A New Model for Automatic Raster-to-Vector Conversion

Shereen A. Taie[1], Hesham E. ElDeeb[2], Diyaa M. Atiya[3]

[1] *Mathematics Department, Computational Science Branch, Faculty of Science*

, Cairo University, Egypt

[1] sh.taie@yahoo.com

[2]*Electronic Research Institute, Computer and Control Department, Egypt.*

[2] heldeeb@mcit.gov.eg

[3] *Mathematics Department, Faculty of Science*

, Cairo University, Egypt

[3]diyaa.atiya@yahoo.com

***Abstract*- There is a growing need for automatic digitizing, or so called automated raster to vector conversion (ARVC) for maps. The benefit of ARVC is the production of maps that consume less space and are easy to search for or retrieve information from. In addition, ARVC is the fundamental step to reusing old maps at higher level of recognition. In this paper, a new model for an ARVC is developed. The proposed model converts the "paper maps" into electronic formats for Geographic Information Systems (GIS) and evaluates the performance of the conversion process. To overcome the limitations of existing commercial vectorization software packages, the proposed model is customized to separate textual information, usually the cause of problems in the automatic conversion process, from the delimiting graphics of the map. The model retains the coordinates of the textual information for a later merge with the map after the conversion process. The propose model also addresses the localization problems in ARVC through the knowledge-supported intelligent vectorization system that is designed specifically to improve the accuracy and speed of the vectorization process. Finally, the model has been implemented on a symmetric multiprocessing (SMP) architecture, in order to achieve higher speed up and performance.**

Keywords-Automatic vectorization, GIS, SMP

I. INTRODUCTION

Graphics recognition is a pattern recognition field that closes the gap between paper and electronic documents. Currently, the graphic design systems are able to accurately edit and print out electronic diagrams. What remains a challenge [8], however, is the reverse step; that is, the accurate and automatic conversion from a paper-based document to an electronic, computer-aided design (CAD), format. This field of graphics recognition is particularly important due to its application in the electronic archiving and analysis of geographic paper-based maps.

Paper maps have always been an inevitable and fundamental part of surveying processes. By late 60's, technological advancement in surveying has led to the introduction of the GIS, which requires digital maps for processing. This in turn has accentuated the need for automatic conversion from a paper-based document to electronic versions. At present, there is a greater emphasis on digital data than on paper maps. In a sense, this corresponds to the proliferation of GIS in industrial and research environment [18]. Consequently, there is now an even greater need to transform existing paper maps into digital formats [4], [7].

Recently, tool support for ARVC of paper maps into CAD forms has received much attention in Egypt. In the "National Project for Automating Corporeal Land Registry", one of Egypt's main national projects [1], the prime objective is to convert 138,000 maps drawn since 1905 till now, from paper-based to electronic form. These maps represent 8.5 million acres and contain different graphical and textual information. Accordingly, high accuracy is crucial, as small margin of errors could lead to significant distortions of land ownership.

A straightforward approach for transferring paper-based maps into CAD format is to use a digitizing tablet. However, this process is can take days or weeks to complete because all lines have to be traced by hand [23]. A faster approach is to use automated raster-to-vector conversion technology. In that approach one can scan an image from hardcopy map, and then a software tool can convert that image into a vector format in a matter of minutes or even seconds. The benefits, of course, are the production of maps that are easy to search for or retrieve information from, not to mention using less space. Also, the vectorization process facilitates the reuse of old maps at higher level of recognition [11]. However, the vectorization process can be impeded by various factors.

First, the complexity of the map and the noise/distortion introduced by scanning are typical examples of challenges that limit the efficiency of the vectorization process [5]. Many algorithms have been developed to overcome these limitations [2], [24]. However, though they achieved various degrees of success, these algorithms are far from providing optimal solutions [19]. The reason behind this phenomenon is that most of these algorithms are generic and target many types of maps drawings, e.g. mechanical, electronic, construction, and geological drawings. To that end, these algorithms employ generic vectorization models and several sets of parameters for each type of drawing [3]. Customizing methods for specific type of drawing, in contrast to generic vectorization, can lead to significant improvement in performance [20].

