Earth Science Informatics (2020) 13:1187–1199
https://doi.org/10.1007/s12145-020-00501-5

RESEARCH ARTICLE

A nonrecursive GR algorithm to extract road networks in high-resolution images from remote sensing

Guilherme Pina Cardim[1,2] · Erivaldo Antônio da Silva[3] · Mauricio Araújo Dias[3] · Ignácio Bravo[4] · Alfredo Gardel[4]

Received: 27 January 2020 / Accepted: 10 August 2020 / Published online: 13 August 2020

Abstract

A number of studies address the development of algorithms based on the Growing Region (GR) technique adaptations for extracting road networks in images. However, these algorithms are high-computationally demanding and time-consuming while processing high-resolution images. The aim of this study is to introduce a modified version of the GR algorithm, named Nonrecursive Growing Region (NRGR), to extract road networks in high-resolution images from remote sensing. This study describes how the NRGR algorithm works to perform the extractions in a faster way. The proposed algorithm was developed taking into consideration the reduction of the data dependence between its tasks in order to allow the GR algorithm to process these tasks with the help of Graphical Processor Units (GPUs). The experiments were conducted to demonstrate the ability of the NRGR to process low or high spatial resolution images with or without the help of GPUs. Results achieved by experiments performed in this study suggest that the NRGR algorithm is less complex and faster than previous adaptations versions tested of the GR algorithm to process images. The NRGR was able to process the tested images with less than 30% of the time used by the recursive algorithm, reaching values below 10% in some cases. The NRGR algorithm can be used as software or hardware-software system's co-design solutions to develop maps of road networks for Cartography.

Keywords Growing region · Data processing · Algorithms · Image analysis

Introduction

An interesting field in the remote sensing area is the development of algorithms to extract cartographic features, i.e. targets of interest, such as highways, and urban or rural road networks. Numerous researchers have presented the development of algorithms to extract cartographic features from remote sensing images and reviews to present and compare their results (Cardim et al. 2018; Cheng et al. 2017; Kaur and Singh 2015; Wang et al. 2016). There are extraction algorithms based on mathematical morphology (MM) (Santiago et al. 2012; Wang and Shan 2012), fuzzy logic (Mohammadzadeh et al. 2006), edge detectors (Ali and Clausi 2001), GR algorithms (Cardim et al. 2014; Herumurti et al. 2012, 2013; Jeon et al. 2000; Lu et al. 2014; Senthilkumar et al. 2010; Xiaolin et al. 2018), among others. GR algorithms are powerful tools used by Digital Image Processing (DIP) to extract road networks in images from remote sensing. However, it is necessary to develop a road extraction GR algorithm able to process efficiently current high-resolution images from remote sensing.

Road extraction based on GR algorithms has been of growing interest for the literature (Cardim et al. 2014, Cardim et al. 2016; Herumurti et al. 2012, 2013; Jeon et al. 2000; Lu et al. 2014; Senthilkumar et al. 2010; Xiaolin et al. 2018). Lu et al. (Lu et al. 2014) developed a road extraction algorithm for Synthetic Aperture Radar (SAR) images. Their algorithm used GR to construct road networks based on roads samples (seeds) and the weighted ratio result of a line detection. Jeon et al. (Jeon et al. 2000) also developed a road extraction algorithm for Synthetic Aperture Radar (SAR) images. Their algorithm applied GR, genetic algorithm and grouping of curve segments to

Communicated by: H. Babaie

✉ Guilherme Pina Cardim
gpcardim@gmail.com

[1] State University of Londrina (UEL), Londrina 86057-970, Brazil

[2] Centro Universitário de Adamantina (UNIFAI), Adamantina 17800-000, Brazil

[3] School of Sciences and Technology, São Paulo State University (UNESP), Presidente Prudente 19060-900, Brazil

[4] Politechnic School, University of Alcalá (UAH), 28805 Alcalá de Henares, Spain

improve road extraction. Xiaolin et al. (Xiaolin et al. 2018) developed a road extraction algorithm for colorful images. Their algorithm applied GR to segment roads for autonomous driving systems. Herumurti et al. (Herumurti et al. 2013) developed a road extraction algorithm for Digital Surface Model (DSM) data. DSM was used to minimize the effects of the incidence of buildings' shadows over urban areas. A simple threshold of the model helped to extract segments of roads (seeds). Hough transform improved the seeds. Their GR algorithm used the improved seeds to extract roads. Herumurti et al. (Herumurti et al. 2012) developed another road extraction algorithm for DSM data. Their algorithm applied GR and mixed Adaptive Resonance Theory (ART) to extract roads.

Although the literature provides important papers describing road extraction based on GR algorithms for different types of images, those algorithms remain facing two challenges while processing current high-resolution images from remote sensing. They need to deal with significant quantity of data and to apply several operations on them. Road extraction methodologies are high-computationally demanding and time-consuming while processing current high-resolution images from remote sensing. The development of road extraction algorithms in hardware represents an alternative to address those limitations (Alali et al. 2013; Benkrid et al. 2001). There is also an increase demand for research related to hardware-software systems' co-design (Bartovsky et al. 2012; Déforges et al. 2013; Plaza et al. 2011; Zhu et al. 2012). Therefore, more work is needed to present new solutions in software or hardware based on road extraction GR algorithms, which can overcome these two challenges.

As mentioned, the GR algorithm is very used in researches patterns recognition in digital images. In road extraction methodologies there is a common adaptation of the GR algorithm, named in the sequence as RGR, which is supervised due to the dependence of its users to obtain samples of the targets of interest (Cardim et al. 2014). The RGR algorithm has been widely used due to its ability to segment images and choose only the interest targets connected to samples. Furthermore, RGR are easy-to-implement algorithms and their results are useful to extract cartographic features, such as road networks. This algorithm statistically calculates a range of acceptance values from those samples. For each sample, neighboring pixels are evaluated by checking which ones belong to the acceptance range. The search for neighboring pixels is performed until there are no more pixels connected to the samples belonging to the acceptance range. Therefore, this search for neighboring pixels is based on the recursion concept; however, this concept can be simulated to obtain an algorithm without the use of a real recursion, but still using this recursion search idea. Although the PGR algorithm do not use a real recursion, the search for neighboring pixels is performed simulating a recursion. In contrast to the RGR algorithm, this paper introduces the NRGR algorithm to extract road networks in high-resolution images from remote sensing. The NRGR algorithm is suitable for software or hardware-software system's co-design solutions. The NRGR algorithm overcomes those two challenges, because the algorithm is able to deal with significant quantity of data efficiently and do not need to apply operations recursively on data while processing high-resolution images. Results achieved by experiments performed in this research suggest that the NRGR algorithm overcomes the RGR algorithm (Cardim et al. 2014) while processing images with or without the help of GPUs.

