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

http://dx.doi.org/10.3991/ijim.v10i3.5056

Adamu Abubakar[1], Teddy Mantoro[2], Sardjoeni Moedjiono[3], Media Anugerah Ayu[2], Haruna Chiroma[4], Ahmad Waqas[1], Shafi'i Muhammad Abdulhamid[5], Mukhtar Fatihu Hamza[6], Abdulsalam Ya'u Gital[7]
[1]International Islamic University Malaysia, Kuala Lumpur
[2]Sampoerna University, Jakarta, Indonesia
[3]Budi Luhur University, Jakarta
[4]University of Malaya, Pantai Valley, Kuala Lumpur
[5]Universiti Teknologi Malaysia, Johor Bahru Malaysia.
[6]Bayero University, 3011, Kano, Nigeria
[7] Abubakar Tafawa Balewa University Bauchi, Nigeria

***Abstract*—3D maps for mobile devices provide more realistic views of environments and serve as better navigation aids. Previous research studies show differences on how 3D maps effect the acquisition of spatial knowledge. This is attributable to the differences in mobile device computational capabilities. Crucial to this is the time it takes for a 3D map dataset to be rendered for a complete navigation task. Different findings suggest different approaches on how to solve the problem of time required for both in-core (inside mobile) and out-core (remote) rendering of 3D datasets. Unfortunately, there have not been sufficient studies regarding the analytical techniques required to show the impact of computational resources required to use 3D maps on mobile devices. This paper uses a Support Vector Machine (SVM) to analytically classify mobile device computational capabilities required for 3D maps that are suitable for use as navigation aids. Fifty different Smart phones were categorized on the basis 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 maps which provide a more realistic view of the environment and serve as an improved navigation aid [1]. The main advantage of 3D maps is that they enhance visual quality and, when used as a navigation aid, improve navigation practices [2]. Current mobile device resources, combined with increasing wireless networking capabilities and Global Positioning System (GPS) receivers, offer improved navigational capacities [3]. There is a need for computing communities to take advantage of these properties in order to improve mobile services and enhance spatial knowledge.

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

Unfortunately, analytical durations for both in-core and out-core rendering with respect to visualization quality of 3D maps for navigation have been largely neglected. Analytical model construction and the study of data patterns required for 3D map visualization on mobile devices for navigation mostly jumped to conclusions [4, 7-8]. Most studies agreed that the greater the computational resources the better the 3D map visualization navigation aid [4-9].

This paper utilizes support a vector machine (SVM) algorithm to map the mobile device resources required for navigation while using a 3D map. The reason for using this algorithm is because it imitates the real-life process of demarcating two or more elements for optimization to obtain the best solution for understanding the required computational resources for a 3D map that is suitable for aiding navigation using mobile devices.

II. RELATED WORK

Devices to aid navigation aid help people 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 the names of streets, an important structure, or sometimes even landmarks. This information, when combined with a paper 2D map, helps persons identify places to a reasonable degree. The devices to aid navigation such as in-car GPS navigation devices like Garmin or TomTom's, have gained huge support [4]. Such tools typically use 2D and 3D projection of the upcoming road layout, without substantial pictorial realism, and without freedom for the user to substantially manipulate the perspective [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]. Most commonly, however, others combine 2D maps [4]. All of these approaches intend to help circumvent the drawback of paper maps that need further description of symbols or legends [4].

Technical problems associated with devices designed to aid navigation having on-board Global Positioning Systems (GPS) were addressed in [4, 7-8]. Unfortunately, problems with navigation practices are mostly evaluated subjectively like, for instance, route choice and crossing pedestrian behavior. Navigation was evaluated in urban areas by subjective assessment, where it was assumed that pedestrians started from one network and traversing to another network by selecting consecutive links, then drawing to a conclusion [10]. Similarly, the tactical model for route choice behavior of a pedestrian's travel time in multidirectional flows is 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 system to aid navigation. The first step requires acquiring the real-life data of pedestrian flow to provide a proof of concept, similar to the study performed in [13]. After this, an analysis of rendering speed (for in-core and out-of-core rendering) and download rate or (out-of-core rendering) of the 3D map must be perform, similar to the study [14]. Finally, implementations of the designed model must be tested and evaluated, similar to the study performed in [4].