Second, commercial vectorization software packages suffer from the, so called, *localization problems*. That is, difficulties in the detection and conversion of local symbols and local language. Also, old maps frequently exhibit problems such as shrank and/or dirt. Thus, there is a need for automatic vectorization model that can overcome the issues related to old maps localization problems.

Finally, current commercial vectorization software packages are semi-automatic at best; the vectorization process requires the assistance of a human operator. This could be time consuming and error-prone, particularly when attempting to convert a large number of maps, as in the Egyptian national project indicated above. Thus, there is a need for faster and more reliable conversion process.

II. The Proposed New Model

We have developed an ARVC model that enables commercial vectorization software packages to overcome the problems indicated above. The innovative aspect of our model is augmenting the vectorization process with two new pre-processing steps. First, we perform a cleaning process to remove the dirt and noise from the digital image. Second, we split the image under consideration into two images, one with all the textual information and local symbols and the other with the remaining graphical information. This allows for faster and more accurate processing of the *graphical* image, then the textual information could be superimposed on the processed map.

Thus, the proposed model comprises four phases. In the first phase, the quality of the binary map image is improved through a noise cleaning process. In the second phase, the binary map image is separated into two kinds of information: textual information and graphical information. In the third phase, the commercial vectorization software is used to vectorize the graphical information binary map image. In the last phase, the two images: the vectorized graphical information binary map image and the textual information binary map image are merged. Experimental results show the efficiency of the proposed model. Also, we show that the conversion process can be accelerated by parallelization using SMP architecture [6]. Figure 1 depicts an overview of the proposed model, while the following subsection illustrate each phase in detail.

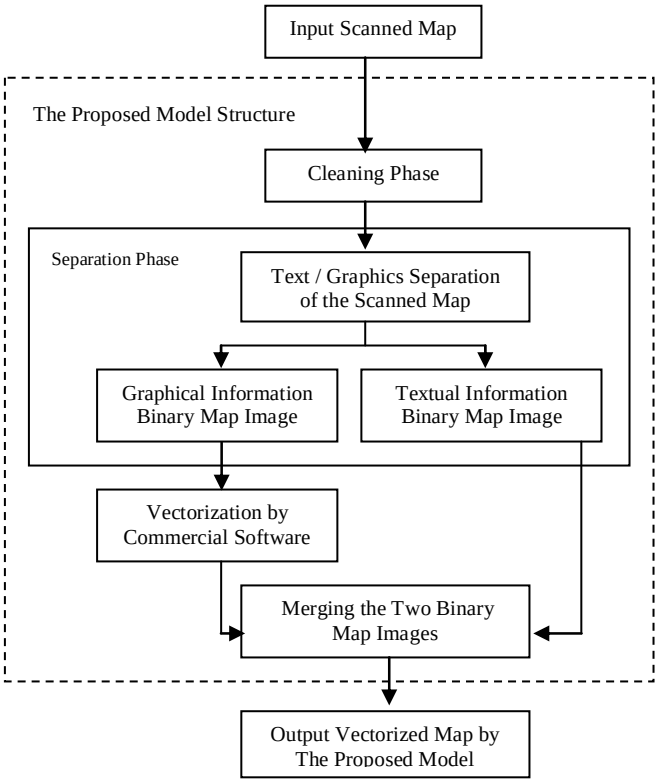
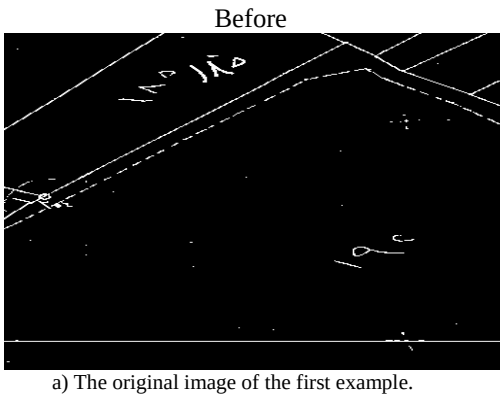