The remaining of this paper is organized as follows. Section 2 presents materials and methods. Section 3 shows and discuss achieved results. Section 4 presents conclusions.

Materials and methods

In this research, aiming at evaluating the NRGR algorithm and comparing it with the RGR algorithm (Cardim et al. 2014), the NRGR algorithm was applied on a variety of images to extract road networks and processed with or without the help of GPUs. All experiments were carried out using the RGR or the NRGR algorithms to semi-automatically (i.e. supervised by user) extract road networks in high-resolution images from remote sensing. Each experiment was performed in order to determine if the NRGR algorithm can deal with significant quantity of data efficiently without applying operations recursively.

Materials

Dataset

The experiments of this research were performed using 97 high spatial resolution images from remote sensing containing varied sizes and different road networks. The 97 images are divided in two data sets. One of the datasets, Dataset 1, contains 91 images presenting different road networks, being 25 highways, 41 urban and 25 rural ones. The Dataset 1 contain images from panchromatic band from different satellites, and also photogrammetric flights, to obtain the maximum diversity of images characteristics (Cardim et al. 2018). There are 16 images presenting urban road networks, which are part of the Vaihingen collection of the International Society for Photogrammetry and Remote Sensing (ISPRS) (Cramer 2010). The remaining 75 images have already been used and presented in (Cardim et al. 2018). The other dataset, Dataset 2, contains six panchromatic images with highways as interest features acquired by different satellites. The widths of the images vary from 512 to 2560 pixels in Dataset 1 and from 2640 to 10,000 pixels in Dataset 2. The heights of the images vary from 512 to 2560 pixels in Dataset 1 and from 1500 to 17,065 pixels in Dataset 2. All images of both datasets are monochromatic and have reference images (Ground Truth) to enable statistical evaluations of results from processing. Figures 1 and 2 show examples of road networks present respectively in Dataset 1 and Dataset 2.

Both datasets were selected to provide images with different sizes and structures for the experiments. The road networks present in both datasets images have straight or winding sections, paving or dirt roads, and sharpness or overlay contamination by shadows, vehicles or treetops cover. The images in Dataset 2 have larger sizes than the ones in Dataset 1, consequently Dataset 2 provides larger amount of data to be processed by the algorithms than Dataset 1. The varied images were chosen because they are important to verify the ability of the extraction algorithm to process images taking into consideration different levels of difficulty.

It is important to emphasize that both datasets are available for future research. Dataset 1 is free available at https://goo.gl/e33K74, except for the 16 images of the ISPRS, which are available at http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html. Dataset 2 is free available at https://goo.gl/m2xwzd.

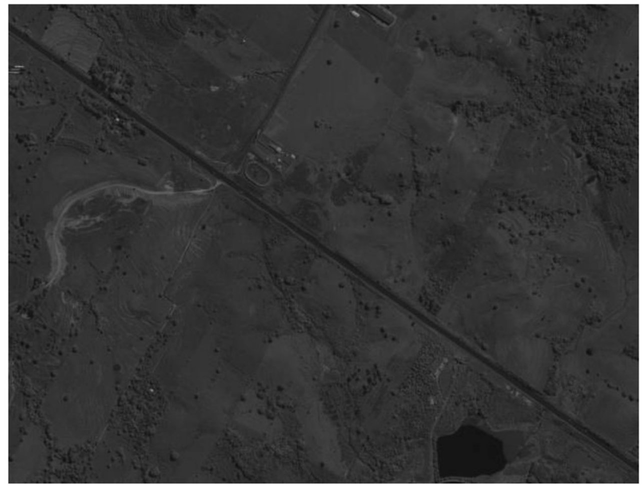

Fig. 2 Example of image from Dataset 2 containing a highway (diagonal line) as target of interest

Equipment

Two different computers were used to perform the experiments for this research. The Computer 1, was equipped with an Intel Core i7–4510 CPU of 2.00 GHz, 8 GB of RAM memory and a GPU of model GeForce GT 740 M, which has 384 processing cores with 810 MHz of clock rate. The other computer, Computer 2, was equipped with an Intel Core i5–7400 of 3.00 GHz, 16 GB of RAM memory and a GPU of model GeForce GTX 1060 6GB, which has 1280 processing cores with clock rate of 1404 MHz.

The differences between components, such as the GPUs, processors and memories, of Computer 1 and Computer 2 are important to evaluate the NRGR algorithm processing in different computational conditions. This evaluation can reveal results which are independent on the computers' components or which suffer interferences from them.

Both computers had the NetBeans IDE 8.2 with toolbox MinGW and gcc compiler for Language C, and the software Matlab R2016b installed on the operating system Windows 10. The C Language was chosen to develop the RGR algorithm because the recursive tasks of the RGR algorithm are executed faster when they are programmed in C Language than in scripts for Matlab. The Matlab was chosen to provide the communication of the NRGR algorithm with GPUs. Matlab enables fast and easy programming of algorithms for GPU platforms because of its easy-to-use interface of communication with those platforms. Matlab also provides a practical software environment for users to work with matrices. Since digital images are matrices, Matlab is suitable for programming PDI-based algorithms. In this sense, the NRGR algorithm was performed using Matlab either with or without the help of GPUs for the experiments. These experiments were performed in order to compare the time consumed by the NRGR algorithm while using or not the GPU platform.