There are presently several research studies that show a positive impact of 3D maps to spatial knowledge, mobile device computational capabilities, mobile device physical properties and navigation tasks [4-9]. However, there is a present lack of analytical approaches that utilize 3D map datasets to investigate the time required for both in-core and out-core rendering of 3D datasets or their visualization. In general, there is a need for the meaningful prediction of the approximate computational resources mobile device require to render 3D maps for navigation aid. This is because viewing 3D maps in mobile devices have the invaluable potential of providing a greater perception of a specific environment, where real-world entities can be recognized easily.

III. RESEARCH METHODOLOGY

A. *Support Vector Machine*

The SVM technique presented in this work 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 SVM is equivalent to a linear constrained quadratic programming problem which translates to the exceptional and global optimum. The optimum solution obtained by SVM for any given problem is highly associated on support vectors [17]. SVMs have demonstrated acceptable performance when compared with other methods, especially when used for classification [18]. Figure 1 shows how SVM works.

Suppose from Figure 1, that two sets of samples classes are given, (black sample and blue sample) within the *x, y* plane that require classification using SVM. Principally, SVM will design a hyperplane between (class black and class blue) as it classifies all training vectors into two classes by means of a linear discriminant function:

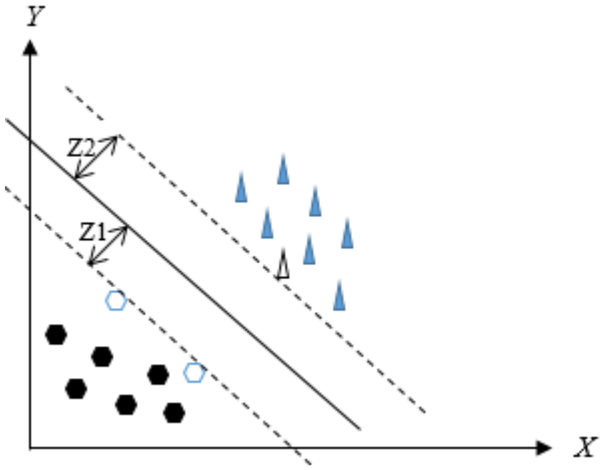

Figure 1. Support Vector Machine

$$g\left(\vec{x}\right) = \vec{\omega}..\vec{x}_i + b\cdot \qquad (1)$$

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

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

it will deliver values greater than 1 for all the input vectors which belong to the class black. On the other hand

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

it will deliver values smaller than -1 for all values belonging to class blue. This implies that if every training sample belonging to the sample of class black starting from the initial value $\vec{\omega}$ and b is greater than zero, it satisfies the class, or $\vec{\omega}$ and b is modified in such a way that the position of the hyperplane is moved to the positive side. Thus, for every vector, there is y_i such that $y_i = +1$ represents a positive sample and $y_i = -1$ represents a negative sample. If the distances (z1 and z2) to the closest elements are 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 vectors. Thus, setting a margin α as the measure of distance of the separating hyperplane and considering the equation from the geometry, the distance between a point and a hyperplane is computed by equation 2 as;

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

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

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

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

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

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 corresponds to the hyperplane will be computed by equation 5.

$$\frac{1}{\left|\vec{\omega}\right|} + \frac{1}{\left|\vec{\omega}\right|} = \frac{2}{\left|\vec{\omega}\right|} \quad (5)$$

To extract the support vectors from the training vectors, the most reliable hyperplane should be positioned where both sides' feature vectors are greatly influenced by the support vectors. $\vec{x}$ is the support vector when equation 6 is equal to 1.

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

Therefore, from equation 2, the margin has to be as great as possible over the distances to obtain 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; this can be solved by the Karush-Kuhn-Tucker (KKT) conditions, using Langrange multipliers. However, using equation 6 as the constraint for the support vector can minimize the weight vector $\vec{\omega}$. Since this is a constraint optimization problem, it can then be converted into an unconstrained problem by using Langrange multipliers λ_i in equation 7 and then optimized;

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

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

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

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

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

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

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

In order to design the Langrange multiplier expression, one has to 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, Langrange multipliers $\lambda_i = 0$ indicate that the corresponding training feature vector $\vec{x}_i$ is not a support vector. If Langrange multiplier λ_i is very high, the corresponding training feature vector $\vec{x}_i$ has a great influence over the decision of the boundary hyperplane. Otherwise, it might be affected by outliers and becomes extraordinarily high.

