. Figure 1 shows how SVM works.

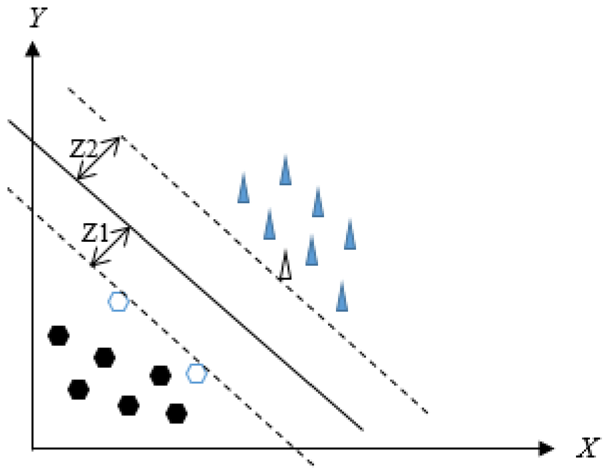

Figure 1. Support Vector Machine

Suppose from Figure 1 above, that two sets of samples classes are given, (black sample and blue sample) within x, y plane and are required to be classified using SVM. Primarily SVM will design a hyperplane between (class black and class blue) as the classifies for all training vectors in two classes by a linear discriminant function

$$g(\vec{x}) = \vec{\omega}..\vec{x}_i + b. \quad (1)$$

Here $\vec{x}$ represents the input in feature vector and $\vec{\omega}$ is the weight vector and b is a constant. Thus it is a linear function in a two dimensional space. In the figure the two dashed lines and one solid line represent the hyperplanes which can classify correctly all the instances in the sample set, but the best choice will be the hyperplane that leaves the maximum margin from both classes. The best margin is the distance between the hyperplane and the closest sample from the hyperplane (z1 and z2). The hyperplane defined by equation 1, if it is

$$g(\vec{x}) \geq 1, \quad \forall \vec{x} \in \; class\ black$$

then it will deliver values greater than 1 for all the input vectors which belongs to class black and if it is

$$g(\vec{x}) \leq -1, \quad \forall \vec{x} \in \; class\ blue$$

means it can deliver values smaller than -1 for all values which belong to class blue. This implies that for every training samples belonging to the sample class black starting from the initial value $\vec{\omega}$ and b if it is greater than zero it satisfies, or modified $\vec{\omega}$ and b in such a way that the position of the hyperplane is moved to the positive side of the hyperplane. Thus for every vector, there is y_i such that $y_i = +1$ for positive sample and $y_i = -1$ for negative sample. The distances (z1 and z2) to the closest elements will be at least 1, the modulus is 1, since the aim is to maximize the distance from the hyperplane separating the boundary from each of the feature vector, then setting a margin α as the measure of distance of the separating hyperplane, and considering equation from the geometry, the distance between a point and a hyperplane, is computed by equation 2 as;

$$\frac{\vec{w}.\vec{x}_i + b}{\|\vec{\omega}\|} \geq \alpha \qquad (2)$$

The decision is independent of the scaling factor applied to $\vec{\omega}$, thus by scaling equation 2 will yield equation 3

$$\vec{w}.\vec{x}_i + b \geq \alpha.\|\vec{\omega}\| \qquad (3)$$

As a matter of scaling $\alpha.\|\vec{\omega}\|$ can be equal 1, therefore equation 3 becomes equation 4

$$\vec{w}.\vec{x}_i + b \geq 1 \qquad (4)$$

In this case for every vector there is a condition that

$$\vec{w}.\vec{x}_i + b \geq 1 \text{ if } \vec{x} \in class\ black$$

$$\vec{w}.\vec{x}_i + b \leq 1 \text{ if } \vec{x} \in class\ blue$$

Therefore the total margin which correspond the hyperplane will be computed by this equation 5.

$$\frac{1}{|\vec{\omega}|} + \frac{1}{|\vec{\omega}|} = \frac{2}{|\vec{\omega}|} \qquad (5)$$

To extract the support vectors from the training vectors, hyperplane more reliable should be in a position where both sides' features vectors are greatly influence by the support vectors. $\vec{x}$ will be the support vectors when equation 6 is equal to 1.