Method

All experiments were performed using the RGR or the NRGR algorithms to semi-automatically extract road networks in high-resolution images from remote sensing. The algorithms were executed following three different strategies: RGR

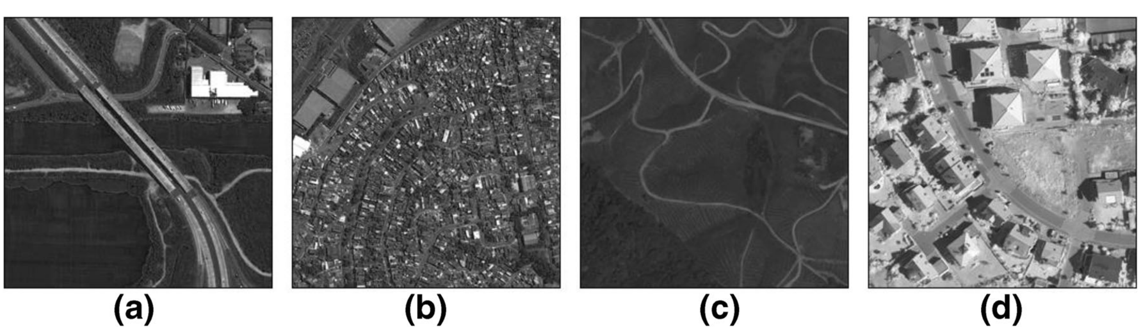

Fig. 1 Examples of road networks present in Dataset 1. (**a**) highways; (**b**) urban road networks; (**c**) rural road networks; (**d**) urban road networks from ISPRS dataset

without GPU, NRGR without GPU and NRGR with GPU. Taking into consideration that GPUs work with parallelism, RGR with GPU would be contradictory because of the recursive concept in the pixels search of the RGR algorithm. Each one of those 97 images, present in Dataset 1 and Dataset 2, was processed by both algorithms aligned with the three strategies. These three strategies were chosen because they allow the study achieving results to assess and compare the behavior of the GR algorithm for its different implementations, i.e. recursive or nonrecursive, and with or without the help of GPUs.

The execution of the RGR algorithm for the experiments of this research is summarized in the flowchart presented in Fig. 3. Regarding the RGR algorithm, at the start samples of the targets of interest were provided by the user. After, the acceptance interval, for the pixels to be detected, was calculated by the RGR algorithm. Next the points of the samples were added to a queue *q* as points belonging to the interest target and to the resultant image as detected pixels. As soon as there were points in the queue, a point *p* was taken out from *q* and its neighboring points were evaluated if they belonged to the calculated acceptance interval. In positive case, its neighboring points were added to *q* as points belonging to the target and to the resultant image as detected pixels. At the end the RGR present the resultant image.

The RGR algorithm was implemented in two different versions, one written in C language and another for Matlab software. These two different versions were implemented because they allow the study achieving results that can be compared each other in order to assess the efficiency of each version of the RGR algorithm. The two versions run in both aforementioned computers for the experiments of this study. We have found that the RGR is a high-computationally demanding and time-consuming algorithm for both versions and in any computer. However because the experiments focus on efficiency and the Matlab version of the RGR was slower than the C language version we have decided to present only the results achieved by the C language version of the RGR in the section Results and Discussions. The results achieved by the Matlab version of the RGR would be indifferent for the comparisons in the section Results and Discussions because they presented the lowest values.

The RGR is a high-computationally demanding and time-consuming algorithm, because it uses a queue to simulate recursion. Therefore, the RGR is not suitable for certain real-time applications or for computationally restricted environments. Aiming at overcoming this drawback, we developed a new algorithm, the NRGR. The NRGR is faster than the RGR to achieve the same results. It is possible because, different from the RGR, the NRGR does not use the recursion concept and can be executed in parallel mode. The execution of the NRGR algorithm for the experiments of this research is summarized in the flowchart presented in Fig. 4.

It is important to emphasize that the RGR and the NRGR are semi-automatic algorithms, since they need samples of the interest features that must be provided by the user. From those samples, the algorithms statistically define a range of values that is going to be used in the next steps. Those steps verify the pixels that attend to the defined range. Whereas the RGR performs search for pixels around the provided samples, the NRGR performs search for all pixels that belong to the defined range before defining which pixels are connected to the provided samples. Consequently, the NRGR detects all pixels of the image that belong to the calculated range. The NRGR verifies all detected targets by checking which of them have connection with the centroids of the samples. The last steps of the NRGR respectively remove all targets that do not contain any connection to the samples and present the resultant image.

The NRGR algorithm was implemented using Matlab and run in both aforementioned computers for the experiments of this study. Figure 5 presents a pseudo-code of the NRGR.

The NRGR algorithm proposed by this study, although in essence similar to the RGR algorithm, modifies the GR algorithm to enable the processing of data in parallel. On one hand, the RGR use the recursion concept to perform a search for the interest pixels. On the other hand, for the present study, recursion was set aside allowing the GR algorithm to take advantage of the parallel processing performed by GPUs. This advantage makes the NRGR algorithm proposed by this study overcome the efficiency of the RGR algorithm.

As a limitation of our study, it was not possible to investigate the NRGR implemented using C language. It was hard to implement the NRGR using C language because there is no easy interface of communication with GPUs provided by the library of the C language as well there is in Matlab. Nevertheless, the results achieved by the implementations in this study are acceptable to assess the efficiency of all versions of the algorithms investigated here. Future work will investigate if the NRGR implemented using C language can bring improvements to this study.

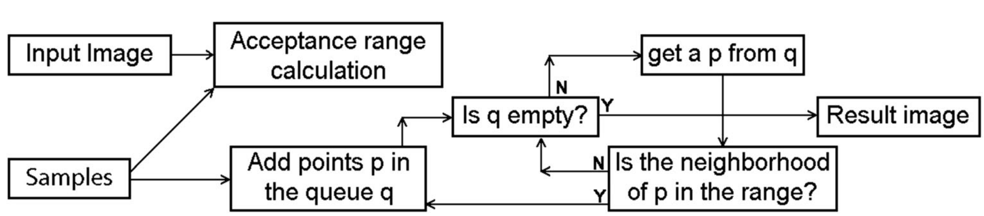

Fig. 3 Simplified flowchart summarizing the execution of the RGR algorithm for the experiments of this research

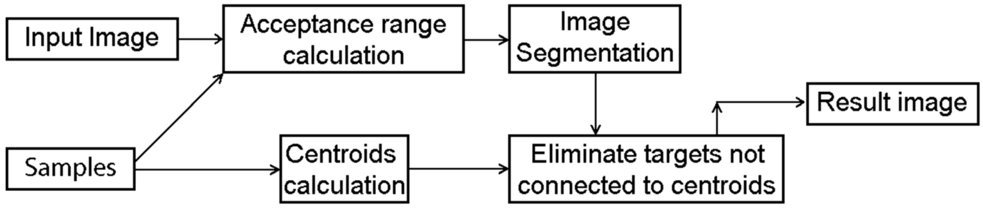

Fig. 4 Simplified flowchart summarizing the execution of the NRGR algorithm for the experiments of this research

Results and discussions

As outlined in the introduction, this paper introduced the NRGR algorithm to extract road networks in high-resolution images from remote sensing. The NRGR algorithm is a modified version of the GR algorithm that do not need to apply operations recursively on data while processing images. Therefore, our NRGR algorithm works with parallel programming in GPUs, to overcome the limitations of the RGR algorithm.