Fig. 1. Flowchart of the proposed ARVC model

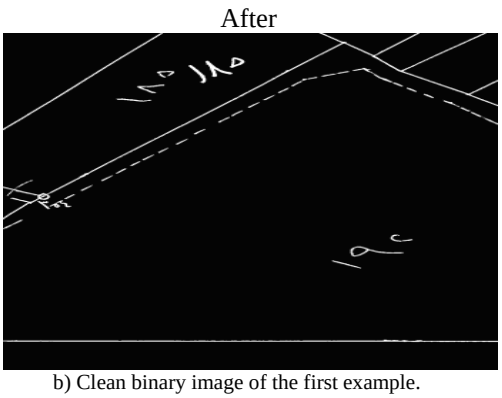
A. The Cleaning Phase

Because noise and other undesirable effects are inevitable part of scanned binary map images, we perform an automatic cleaning process [14] in order to improve the quality of input images. This process is performed through two steps. In the first step, small noise areas are removed. To achieve this goal, a threshold value is chosen according to the image resolution and then a filter is applied according to that threshold. The choice of the threshold may require some preliminary experimentation with the maps concerned, but this is important to obtain a quality result without losing significant information from the original maps. To illustrate the second step, let us assume that white pixels of the binary image are labeled with 1 (w-pixels) and the black pixels with 0 (b-pixels). Then, opening a 3 × 3 window around each pixel *p* the numbers of w-pixels and b-pixels in this window are computed. If p is labeled with 1(0) and the w-pixels number is lower (higher) than b-pixels number, then x is labeled with 0(1). This iterative analysis is applied until no pixel of the binary image is changed. The added value of that step is connecting possible discontinuities in the defining lines in the map.

Figure 2, illustrates the results of the above cleaning steps on two examples exhibiting low/high levels of textual contents and interference between textual and graphical information.

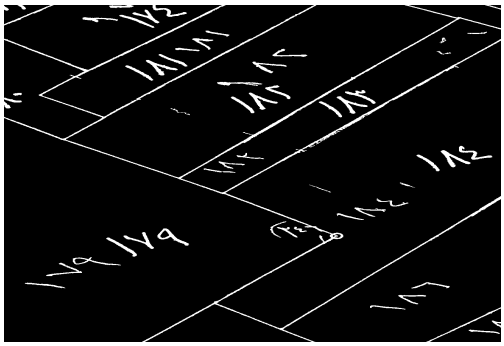


a) The original image of the first example.



b) Clean binary image of the first example.

c) The original image of the second example.



d) Clean binary image of the second example.

Fig. 2. The results of the cleaning steps on two examples

B. Text and Graphics Separation

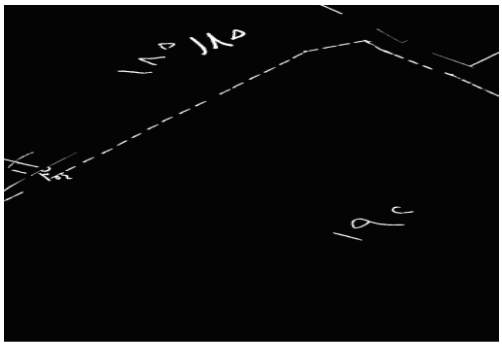
A binary map image contains two kinds of information: textual information and graphical information [10]. In this phase these two kinds of information are distinguished. The separation is done through edge based segmentation techniques [15], where edge detection filters are applied to all the components in the image to classify them into text and graphics [9]. The text and graphics classified are then stored as two separate images for further processing.

The text information obtained is stored as a pair: string / number/ Symbol and x-y coordinates of that string / number/ Symbol on the original map. Canny filter are used to detect edges, it is the basic algorithm deployed for edge detection [13]. Hough transform (HT) [21] are used to segment the raster image and detect the details, with Hough peaks 1000. Figure 3, illustrates the results of the text and graphics separation phase on the two examples.