Using equation 8, the classification decision will be for an unknown z where $d\left(z\right)$ is 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, it is classified to class black; otherwise, class blue. Therefore, the value of b is obtained by equation (11)

$$\frac{1}{2}\begin{bmatrix} \min\left(\sum \lambda_i y_i \left(\vec{x}_i..\vec{x}_j\right)\right)\left(i\left|y_i = +1\right.\right) + \\ \max\left(\sum \lambda_i y_i \left(\vec{x}_i..\vec{x}_j\right)\right)\left(i\left|y_i = -1\right.\right) \end{bmatrix} \quad (11)$$

One of the key features of a support vector machine is its ability to let ω 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 an unknown feature vector set. Importantly, where there are multiple class problems, there is a need for multiple numbers of support vector machines [19].

B. Dataset and Preprocessing

This research considers using modern smartphones available in the market. The key feature for selecting a smartphone for this work is a programmable GPU unit, which is responsible for graphic rendering, although many smartphones possess low-powered mobile GPUs, which is a sign they have 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 this will obvious lead to poor visualization as well [20]. Consistent with these issues, this research utilized fifty smartphones and proposes an optimal classification of the resources required for rendering 3D maps for both in-core and out-of-core to mobile devices. With due consideration to [21] which reports that "recent advances in sensing and software technologies enable us to obtain large-scale, yet fine 3D mesh models (nearly 2 × 107 input triangles)", smartphones sizes, resolution and memory are the constraint functions and GPU is the objective function. Thus, 7 smartphones out of the 50 which were chosen for this research are presented in Table 1.

The 50 samples were further preprocessed. It is important to note that the size values are in inches and the resolutions are the display pixels relative to the screen sizes of the smartphones. These values were calculated by taking the product for each smartphone. The GPU values were acquired by their clock speed values, which are similar to size attributes. Finally, the memory reported is the

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

internal memory of each smartphone used. The distribution of parameters were calculated by normality test (see Figure 2).

The screen size dataset attest to a good normal distribution when compared to the rest. The majority of smartphone screen sizes range from 4 to 6 inches. However, there is a huge disparity in the distribution of memory and resolution. GPU distributions are relatively good. These distributions suggest that there is variability among the key computational resources required for 3D map rendering. Therefore, this research proposes classifying computational resources based on the 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 determining the future screen size of mobile devices with 3D maps for navigation aid. Currently, 3D maps are mostly on smartphones with different screen sizes ranging from small, medium to large. Small sizes range from (3 - 4.9 inches, medium (4.5 - 5.9 inches) and large (greater than 6 inches), based on findings from [22-23]. Therefore, the normality of the dateset pertaining to this classification is presented in Figure 3, 4 and 5.

There is some deviation to the normal curve among the three GPU classifications used (see Figure 3). The low to medium size GPU were greater than the high powered GPU size. However, the gaps between them are greater than for memory (see Figure 4). This indicates that 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 difference. The classes of resolutions show that the majority of the devices are in low resolution class (see Figure 6), whereas the medium and large resolutions classes are very low.

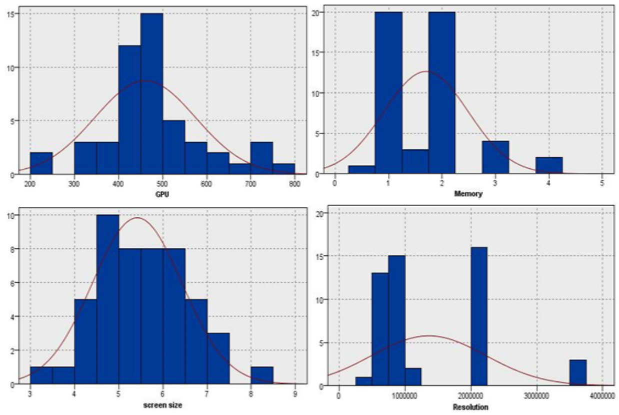

Figure 2. The distribution curve of experimental parameters, the top left is for GPU, the top right is for Memory, the bottom left is Screen Size and bottom right is Resolution