As mentioned earlier, all experiments were carried out using the RGR or the NRGR algorithms applied on a variety of images to semi-automatically extract road networks in high-resolution images from remote sensing. Each experiment was performed in order to determine if the NRGR algorithm can deal with significant quantity of data efficiently without applying operations recursively on data. In this section, we present the results achieved by the experiments commented above. The results are divided into three parts as follows: comparison of the obtained results, relative comparison of processing time, and calculation of algorithm complexity.

Aiming at validating the NRGR algorithm, we compared the images resultant from the processing performed using the RGR or the NRGR algorithms. The comparison was performed mathematically verifying if the values from all pixels of both images (from RGR and NRGR) were equals. Since we did not find any difference when comparing the resultant images taking into consideration both datasets tested, we can infer that the NRGR algorithm does not affect the quality of the cartographic feature extraction. Therefore, since the quality of the cartographic features extraction using GR algorithm was already discussed in the literature (Cardim et al. 2014, 2016) and the NRGR algorithm does not decrease this quality, we have decided not to extend this paper discussing this subject.

The gain obtained by the NRGR algorithm in comparison to the time consumed by the RGR algorithm was also evaluated. We compared the processing time necessary to apply the RGR or the NRGR algorithm on all images from the datasets available. Aiming at reducing the possible interference that may occur in the processing time due to the use of multitasking processors, each algorithm was performed twenty times (i.e. in twenty rounds) for each image. The total number of rounds was defined empirically. The final processing time for each image was found by calculating the average time consumed by the twenty rounds. Furthermore, we compared the performance of the NRGR algorithm running with and without the help of a GPU platform.

The complexities of the RGR and NRGR algorithms were calculated in order to verify if there were improvements in the NRGR in comparison to the RGR. The complexity were calculated based on the analysis of pessimistic complexity of each part of the algorithms, and in accordance with (Toscani and Veloso 2002). The achieved values regarding complexity are presented at the end of this section.

Fig. 5 Pseudo-code of the NRGR

```
Function NRGR
|
|   Input
|   |   img , img_samples : Image
|   |   max , min : Integer
|
|   Output
|   |   img_result: Image
|
|   Begin
|   |
|   |   img_aux get a white pixel for all img pixels between min and max
|   |   calculate the centroids of all samples from img_samples
|   |   repeat for each centroid
|   |   |   if the centroid position corresponds to a target in img_aux do
|   |   |   |   copy the correspondent target to the img_result
|   |
|   End
|
End
```

Figure 6 presents the comparison of the average time required to obtain the results of the extraction methodology using the RGR or the NRGR, applied to the first dataset, as well as the results obtained with the NRGR using GPU platform. The identification (id) of images from 1 to 91 relates to the Dataset 1. In addition, Fig. 6 allowed us to calculate the gain of processing time provided by the NRGR algorithm. For the Dataset 1, the NRGR obtained speed-ups values, in relation to the RGR, in the range [3.33, 11.49] and an average speed-up value around 7.4. Figure 7 presents the same comparison regarding the images from the second dataset, which has images with larger sizes than the ones in Dataset 1. The identification (id) of images from 92 to 97 relates to Dataset 2. For the Dataset 2, the NRGR obtained speed-ups values in relation to the RGR in the range [3.37, 8.49] and an average value around 7.18. The average speed-up, considering all the 97 images, is ~7.39, showing the significant gain of processing time obtained by the NRGR algorithm.

Considering that each tested image has its own dimensions, and consequently different amount of data, it is important to evaluate the amount of data processed and detected in each image tested aiming at checking if there is any relation with the processing time. Figures 8 and 9 present, respectively for the Dataset 1 and Dataset 2, the total amount of pixels in the tested image and the total of pixels detected as interest feature.

Directly comparing Figs. 6 and 8, as well as Figs. 7 and 9, it is notable that the format of the processing time graphs of the RGR algorithm is very similar to the sum of total data processed and detected by the NRGR. It suggests the existence of a high correlation between that information, i.e. the processing time of the RGR algorithm is directly affected by the amount of data processed and detected as part of the interest feature. On the other hand, the same correlation is not evident regarding the NRGR algorithm. As an example, taking into consideration Figs. 7 and 9, although the image 92 has a sum of total data processed and detected larger than the image 97, the time spent for the processing of the image 92 was smaller than the required for image 97. However, when evaluating only the total of detected pixels, it is possible to verify that despite being a smaller image, the total of pixels detected in image 97 is much larger than that detected in image 92. This fact suggests that the time of processing consumed by the NRGR algorithm is more dependent on the total of pixels detected than on the total pixels of the image.

Furthermore, to make easier the visualization of the graph of comparative times, we calculated the percentage of time required to perform the proposed NRGR algorithm, using or not the GPU platform, related to the RGR approach. Figure 10 presents this relative comparison for the Dataset 1, while Fig. 11 presents the same comparison for the Dataset 2, which contains bigger amount of data.