C. Vectorization

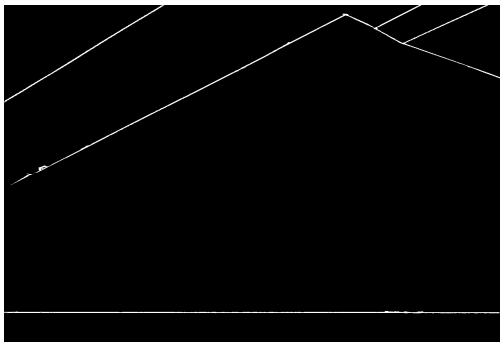
With the separation process illustrated above, we end up with two images. The first image includes all the textual information. The second image includes only the graphical information, i.e. the lines and polygons delineating the map. The graphical image is now more amenable for automatic vectorization using commercial software tools for it exhibits higher quality in terms of lower noise levels and distortions. Also, the graphical image no longer contains local symbols and textual information, which would normally hinder the process of vectorization in current commercial tools as they deal with polygons, lines and points only [7]. Thus, the result is a faster and more accurate vectorization process, as will be shown in Section V Sub-Section *B.*

Before

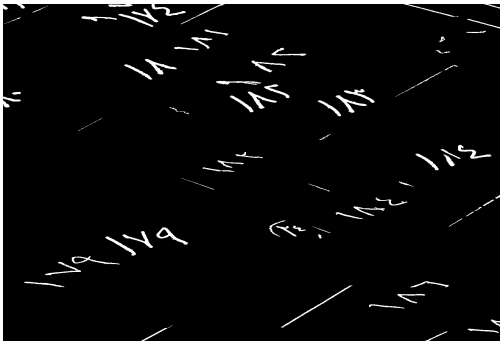


a) Textual binary map image of the first example.

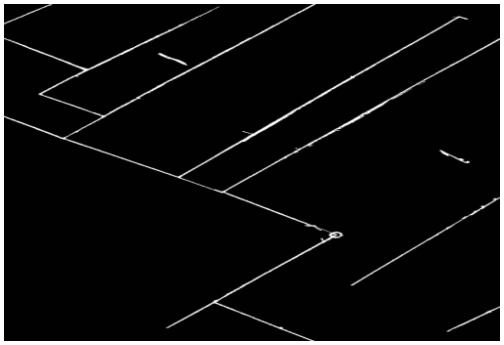
After



b) Graphical binary map image of the first example.



c) Textual binary map image of the second example.



d) Graphical binary map image of the second example.

Fig. 3. The results of the text and graphics separation phase on two examples

D. Merging Phase

With the vectorized graphical map at hand, we can restore the textual information and reproduce the original map in a vectorized form. Simply, we do this by superimposing the image containing the textual information on the vetctorized graphical map. This is done by using an "OR" logical function, which take the maximum value of each pixel in the two images. Figure 4 show the final output for the high-difficulty example in Figure 2-c. One could see that the result is exactly like the cleaned map in Figure 2-d, except that this time the output is in vectorized format rather than a raster one.

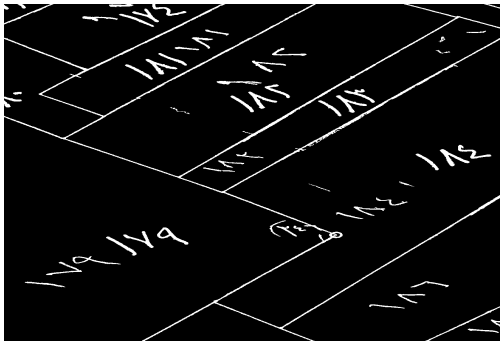


Fig. 4. The results of the merging phase

III. Parallelization of the Proposed New Model on SMP

Although the proposed model significantly reduces the time of the ARVC process, see Section V, more improvement in speedup is always desired. This is particularly important in projects involving large numbers of maps, as in the case of the Egyptian national project indicated above. Therefore, the use of high performance and parallel computing is essential to experiment with.

In this paper we used symmetric multiprocessor (SMP) cluster architecture. Where, SMP involves a multiprocessor computer hardware architecture where two or more identical processors are connected to a single shared main memory and are controlled by a single OS instance. Although, there are many other trends in High Performance Computing (HPC) [16] A symmetric multiprocessor (SMP) cluster has recently become very popular as the architecture for massively parallel computers [17] so, we use SMP in this research to help the spread of the new model.