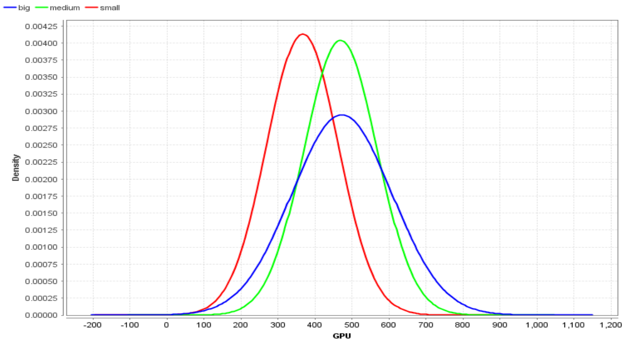

Figure 3. GPU classification over the samples of the smartphone devices

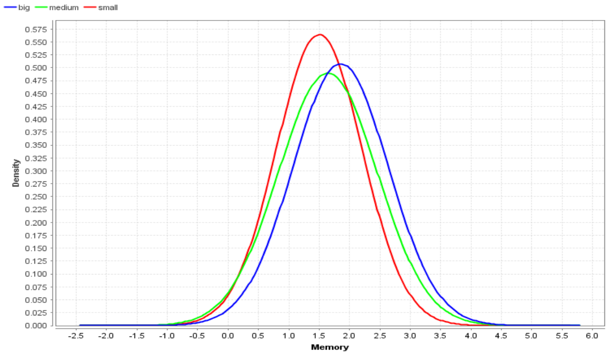

Figure 4. Memory classification of the smartphones sampled

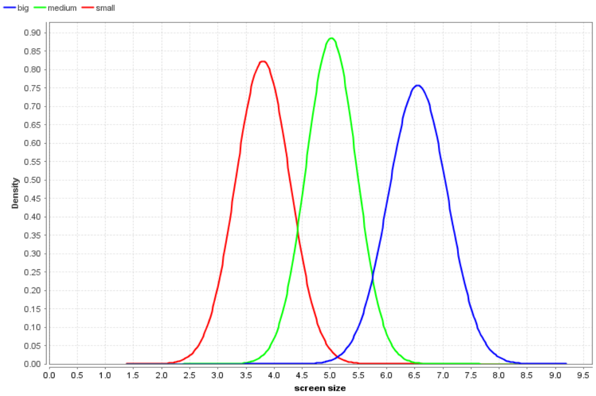

Figure 5. Screen size classification of the smartphones (small, medium and large) sampled

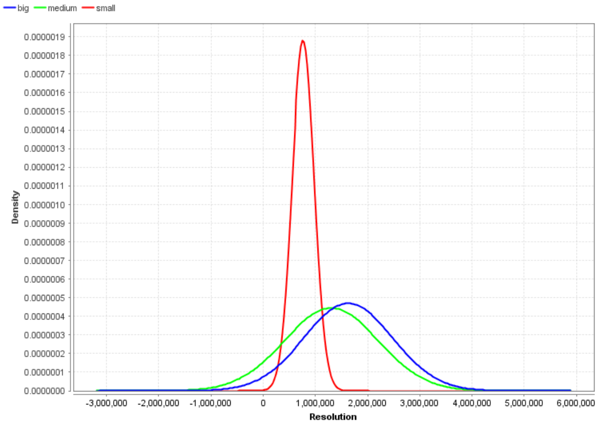

Figure 6. Resolution classification of the smartphones (small, medium and large) sampled