Figures 10 and 11 show that the NRGR algorithm obtained the extraction results with less than 20% of the time required by the RGR algorithm for the majority of the tested images, which shows the efficiency of the NRGR algorithm. On the other hand, when analyzing the previous graphs, it still difficult to assess whether the use of GPU has advantages over its non-use. Figure 12 presents the percentage of the processing time required by the NRGR algorithm to process an image

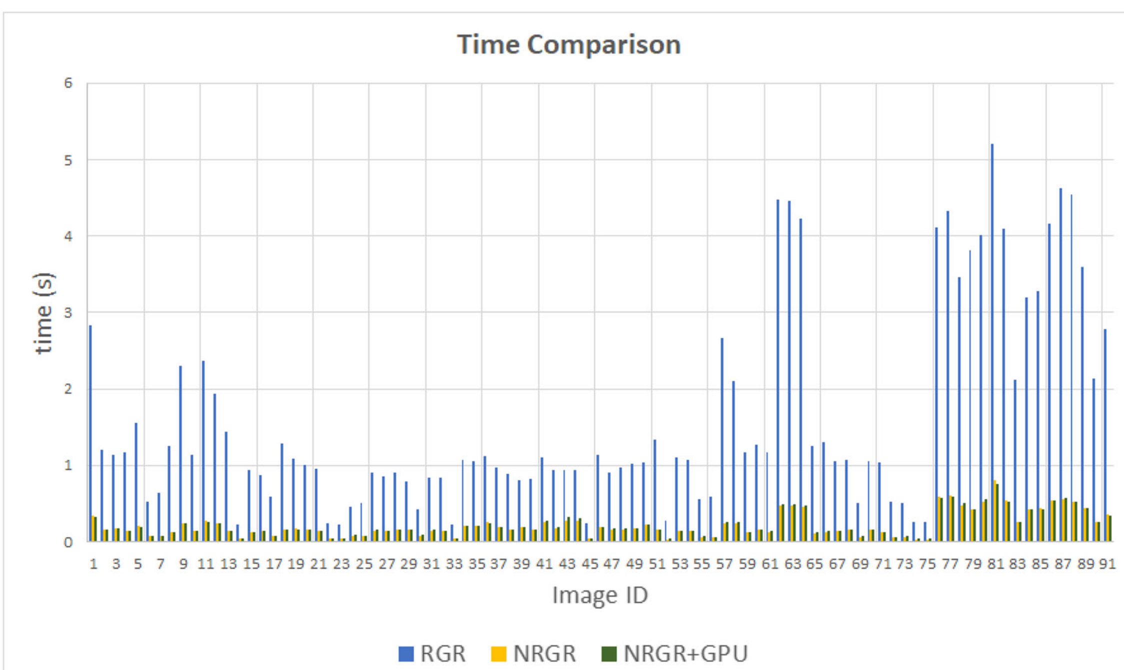

Fig. 6 Comparative time between different approaches (RGR, NRGR, NRGR+GPU) of the growth region algorithm applied to the Dataset 1

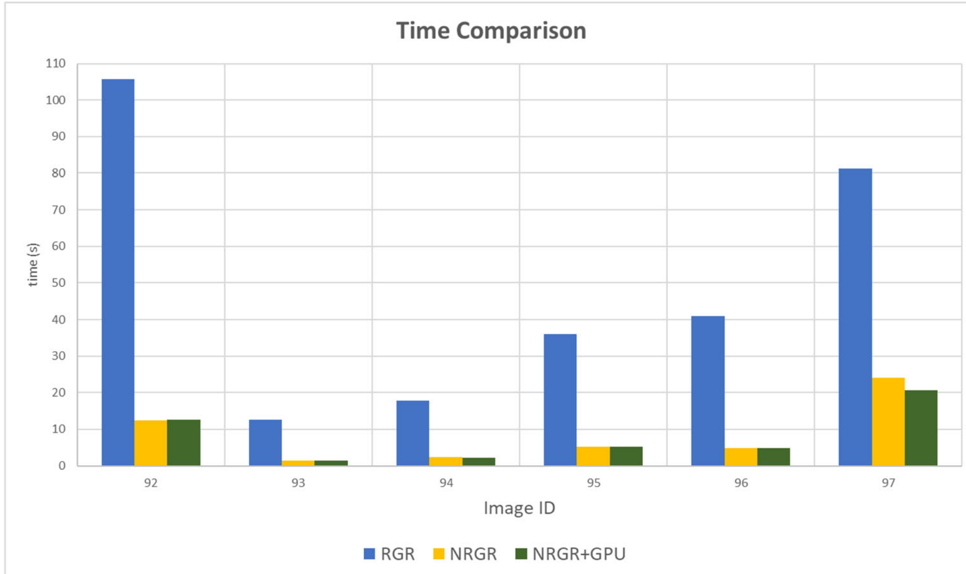

Fig. 7 Comparative time between different approaches of the growth region algorithm applied to the Dataset 2

with the help of GPU platform in comparison to the non-use of GPU, both cases applied to the Dataset 1.

Contradicting the idea that the use of the GPU platform would have significant advantages over the processing time to perform the road extraction methodology based on the GR algorithm, we can not take this conclusion when observing the graph presented by Fig. 12. It is notable that the use of the GPU platform takes more processing time than the non-use of this technology for most images. One of the reasons that may explain such situation lies on the fact that is necessary to transfer all input image data from main memory to the GPU memory before processing the image and carry out the inverse path with the resulting data, which takes long time. The data transfers between the devices are costly and, given the efficiency of the proposed approach, becomes not viable for most images from the Dataset 1. In addition, the fact that each image has different dimensions and amount of pixels to be detected makes it impossible to predict a time pattern needed to detect the interest feature. On the other hand, we have the Dataset 2, which has six images containing larger amount of data to be processed. Figure 13 presents the percentage of processing time required to perform the proposed NRGR