IV. A Summary of the Applied Model

A. Sequential Implementation

The layout of our proposed model can be presented as follows.

Input : A binary map image that contains graphical and textual information.

Output : A vectorized version of the input map.

Method:

Begin

Step 1 : Apply the cleaning process – illustrated in Section II Sub-Section *A*.

Step 2 : Separate the textual information and the graphical information of the cleaned map – see Section II Sub-Section *B*.

Step 3 : Vectorize the graphical map using commercial software – see Section II Sub-Section C.

Step 4: Merge the textual binary map and vectorized graphical binary map – see Section II Sub-Section D.

End

B. Parallel Implementation

In the applied SMP implementation all processors are treated as equal; any processor can do the work of any other processor. The application is divided into threads that can run concurrently on any available processor. Any processor in the pool can run the OS kernel and execute the written threads.

Let Sn be the computer architecture with Pn, processors where n is the number of processors, and spawn is a function to deal processors. The parallel layout of our proposed model can be presented as follows:

Input : A binary map image that contains graphical and textual information.

Output : A vectorized version of the input map.

Method:

Begin

Step 1 : The first processor divide the input scanned map.

Step 2 : The first processor spawns the P_n processors including itself.

Step 3 : The first processor sends a portion of the scanned map to each one of the P_n processors including itself.

Step 4: Each P_n processor executes the full sequential implementation – see Section IV Sub-Section *A*.

Step 5: Each P_n processor saves its output vectorized version to the hard disk then check if it is the last part of the image (number of existing parts equal to the number of processors)

Step 6: The last processor merge the divided portions that output from P_n processors and output the full vectorized version of the input map.

End

V. Experiments and Results

A. The Experiments

The reader might have noticed that Step 3 in the above model is silent on which commercial vectorization tool to use. Indeed, there are approximately 20 vectorization software packages available in the market to choose from [7]. In this work we used more than one vectorization tool, to ensure the robustness of our results. This tools are the Softelec's VP Map Pro vectorization tool and WinTopo Professional vectorization tool, as these are two of the most popular products available, and requires minimum human support [20], [22]. Like other vectorization tools, VP Map Pro and WinTopo Professional still suffer from the problems of dealing with local symbols and local languages, especially Arabic.

In our experiments, we did not have to directly deal with the VP Map Pro and WinTopo Professional vectorization tools. Rather, to speed up the process, we used a batch file that takes the graphical map output of the separation phase, send it to the vectorization software, and then return the final vectorized format of the map.

We applied the proposed model on real life examples from the National Project for Automating Corporeal Land Registry1. Thirty five maps were chosen for the experiments, exhibiting various levels of difficulties in terms of both the amount of textual information included and the level of interference between the graphical and textual information.

In sequential implementation, all experiments were conducted using a PC Celeron 550, 2.00 GHZ, and 1 GB of RAM. In each experiment, we compared the execution time and accuracy of the output using our proposed model to that of:

1) Directly using the commercial software, with no preprocessing or human support
2) Manual vectorization, using AutoCAD[2]

[1] Maps are curtsey of Egyptian General Survey Authority

[2] Due to locality problems, as well as noise and distortion, both the Egyptian General Survey Authority and its subcontractor`s software subcontractors found it inevitable to use manual vectorization in order to deal with maps in the National Project for Automating Corporeal Land Registry.

In Parallel implementation, all experiments were conducted using a multiprocessor server, 18 GB of RAM, 8 processors 2.5 GB for each, and hard 3 TB. In each experiment, we compared the execution time and accuracy of the output using our proposed model in sequential and parallel implementation.

B. The Results

We measured the accuracy as "1 – error" rate, where error rate is the ratio of the number of pixels deviating from the original map divided by "the total number of pixels" [12].

The Execution time is the total execution time from the separation phase to the merging phase including all the phases in-between.