$$y_i\left(\vec{w}.\vec{x}_i + b\right) = 1, \qquad (6)$$

Therefore from equation 2, the margin have to be as maximum as possible over the distances to get the support vector. This can be done by minimizing $\vec{\omega}$ and maximizing $\vec{w}.\vec{x}_i + b$. Thus, the aim of minimization here is to maximize the margin that will split the two classes equally. Minimizing the weight vector $\vec{\omega}$ in equation 2, leads to a nonlinear optimization task, which can be solved by the Karush-Kuhn-Tucker (KKT) conditions, using Langrange multipliers. However, using equation 6 as the constraint for support vector, then the weight vector $\vec{\omega}$ can be minimize. Since its a constraint optimization problem, it can then be converted in to unconstrained problem by using Langrange multipliers λ_i in equation 7 and then optimize;

$$L(\vec{\omega}, b) = \frac{1}{2}(\vec{\omega}.\vec{\omega}) - \sum \lambda_i \left[y_i \left[\vec{\omega}.\vec{x}_i + b \right] - 1 \right] \qquad (7)$$

The optimization can be done by taking the derivative of Langrange multipliers with respect to $\vec{\omega}$ and b.
Therefore taking the derivative of Langrange multipliers with respect to the weight vector $\vec{\omega}$ is

$$\vec{\omega} = \sum_{i=1}^{N} \lambda_i . y_i . \vec{x}_i \qquad (8)$$

Here N represent the number of training samples. Thus taking the derivative of Langrange multipliers with respect to b gives

$$\sum_{i=1}^{N} \lambda_i y_i = 0 \qquad (9)$$

Here N represent the number of feature vectors which are given for designing the classifiers. Putting equation 8 into equation 7, gives the Langrange multipliers expression as in equation 10.

$$\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum \lambda_i \lambda_j y_i y_j \left(\vec{x}_i . \vec{x}_j \right) \qquad (10)$$

In order to design the Langrange multipliers expression have to be maximize with difference values of λ_i, which are the Langrange multipliers and they have to be positive. For this the first constraint suitable will be $\lambda_i \geq 0$ and the other constraint is equation 9. Thus if Langrange multipliers $\lambda_i = 0$ then it indicate that the corresponding training feature vector $\vec{x}_i$ is not a support vector. Whereas if Langrange multipliers λ_i is very high it indicates that the corresponding training feature vector $\vec{x}_i$ has a high influence over the decision of the boundary hyperplane. Otherwise it might be affected by outliers and becomes extraordinarily very high.

Using equation 8, the classification decision will be for an unknown z where $d(z)$ will be the computation of $\omega.z$ through the sign expression $\left(\sum_{j=1}^{N} \lambda_j y_j \vec{x}_j . z + b \right)$. If the sign is positive, then it will be classified to class black otherwise class blue. Therefore the value of b will be obtained by equation (11)

$$\frac{1}{2}\left[\begin{array}{l} \min\left(\sum \lambda_i y_i \left(\vec{x}_i .. \vec{x}_j \right) \right) \left(i \middle| y_i = +1 \right) + \\ \max\left(\sum \lambda_i y_i \left(\vec{x}_i .. \vec{x}_j \right) \right) \left(i \middle| y_i = -1 \right) \end{array} \right] \qquad (11)$$

One of the key features of support vector machine is its ability to let $\vec{\omega}$ in equation 8 and b in equation 11 into the sign expression $\left(\sum_{j=1}^{N} \lambda_j y_j \vec{x}_j . z + b \right)$ for classification of unknown feature vector set. Where there is multiple class problems, then there is a need for multiple number of support vector machine [19].

B. Dataset and Preprocessing

This research consider to use modern smartphones available in the market. The key feature for selecting a smartphone for this work is programmable GPU unit, which is responsible for graphic rendering, although many smartphones are with low powered mobile GPUs. This is a signal that reflect poor volumetric data visualization. Although, high powered GPUs might still be affected by poor visualization when the amount of 3D mesh is too large to obtain sufficient frame rate on mobile device and also if the rendering is on mobile low-power devices [20]. Consistence to this issues, this research utilized fifty smartphones and proposes an optimal classification of the resources required for rendering 3D map for both in-core and out-of-core to mobile devices. With due consideration to [21] which report that "recent advances in sensing and software technologies enable us to obtain large-scale, yet fine 3D mesh models (nearly 2 × 107 input triangles)", the smartphones sizes, resolution and memory are the constraint function and GPU is the objective function. Thus, some samples of 7 smartphones out of 50 collected for this research are presented in Table 1