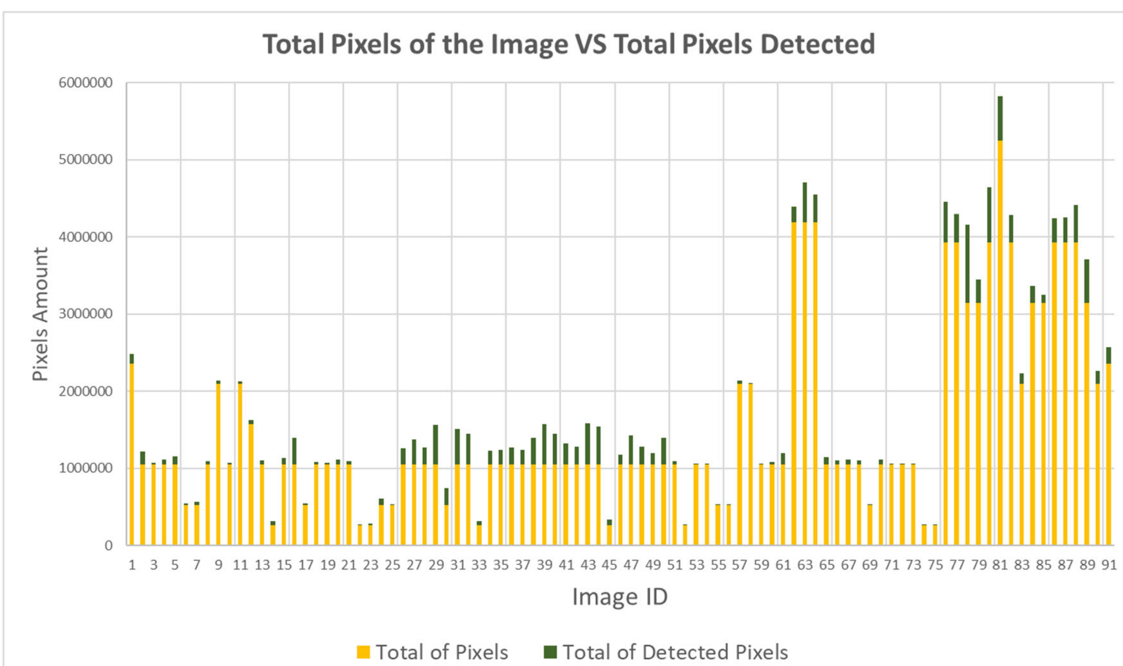

Fig. 8 The amount of data processed in the Dataset 1

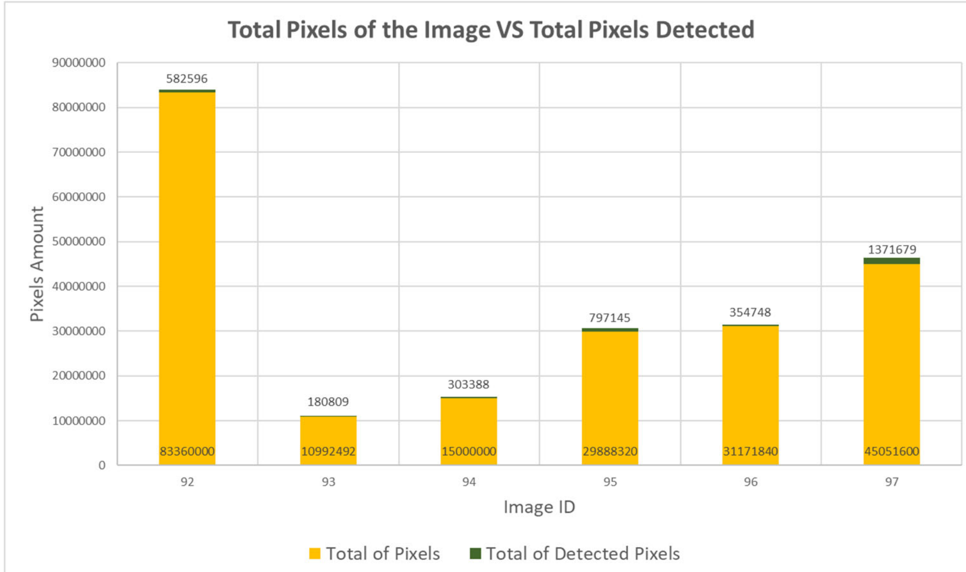

Fig. 9 The amount of data processed in the Dataset 2

algorithm when using GPU platform in comparison to its non-use applied to the Dataset 2. It is possible to notice that in cases of large amount of information the use of the GPU can become advantageous, since for some images the percentage is under 100%, e.g. it is under 86% for the image 97. Unlike what occurred for the Dataset 1, which have smaller amount of data processed, for Dataset 2 the cost of data transfer is diluted in the gain obtained in processing time.

It is worth mentioning that the tests and analyses presented in this paper depend on the technical characteristics of the computer used. It is also important to note that the tests presented until this point of the paper were performed using the first computer mentioned and described in Equipment Subsection. Considering the influence between the equipment and the image processing, the NRGR algorithm was reevaluated using a second computer, also mentioned and presented in Equipment Subsection. Using this new equipment, which has a more powerful GPU platform, the same previous experiments were carried out. Figure 14 presents the percentage of processing time required to apply the NRGR algorithm to the first data set evaluated when using the second GPU in comparison to its non-use.

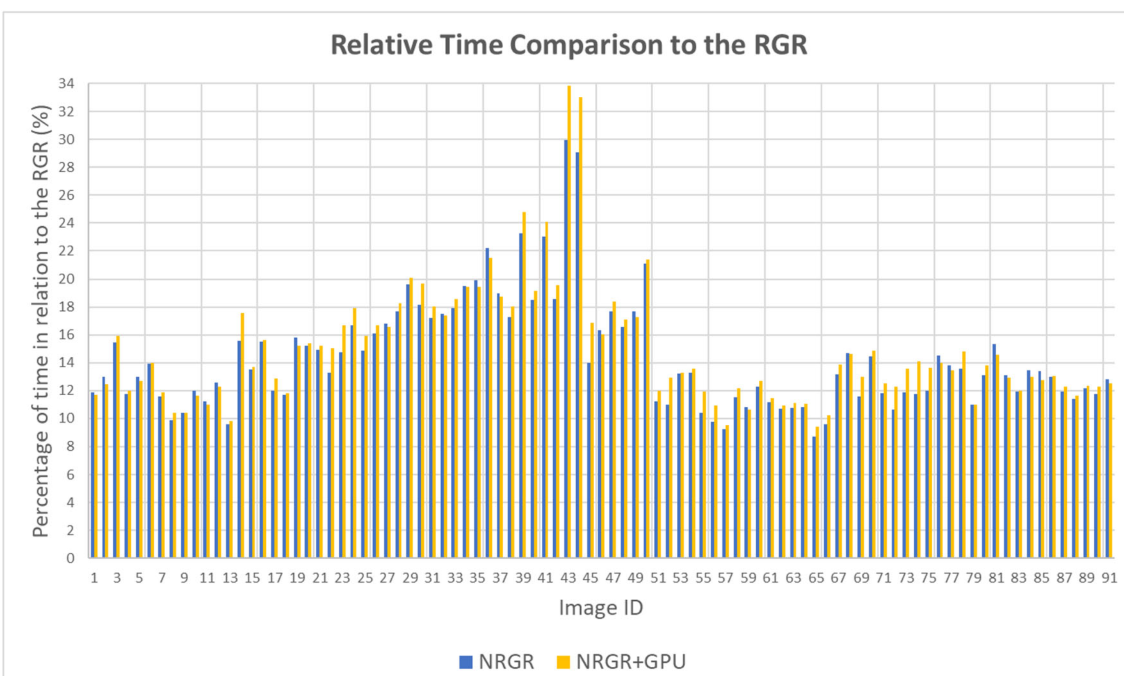

Fig. 10 Relative comparison of the different approaches of the GR algorithm tested in the Dataset 1