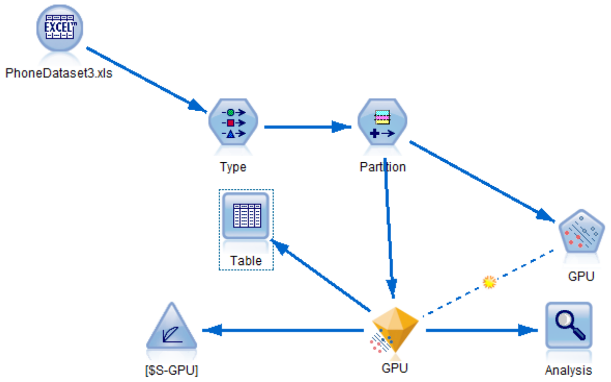

Figure 7. The proposed experimental framework

IV. EXPERIMENT

The proposed experimental process is presented in Figure 7. The experiment was carried out with the IBM SPSS Modeler Version 15 on a DELL Inspiron 15 with 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 was partitioned into training and test datasets. This process was repeatedly taken in order to ensure consistent findings. Therefore, the dataset was partitioned into several ratios for the experiment.

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

V. RESULTS

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

Figures 8-11 presents performance of the experiments observed on different grouping of dataset. The data show that the performance on group of dataset with the 70% - 30% and 80% - 20% partitioning ratios for training and testing is the same. The best performance was obtained in the experiment with 60% - 40% dataset partitioning ratio. Because computational time is an important performance metric, the computational time for the entire experiments was fast, measure in milliseconds. The minimum experimental error for training in all the scenarios was relatively the same when compared to experimental error for testing. The maximum error at which the experiments were carried out is shown in Table II. The mean error and the absolute mean error 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 with partition ratio 60% - 40%.

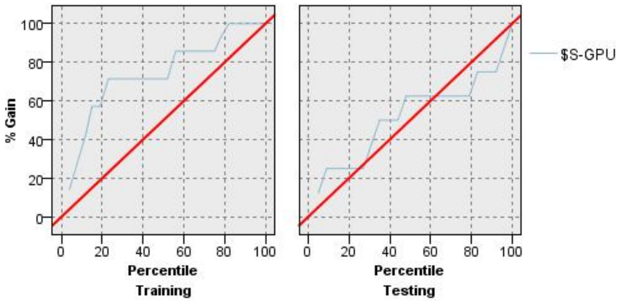

Figure 8. Performance of SVM on training-test (50%-50%)

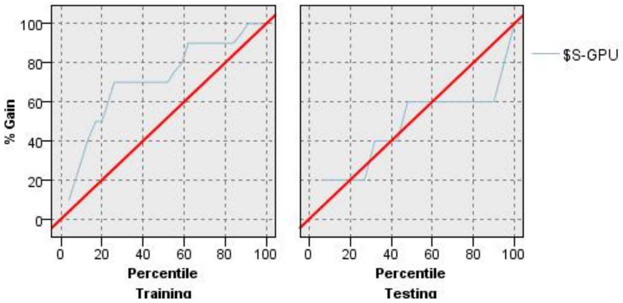

Figure 9. Performance of SVM on training-test (60%-40%)

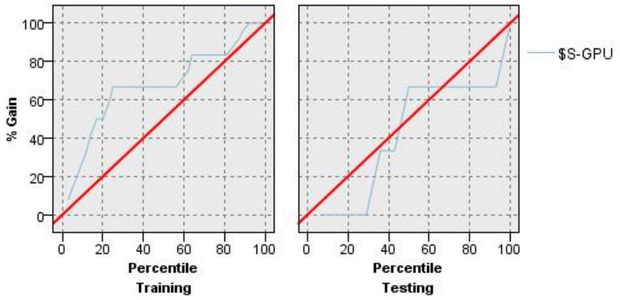

Figure 10. Performance of SVM on training-test (70%-30%)

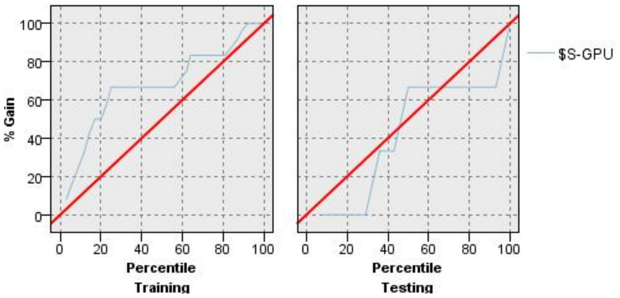

Figure 11. Performance of SVM on training-test (80%-20%)

Results indicate that at a 60% - 40% partition ratio, the predicted values produce the best outcomes, although it is customary to find that model performance might be better on the training dataset compared to the test dataset [25]. Thus, performance of a model might not be stable based on the aggregate of the modelling process. Based on this the results of the proposed SVM classification of the GPUs with respect to memory, screen size and resolution of mobile device were obtained. The target and predicted outcomes for the 10 entries chosen are presented in Table III.