Table 1 shows the mean and standard deviation for both our control variables (accuracy and execution time) of the three compared methods throughout the 35 experiments. For illustration purposes, the results are also depicted graphically in Figure 5 and 6.

TABLE I: The Mean and Standard Deviation for Accuracy and Execution Time

Vectorization Method	Accuracy		Execution Time (Sec.)	
	Mean	Standard Deviations	Mean	Standard Deviations
Commercial Software Directly	11.0%	3.8%	1.0	0.2
The Proposed Model	99.7%	0.2%	23.1	14.4
Manual Vectorization	100%	0%	104.0	49.4

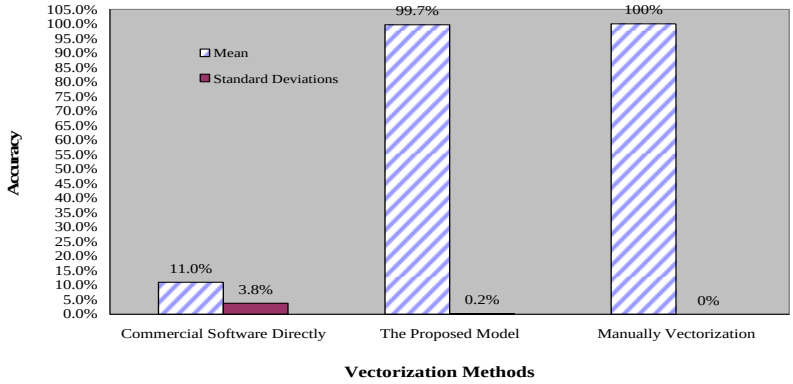


Fig. 5. The mean and standard deviation for the accuracy of the three compared methods

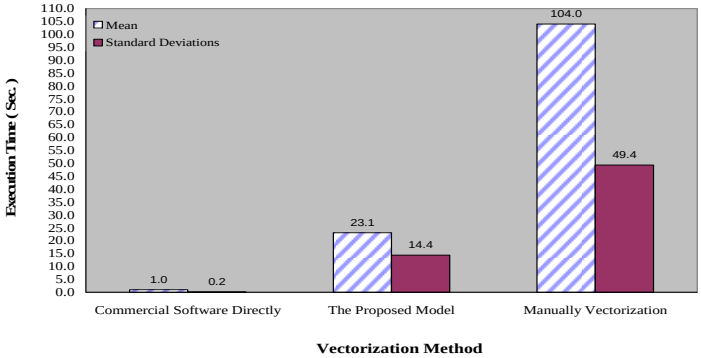


Fig. 6. The mean and standard deviation for the execution time of the three compared methods

As illustrated in figures 5, 6 one can see that, in contrast to directly usage of the commercial software, the proposed model has introduced significant improvements to the accuracy of final maps, with even less variance in performance. This has come on the expense of increasing the average execution time from 1 second in the

commercial software, to 23 seconds using the proposed model. However, we believe this is an acceptable sacrifice given the great improvement to accuracy (99.7%) and given the alternative of 104 seconds3 in using the manual process to achieve comparable accuracy in final maps.

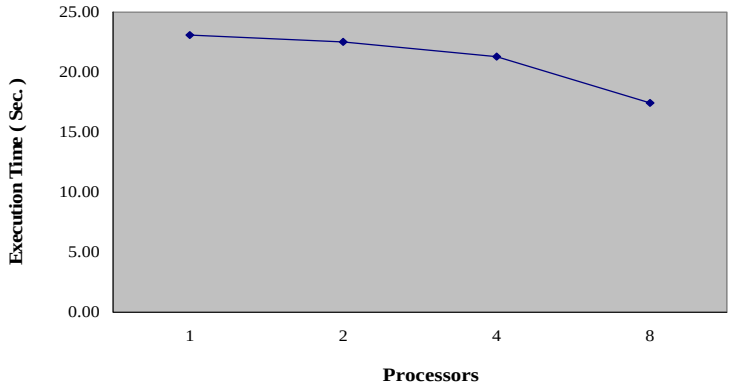


Fig. 7. The execution time with different number of processors for the proposed model

Figure 7 illustrates that the execution time of the proposed model is proportionally decreasing as the number of processors increases.