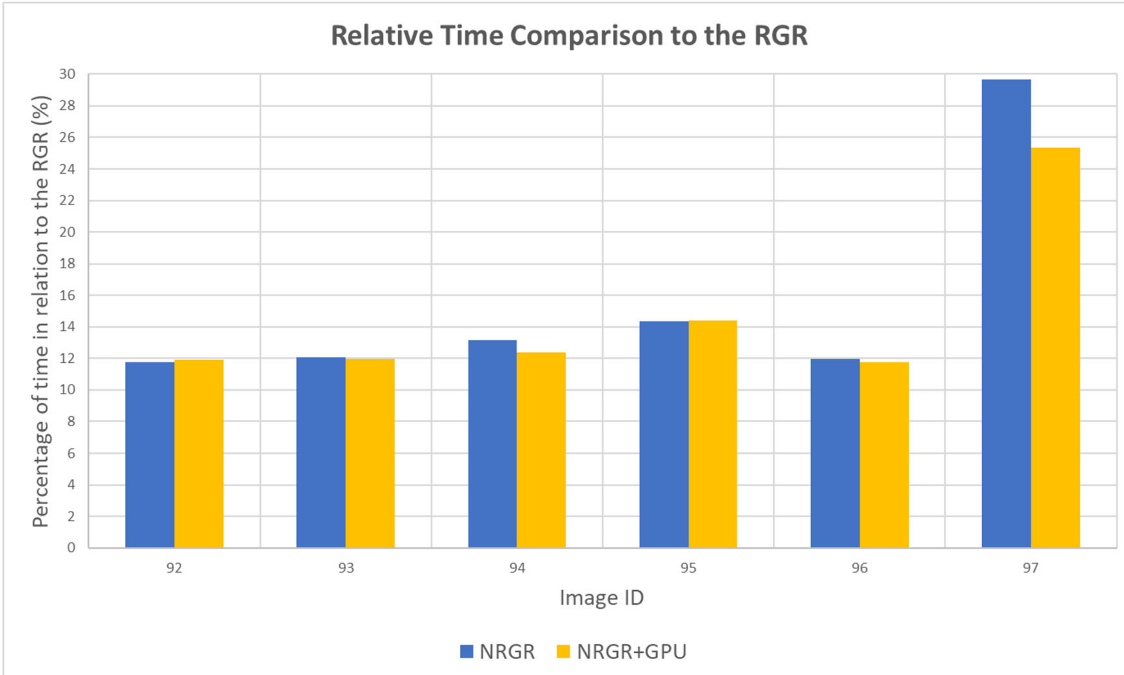

Fig. 11 Relative comparison of the different approaches of the GR algorithm tested in the Dataset 2

Contrary to Figs. 12 and 14 presents that, for most images, the use of GPU achieved shorter processing times than its non-use. It can be explained by the fact that the data from Fig. 14 was obtained with a more powerful GPU platform than the data from Fig. 12, related respectively to Computer 2 and Computer 1, both described in Subsection 2.1.2.

In addition, confirming what was already verified by the first test, for the second image data set, which has a large amount of data to be processed, the use of GPU became a factor to be considered, as can be seen in Fig. 15. However, in Figs. 13 and 15 we realize that for the image 92 this fact is not valid, since the use of GPU increase the processing time for this specific image. It can be explained by several factors, such as the fact that the total pixels detected in image 92 corresponds to less than 0.7% of the total pixels of the image, while the others images in the dataset obtained values over 1%

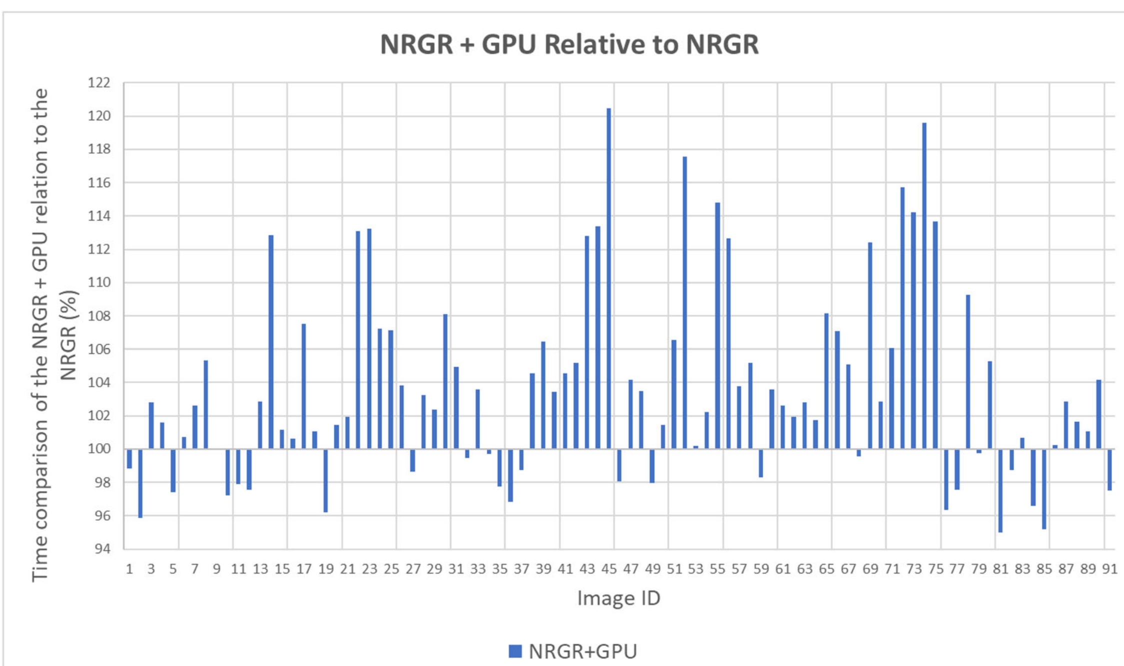

Fig. 12 Comparison of the proposed NRGR algorithm applied to the GPU relative with it non-use in Dataset 1