TABLE I.
SMARTPHONE VISUALIZATION PROPERTIES

Smart-Phone	Size (inch)	Resolution (pixel)	GPU (MHz)	Memory
HTC One X +	4.7	720 x 1280	520	1GB
iPhone 6 Plus	5.5	1080 x 1920	475	1GB
Samsung Galaxy S6 edge+	5.7	1440 x 2560	772	4GB
LG Nexus 5	4.95	1080 x 1920	450	2GB
Nokia Lumia 730 Dual SIM	4.7	720 x 1280	450	1GB
Sony Xperia Z3+	5.2	1080 x 1920	450	3GB
BlackBerry Z10	4.2	768 x 1280	400	2GB

The 50 samples were further preprocessed. The sizes attribute are in inches, therefore it was left on its numeric values. The resolutions are the display pixels relative to the screen sizes of the smartphones. These values were calculated by taken the product for each smartphone. The GPU values were acquired by their clock speed values, similar to size attributes, and this is also left since it is numeric. Finally the memory is the internal memory of each smartphone used. The distribution of parameters were calculated by normality test (see Figure 2).

The screen sizes dataset attend a good normality distribution when compared to the rest. The majority of smartphone screen sizes ranges from 4 to 6 inches. However, there is a huge disparity in the distribution of memory and resolution. GPU distributions are relatively good. This distributions suggest a variability among the key computational resources required for 3D map. Therefore this research propose classification of the computational resources based on GPU. Hypothetically, high powered programmable GPU units are directly associated with high resolution, and memory. The result of testing this relationship will have a huge impact on screen size of a mobile devices with 3D map for navigation aid. Currently, 3D map were mostly on smartphones with different screen size ranges from small, medium to large. Small sizes ranges from (3 - 4.9) inches, medium (4.5 - 5.9) inches and big size is above 6 inches, based on findings from [22-23]. Therefore, the normality of the dateset on this classification are presented in Figure 3, 4 and 5.

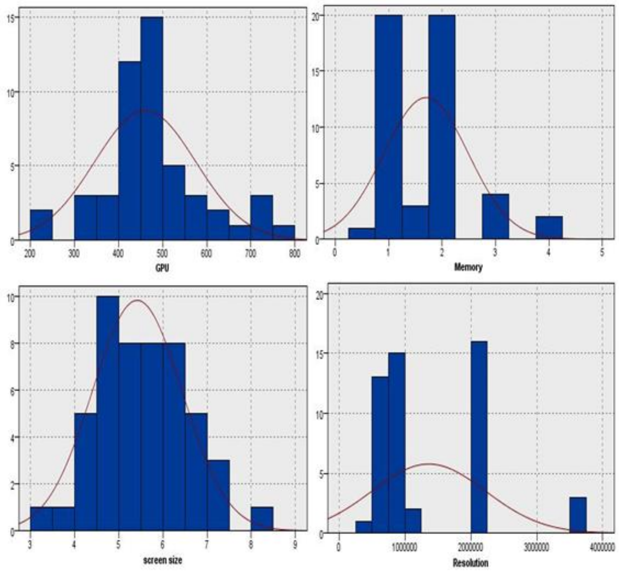

Figure 2. The distribution curve of experiment parameters, the first at the top left is for GPU, by the top left is for Memory, bottom left is Screen size and finally bottom right is Resolution

There are some deviation to the normal curve among the three GPU classification used (see Figure 3). The medium to low size GPU were greater than the high powered GPU size. However the gaps between them are found to be wider compared to memory (see Figure 4). That indicate the majority of the memory sizes were relatively the same. In terms of screen size (see Figure 5), there are different ranges which are of almost the same in size with little differences. The classes of resolutions show that the majority of the devices were in low resolution class (see Figure 6) whereas big and medium resolutions class are very low.