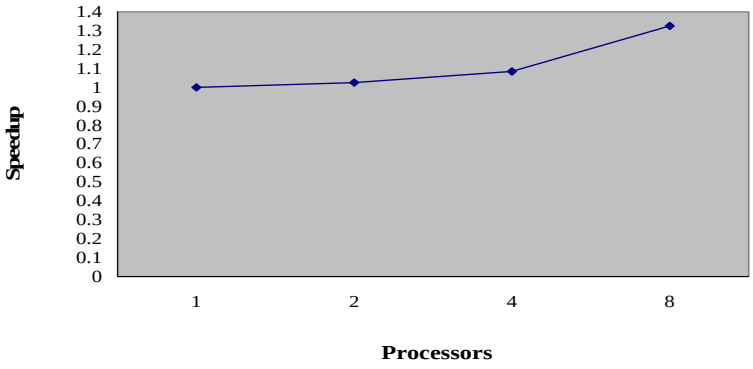


Fig. 8. Speedup

Observing that the speedup Sn is evaluated as follows [17]:

$$Sn = \frac{T1}{Tn},$$

Where n is the number of processors. T1 and Tn are the elapsed time by a single processor and that by n processors, respectively.

Figure 8 presents speedup for our proposed model on up to 8 processors. All calculations are with respect to the sequential execution time of our proposed model. We attain a linear speedup, since the communication time is negligible compared to the computation time, this is because the communication is only necessary in a global broadcast that performed before executing the full sequential implementation on each P_n processor and for the merge of the divided portions that output from P_n processors.

VI. Conclusion

In this paper, we proposed a new model for automatic vectorization of binary map images. The key advantage of the proposed model is its ability to overcome the localization problems, without a need for human interference during the vectorization process. In addition, using real life examples, we showed that the performance of the proposed model surpasses that of using commercial vectorization tools and manual vectorization, in terms of accuracy and execution time. For the work at hand, we compared the proposed model with the Softelec's VP Map Pro and the WinTopo Professional vectorization tools – two of the most popular and most automated products available. The results were identical using the Softelec's VP Map Pro and the WinTopo Professional vectorization tools. However, the experiments could be equally repeated with any other

[3] Remember that the manual process also assumes an expert user of the system.

vectorization tool; all is needed is a batch file to transfer the output of our cleaning and separation phases to the commercial tool of choice.

One could still argue that the reported average execution time of the proposed model (23 seconds) may lead to significant delays in large projects that require the processing of thousands of maps. We believe that the execution time is completely justifiable, given the high accuracy achieved and requiring no human interference during the vectorization process. Nonetheless, using SMP cluster architecture will improve the execution time of the proposed model.

REFERENCES

[1] Arab Republic of Egypt, Ministry of Communications and Information Technology, National Project for Automating Corporeal Land Registry. http://www.mcit.gov.eg, 2011.

[2] I. Baz , A. Geymen, S. N. Er, "Development and application of GIS-based analysis/synthesis modeling techniques for urban planning of Istanbul Metropolitan Area", Elsevier Science Ltd, Advances in Engineering Software, Volume 40, Issue 2, p. 128-140, February, 2009.

[3] B. Baily, M. Riley, P. Aucott, H. Southall, "Extracting digital data from the First Land Utilisation Survey of Great Britain – Methods, issues and potential ", Elsevier Ltd, Applied Geography, Volume 31, Issue 3, p. 959-968, July 2011.

[4] J. Barraud, "The use of watershed segmentation and GIS software for textural analysis of thin sections", Journal of Volcanology and Geothermal Research, Volume 154, Issue 1-2, p. 17-33, 2006.

[5] Y. B. Bai and X. W. Xu, "Object Boundary Encoding — a new vectorisation algorithm for engineering drawings", Elsevier Science B.V., Computers in Industry, Volume 46, Issue 1, Pages 65-74 , 9 August 2001.

[6] F. Cappello, O. Richard and D. Etiemble, " Understanding performance of SMP clusters running MPI programs", Elsevier Science B.V., Future Generation Computer Systems, Volume 17, Issue 6, p. 711-720, April 2001.