The proposed classification of the GPUs based on different mobile device used for this research agrees with the fact that GPUs are used with CPU in an order. The higher

TABLE II.
EXPERIMENTAL RESULTS ON THE SMARTPHONE DATASETS

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

TABLE III.
THE RESULT OF CLASSIFICATION 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 problem which CPU handles the higher the GPU requires to make similar load of computation possible. This also depends on other variables as well. This research in particular considered event for 3D map navigation is the service at hand. The range of predicted values in Table III is within 450MHz, demonstrating that GPUs of such ranges not only can speed up the processing by parallelizing the problem, but can also reduce memory transfer from the main memory

VI. CONCLUSION

This research proposes the classification of computational resources necessary for 3D maps that can be used in mobile devices aiding navigation. This proposition comes from previous research studies that show differences in mobile device computational capabilities affect the use of 3D maps in mobile devices as a navigation aid. The motivation for the work is based on the fact that analytical study of 3D map navigation systems for mobile devices has been largely neglected. As a result, this research uses SVM to analytically classify mobile device computational capabilities required for 3D maps that will be suitable for use as navigation aids. Screen size, resolution and memory were evaluated with different GPUs to support to determine how efficient a navigation service they could provide. The classification used samples of fifty different smart phones. The performance and classification accuracy in both training and test dataset was adequate. In general, the proposed classification was found to be accurate on the SVM experiment. These results are significant and will help mobile navigation aid designers and developers and computational communities at large better understand the required computational resources for 3D mobile maps.

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. http://dx.doi.org/10.1145/2428955.2428992

[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. http://dx.doi.org/10.1145/2095697.2095758

[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. http://dx.doi.org/10.1145/1122591.1122593

[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. http://dx.doi.org/10.1109/ict4m.2010.5971896

[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. http://dx.doi.org/10.1007/s00779-008-0209-0

[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. http://dx.doi.org/10.1109/isiea.2011.6108772

[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. http://dx.doi.org/10.1016/j.trc.2010.01.005

[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. http://dx.doi.org/10.1371/journal.pone.0083355

[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. http://dx.doi.org/10.1007/978-3-319-07692-8_25

[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. http://dx.doi.org/10.1016/j.patcog.2013.07.022

[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. http://dx.doi.org/10.1016/j.jhydrol.2011.05.017

[18] D. Meyer, F. Leisch, and K. Hornik. "The support vector machine under test." Neurocomputing 55, no. 1 (2003): 169-186. http://dx.doi.org/10.1016/S0925-2312(03)00431-4

[19] L. Hamel "Knowledge discovery with support vector machines". John Wiley & Sons, Inc. New Jersey (2009) http://dx.doi.org/10.1002/9780470503065

[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. http://dx.doi.org/10.1016/j.ijinfomgt.2013.11.005

[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. http://dx.doi.org/10.1007/s11263-010-0383-1

[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. http://dx.doi.org/10.1007/s00500-003-0328-5

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).

M.A. Ayu is currently an Associate Professor at faculty of science and technology, Sampoerna University, Jakarta, Indonesia. She holds a position as Head of Institute of Social Development and Community Outreach in Universitas Siswa Bangsa Internasional (USBI), Jakarta, Indonesia. Her research interest is set around the area of web application development, ubiquitous computing, mobile application for intelligent environment, activity recognition, ICT for teaching and learning, and decision support systems. Dr. Ayu is a member of IEEE and a senior member of IACSIT.

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).

A. Y. 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.

The research is partially supported by Faculty of Science and Technology, Sampoerna University, Jakarta, Indonesia. The authors wish to acknowledge the International Islamic University Malaysia (IIUM) Grant of the Ministry of Education under Foundational Research Grant Scheme (FRGS15-245-0486) support. Submitted 21 September 2015. Published as resubmitted by the authors 23 March 2016.