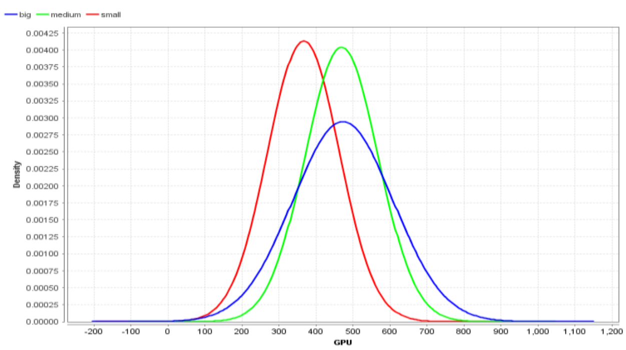

Figure 3. GPU Classification over the samples of the smartphones devices

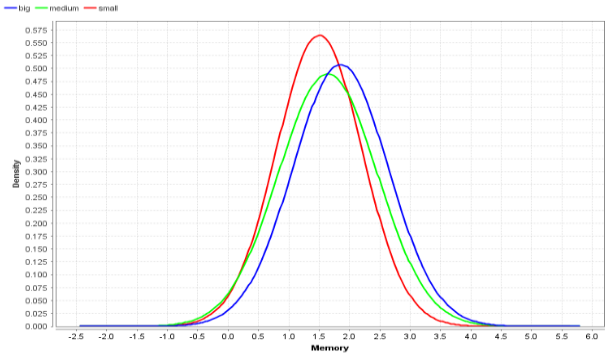

Figure 4. Memory classification over the samples of the smartphones

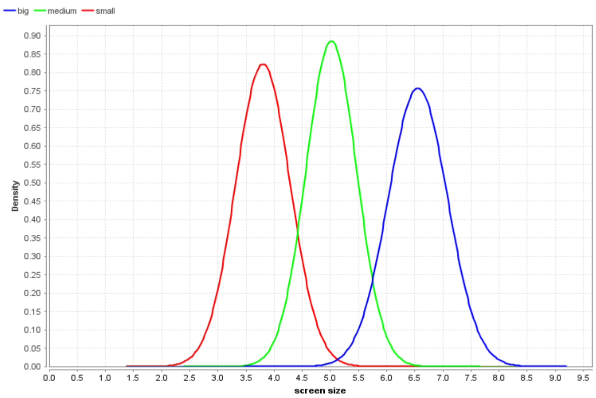

Figure 5. Screen size classification over the samples of (Small, Medium and Large) smartphones

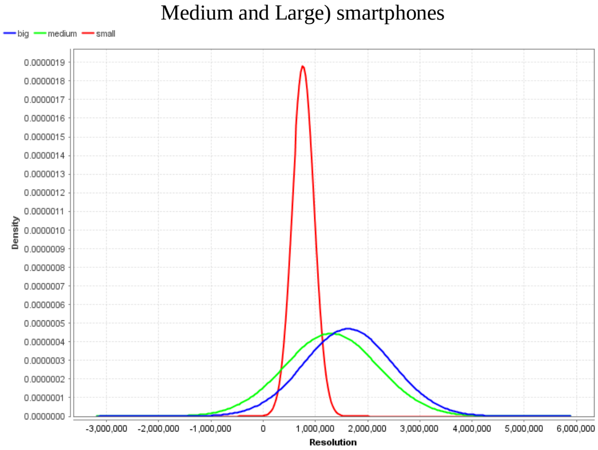

Figure 6. Resolution classification over (Small, Medium and Large) samples of the smartphones devices

IV. EXPERIMENT

The propose experimental process is presented in Figure 7. The experimental was carried out with IBM SPSS Modeler Version 15 on a Machine DELL Inspiron 15 model, 4Gb RAM, 500 GB HDD, 64- bit OS, Intel (R) Core (TM)2 Duo CPU @ 4.00 GHz.

The experiment was carried out after preprocessing the dataset. The loaded experimental dataset were partitioned into training and test dataset. This process was repeatedly taken in order to ensure consistent findings. Therefore the dataset were partition into several ratios for the experiment