[7] G. Dharmaraj, "Algorithms for Automatic Vectorization of Scanned Maps", Master of Science thesis, Calgary, Alberta, Department of Geomatics Engineering, the University of Calgary, July 2005.

[8] L. Fu, L. B. Kara, "From engineering diagrams to engineering models: Visual recognition and applications", Computer-Aided Design, Volume 43, Issue 3, Pages 278-292, March 2011.

[9] W. Huang, C. L. Tan and W. K. Leow, "Associating Text and Graphics for Scientific Chart Understanding", IEEE Conference, Eighth International Conference on Document Analysis and Recognition, Vol. 2, p. 580 – 584, 2005.

[10] W. Huang, C. L. Tan and W. K. Leow SOC, "Model-based chart image recognition". Lecture Notes in Computer Science, Graphics Recognition. Barcelona, Spain: Springer, LNCS 3088, July 2003.

[11] R. D. T. Janssen and A. M. Vossepoel, "Adaptive Vectorization of Line Drawing Images", Academic Press, Computer Vision and Image Understanding, Volume 65, Issue 1, p. 38-56, January 1997.

[12] K. H. Lee, S. B. Cho, Y. C. Choy, "Automated Vectorization of Cartographic Maps by a Knowledge-based System", Elsevier Science Ltd, Engineering Applications of Artificial Intelligence, Volume 13, Issue 2, p. 165-178, April 2000.

[13] L. Ding, A. Goshtasby, "On the Canny edge detector", Elsevier Science B.V., Pattern Recognition Society, Volume 34, Issue 3, p. 721-725, March 2001.

[14] J. B. Mena, "Automatic vectorization of segmented road networks by geometrical and topological analysis of high resolution binary images", Knowledge-Based Systems, Volume 19, p. 704–718, December 2006.

[15] S. Rujikietgumjorn, "Segmentation Methods for Multiple Body Parts", Master of Science thesis, The University of Tennessee, Knoxville, July 31, 2008.

[16] E. Strohmaier, J. J. Dongarra, H. W. Meuer and H. D. Simon, "Recent Trends in the Marketplace of High Performance Computing", Elsevier Science B.V, Parallel Computing, Volume 31, Issues 3-4, p. 261-273, April 2005.

[17] M. Suzuki, H. Okuda, G. Yagawa, "MPI/Open MP hybrid parallel molecular dynamics simulation of a protein structure on SMP cluster architecture", Transactions of the Japan Society of Mechanical Engineers. A, ISSN: 0387-5008, vol. 70, no. 692, p. 519-524, 2004.

[18] J. Song , F. Su , H. Li and S. Cai, "Raster to vector conversion of construction engineering drawings", Elsevier Science B.V., Automation in Construction, Volume 11, Issue 5, p. 597-605, August 2002.

[19] K. Tombre, "Analysis of engineering drawing: state of the art and challenges", Graphics Recognition—Algorithms and Systems, Volume 1389 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, Germany, p. 257– 264, 1998.

[20] VP Map Pro ver. 6 rev. 2, "Raster to Vector Conversion Software". Available:http://www.softelec.com, 2011.

[21] D. Walsh, A. E. Raftery, "Accurate and efficient curve detection in images: the importance sampling Hough transform", Elsevier Science B.V., Pattern Recognition, Volume 35, Issue 7, Pages 1421-1431, July 2002.

[22] WinTopo Professional 3.45, "Raster to Vector Conversion Software". Available: http://www.wintopo.com, 2011.

[23] Y. T. Wu, "Raster, Vector and Automated Map Digitizing", Special Contribution by: Yecheng Ted Wu, Ph.D., Able Software Corp. (http://www.ablesw.com), 1999.

[24] A. Zanchi, S. Francesca, Z. Stefano, S. Simone and G. Graziano, "3D reconstruction of complex geological bodies: Examples from the Alps", Elsevier Ltd, Computers & Geosciences, Volume 35, Issue 1, p. 49-69, January 2009.