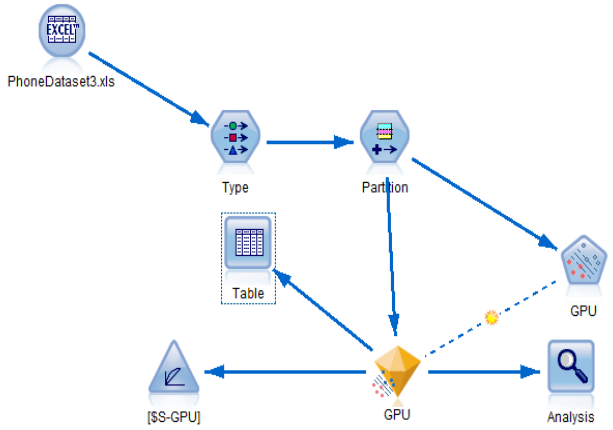

Figure 7. The proposed experimental framework

The first partition used for the experiment was in the ratio of 50% for training and 50% for testing. This lies with the fact that errors in the modeling process will be decrease. The four subsequent partition ratios were set to have the number in training data to be greater than the test dataset [24] in a ration of (60% - 40%), (70% - 30%), (80% - 20%). The data were explored to ensure all attributes are in their correct respective columns as expected. Simulations were carried on the dataset using SVM which adheres to the procedure described in section 3.1. SVM parameters

V. RESULT

The results of the simulations from four difference partitioned dataset are presented in this section. The SVM for the classification has been performed for both the target and predictors. The class of the GPU are determines by three attributes (screen size, resolution and memory) to produce the predicted result at different levels of experiment. The performance accuracy at four different partitioned dataset of the model is presented in Figure 8 to 11.

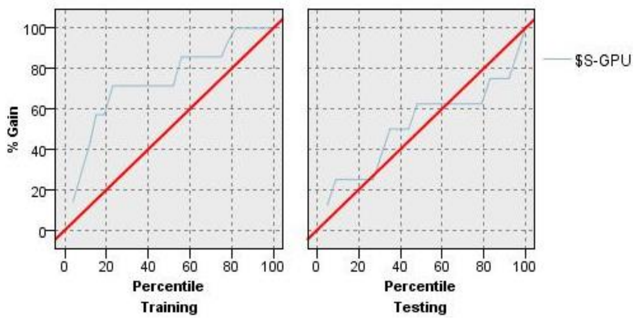

Figure 8. Performance of SVM on training-test (50%-50%)

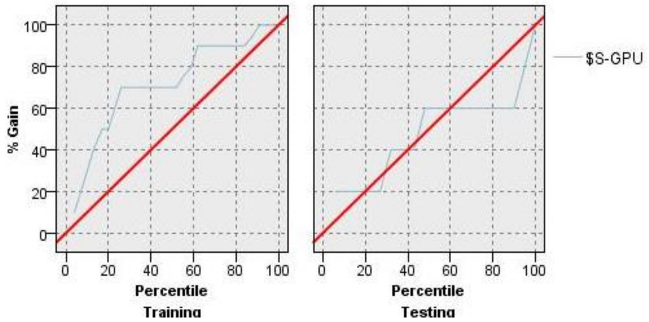

Figure 9. Performance of SVM on training-test (60%-40%)

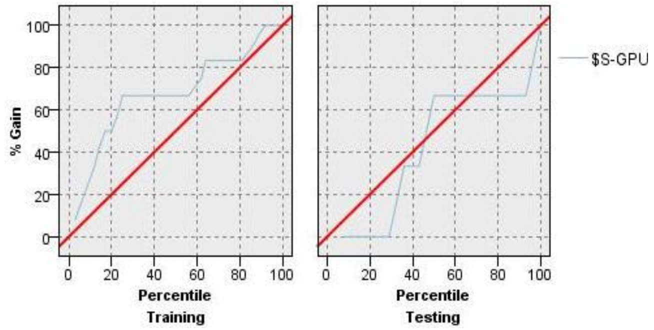

Figure 10. Performance of SVM on training-test (70%-30%)

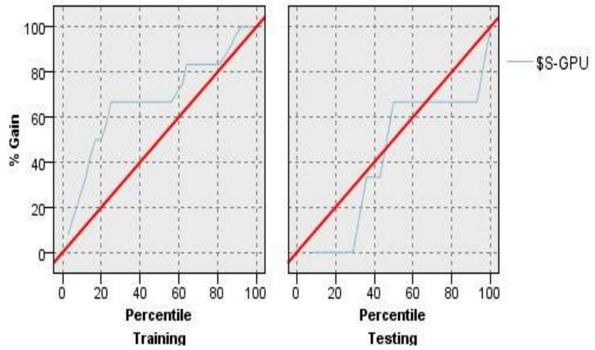

Figure 11. Performance of SVM on training-test (80%-20%)

The performances of the experiments indicated from Figure 8 – 11 were observed based on many thresholds for the independence of the attributes included in the datasets. It has been observed that the experimental performances for the partitions ratio 70% - 30% and 80% - 20% are the same. The best performance testing was obtained in the experiment with the partitioning of the dataset at 60% - 40% ratio. For the fact that computational time is an important performance metric, the computational time for the entire experiments were fast, measure in milliseconds. The minimum experimental error for training in all the scenarios were relatively the same as compared to experimental error for testing. The maximum error at which the experiment were carried out are shown in Table 2. The mean error and the absolute mean error which the experiment was carried out were almost the same in all cases. The linear correlations in all cases were positive and ranges between "0.21 to 0.69". The highest value was obtained at the training with partition ratio 60% - 40%.

TABLE II. EXPERIMENTAL RESULTS

Partition	Tr_50/50	Te_50/50	Tr_60/40	Te_60/40	Tr_70/30	Te_70/30	Tr_80/20	Te_80/20
Minimum Error	-249.233	-151.16	-249.433	-152.935	-249.364	-96.054	-249.364	-96.054
Maximum Error	317.435	249.602	314.135	248.051	313.585	247.926	313.585	247.926
Mean Error	-4.583	27.468	8.485	10.207	6.19	16.665	6.19	16.665
Mean Absolute Error	83.843	71.055	86.071	64.487	84.259	61.264	84.259	61.264
Standard Deviation	121.828	103.35	121.748	101.056	117.82	103.95	117.82	103.95
Linear Correlation	0.671	0.339	0.699	0.214	0.63	0.214	0.63	0.214
Occurrences	27	23	31	19	36	14	36	14

TR=TRAINING, TE = TESTING

This result indicate that at partition ratio 60% - 40% experiment, the predicted values are with the best outcomes. Although, it's customary to find that models performance might be better on training dataset compared to test dataset [25]. Thus, performance of a model might not be stable based on the aggregate of modelling process. Therefore for this research, the result of the proposed SVM classification of GPU with respect to memory, screen size and resolution of mobile device has been obtained. The target and predicted outcomes for 10 entry is presented in Table 3.

TABLE III.
THE RESULT OF THE EXPERIMENTS

Target	Experiment	Predicted
400	1_Training	450.822
320	1_Training	449.13
520	1_Training	449.77
400	2_Testing	450.851
320	1_Training	449.131
384	1_Training	449.001
400	2_Testing	450.696
450	1_Training	450.69
450	1_Training	449.13
400	1_Training	452.921

The classification of GPU based on proposed combinations of other mobile devices attributes concur with the fact that GPU used to parallelize problem which CPU handle thereby making sequentially computation possible is depends on other variables as well. This is research in particular consider event were 3D map for navigation is the service at hand. The range of predicted values in Table 3 is within 450MHz, this result shows that using GPU of such ranges not only can speed up the processing by parallelize the problem, it also can reduces the memory transfer between main memory

VI. CONCLUSION

This research proposes the classification of computational resources necessary for 3D map that could be use in mobile devices for navigation aid. This proposition comes from previous research studies that shows differences in mobile devices computational capabilities affect the use of 3D maps in mobile device for navigation aid. The motivation of the work lies with the fact that analytical study for 3D map navigation system for mobile devices were neglected. As a result this research uses SVM to analytically classify mobile device computational capabilities required for 3D map that will be suitable for use as navigation aid. Three feature of mobile device namely, screen size, resolution and memory were classified with GPU to aid in efficient navigation service. The classification use samples of fifty different Smart phones. The performance of the classification accuracy in both training and test dataset was adequate. In general the proposed classification was found to be accurate on SVM experiment. This result is significant to mobile navigation aid designers and developers and computational communities at large for understanding the required computational resources for 3D mobile map.

ACKNOWLEDGMENT

The research is partially supported by Faculty of Science and Technology, Sampoerna University, Jakarta, Indonesia.

REFERENCES

[1] A.I, Abubakar, T. Mantoro, and M.A. Shafi'i. "Dynamic interactive 3D mobile navigation aid." Journal of Theoretical and Applied Information Technology, vol. 37, no. 2, 2012.

[2] T. Mantoro, A.I. Abubakar, C. Haruna, "Pedestrian position and pathway in the design of 3D mobile interactive navigation aid". In Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia, pp. 189-198. ACM, 2012.

[3] A.I. Abubakar, T. Mantoro, and M.Mahmud. "Exploring end-user preferences of 3D mobile interactive navigation design." In Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia, pp. 289-292. ACM, 2011.

[4] A. Nurminen, "m-LOMA-a mobile 3D city map." In Proceedings of the eleventh international conference on 3D web technology, pp. 7-18. ACM, 2006.

[5] A.I. Abubakar, M.Z. Akram, C. Haruna, "3D Mobile Map Visualization Concept for Re-mote Rendered Dataset". In IEEE International Conference on Advanced Computer Science Applications and Technologies, pp. 228-231. IEEE, 2013.

[6] T. Mantoro, A.I. Abubakar, "Pragmatic framework of 3D visual navigation for mobile user", In IEEE International Conference on Information and Communication Technology for the Muslim World (ICT4M), pp. D19-D24. IEEE Press, Indonesia 2010.

[7] V. Coors, and A. Zipf. "MoNa 3D–mobile navigation using 3D city models." LBS and Telecartography 2007.

[8] A. Oulasvirta, S. Estlander, and A. Nurminen. "Embodied interaction with a 3D versus 2D mobile map." Personal and Ubiquitous Computing 13, no. 4 2009: 303-320.

[9] T. Mantoro, A. Abubakar, and M. Ayu. "Multi-user navigation: A 3D mobile device interactive support." In Industrial Electronics and Applications (ISIEA), 2011 IEEE Symposium on, pp. 545-549. IEEE, 2011.

[10] A.W.J. Borgers, and H. J. P. Timmermans. "Simulating pedestrian route choice behavior in urban retail environments." In Walk21-V Conference "Cities for People", Copenhagen. 2004.

[11] A. Miho, T. Iryo, and M. Kuwahara. "Microscopic pedestrian simulation model combined with a tactical model for route choice behaviour." Transportation Research Part C: Emerging Technologies 18, no. 6 (2010): 842-855.

[12] R.P. Darken and B. Peterson, "Spatial orientation, wayfinding, and representation" Handbook of virtual environments, pp. 493-518. 2001.

[13] M. Davidich, and G. Köster. "Predicting pedestrian flow: A methodology and a proof of concept based on real-life data." PLoS one 8, no. 12 (2013): e83355.

[14] A.I. Abubakar, M.Z. Akram, H. Chiroma, and T. Herawan. "Investigating Rendering Speed and Download Rate of Three-Dimension (3D) Mobile Map Intended for Navigation Aid Using Genetic Algorithm." In Recent Advances on Soft Computing and Data Mining, pp. 261-271. Springer International Publishing, 2014.

[15] V. Vapnik, and A.J. Lerner. "Generalized portrait method for pattern recognition." Automation and Remote Control 24, no. 6 (1963): 774-780.

[16] K. Kim, and J. Lee. "Sentiment visualization and classification via semi-supervised nonlinear dimensionality reduction." Pattern Recognition 47, no. 2 (2014): 758-768.

[17] S. Chen, P. Yu, and B. Liu, "Comparison of neural network architectures and inputs for radar rainfall adjustment for typhoon events". Journal of Hydrology 405 (2011)150–160.

[18] D. Meyer, F. Leisch, and K. Hornik. "The support vector machine under test." Neurocomputing 55, no. 1 (2003): 169-186.

[19] L. Hamel "Knowledge discovery with support vector machines". John Wiley & Sons, Inc. New Jersey (2009)

[20] T. Hachaj, "Real time exploration and management of large medical volumetric datasets on small mobile devices—evaluation of remote volume rendering approach." International Journal of Information Management 34, no. 3 (2014): 336-343.

[21] Y. Okamoto, T. Oishi, and K. Ikeuchi. "Image-based network rendering of large meshes for cloud computing." International Journal of Computer Vision 94, no. 1 (2011): 12-22.

[22] A. Abubakar, M.Z. Akram, S.A. Muaz, H. Chiroma, and T. Herawan. "Visualisation of a Three-Dimensional (3D) Objects Optimal Reality in a 3D Map on a Mobile Device." Appl. Math 7, no. 1-12 (2013): 1.

[23] S. Weiss. Handheld usability. John Wiley & Sons, 2003.

[24] I.H. Witten, E. Frank and A.M. Hall, "Data Mining: Practical Machine Learning Tools and Techniques", San Mateo: Morgan Kaufmann, 2011.

[25] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," Soft Comput, Vol 9. 3-12, 2005.

AUTHORS

A.I Abubakar is currently an Assistant Professor at International Islamic University of Malaysia, Kuala Lumpur. His academic qualifications were obtained from Bayero University Kano Nigeria, for bachelor and post graduate diploma and master degrees, and in International Islamic University Malaysia for PhD degree. His research area of interest is on Navigation. He is now working on 3D Navigation Aid. (e-mail: adamu@iium.edu.my).

T. Mantoro is currently a professor at faculty of science and technology, Sampoerna University, Jakarta, Indonesia. He holds PhD, MSc and BSc, all in Computer

Science. He received the PhD from School of Computer Science, the Australian National University (ANU), Canberra, Australia. His research interest are in Pervasive/Ubiquitous Computing, Context Aware Computing, Mobile Computing and Intelligent Environment. (e-mail: tmantoro@gmail.com).

S. Moedjiono is currently Director of Postgraduate Program, Budi Luhur University, Jakarta, Indonesia. In addition, he is also serving as an Executive Team Member of the Indonesia National ICT Council, and a member of Multi-stakeholder Advisory Group of Internet Governance Forum (MAG/IGF) Indonesian Chapter. He was serving in the Indonesian Navy, then joining the Ministry of Communication and Information Technology, and member of Government Advisory Committee of the Internet Corporation for Assigned Names and Numbers (GAC/ICANN). He earned his Master of Science degree in Computer Science from the United States Naval Postgraduate School, and a Doctor of Science in Computer Science from the George Washington University, Washington, DC, USA. He has authored several research papers posted in journals and proceedings. (e-mail: moedjiono@budiluhur.ac.id).

H. Chiroma is a PhD candidate at the Department of Artificial Intelligence, University of Malaya and a Lecturer at the Federal College of Education (Technical), Gombe, Nigeria. He holds a BTech and MSc in Computer Science from Abubakar Tafawa Balewa University, Bauchi, Nigeria and Bayero University Kano, Nigeria respectively. His main research interest is on metaheuristic algorithms (e-mail: freedonchi@yahoo.com).

A. Waqas is with the department of computer science sukkur institute of business administration, Sukkur Pakistan as a lecturer and currently working on Cloud Computing security issues for a self-securing architecture for federated clouds as part of his PhD research in the Department of Computer Science, Faculty of Information and Communication Technology, International Islamic University Malaysia. His major research interest are on, Distributed systems, Cloud Computing, Network Design, Security and Management, Algorithms and Data Structures. (e-mail: ahmad.waqas@iba-suk.edu.pk).

S.M Abdulhamid is a PhD candidate at the faculty of computing, Universiti Teknologi Malaysia, Johor Bahru Malaysia and a Lecturer at the Federal University of Technology Minna, Nigeria. He holds a BTech and MSc in Computer Science from the Federal University of Technology Minna, Nigeria and Bayero University Kano, Nigeria respectively. His Research Interests:Cyber Security, Grid Computing, Cloud Computing, Machine Learning, and Network Security (e-mail: shafii.abdulhamid@futminna.edu.ng).

M.F. Hamza is with the department of Mechatronics Engineering, Bayero University Kano, Nigeria and currently he is a PhD candidate at the department of mechanical engineering, University of Malaya, Kuala Lumpur, Malaysia. His major research interest is on PSO-Based Controllers for Non-Linear Systems as a PhD candidate. Malaysia (e-mail: emukhtarfh@gmail.com).

Abdulsalam Ya'u Gital is a lecturer at the Mathematical Science Department, Abubakar Tafawa Balewa University Bauchi, Nigeria. He obtained his PhD in computer science at the University of Technology, Malaysia, B.Tech. and MSC in Computer Science from Abubakar Tafawa Balewa University, Bauchi, Nigeria. His area of research interest includes computer communications, virtual environment, intelligent systems, software engineering etc. He has published over 30 papers in prestigious international journals, proceedings, edited books and local journals.