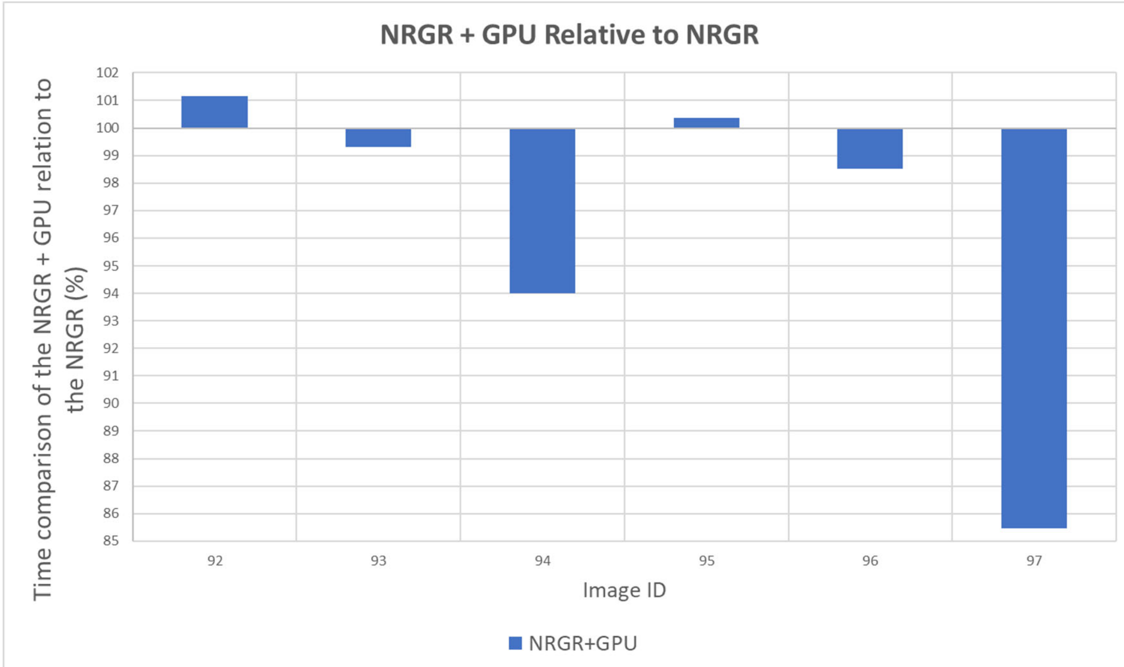

Fig. 13 Comparison of the proposed NRGR approach applied to the GPU relative with it non-use in Dataset 2

reaching, in some cases, values above 3%. In this sense, the smaller number of pixels detected in relation to the image size can negatively interfere with the use of the GPU.

In addition, the RGR and the NRGR algorithms were evaluated in relation to their respective complexities. The algorithm's complexities were calculated to evaluate if the results obtained in relation to the processing time are related to the complexity of the algorithm. The calculations of the algorithm complexity were based on the principles of pessimistic analysis (Toscani and Veloso 2002). According to these principles, the complexity of an algorithm can be obtained by the worst complexity of its parts. Therefore, the RGR and NRGR algorithms were divided into independent parts, which were evaluated separately. The complexity order calculated to the algorithms was $O(n^5)$ for the RGR and $O(n^2)$ for the NRGR. The smaller complexity order obtained to the NRGR

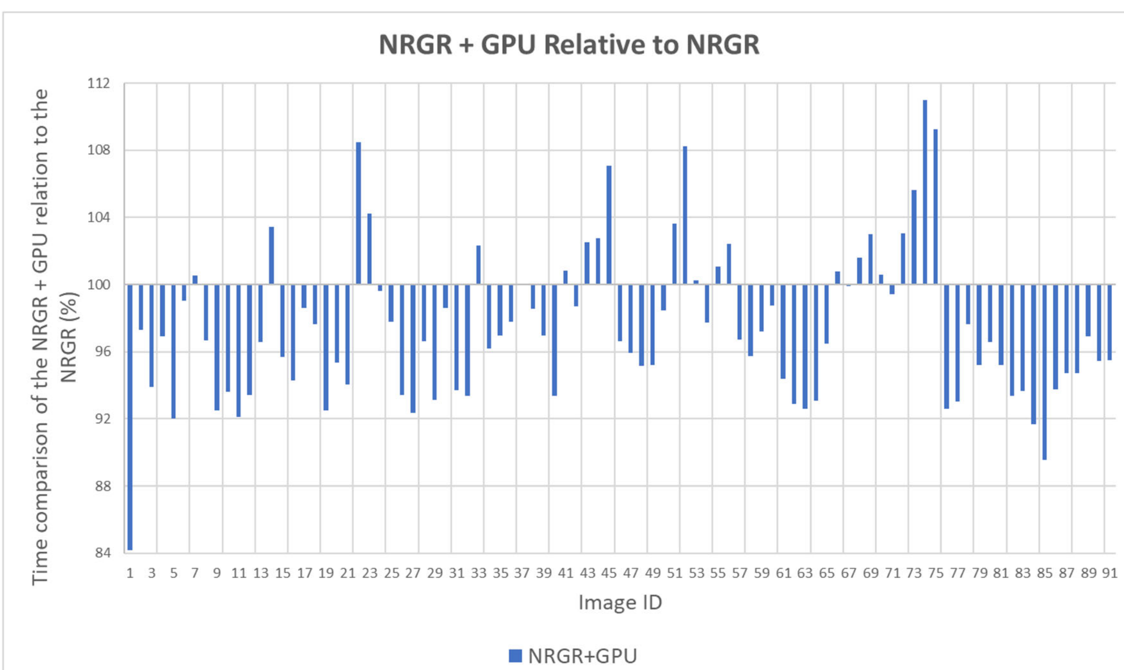

Fig. 14 Comparison of the proposed NRGR algorithm applied to the GPU of computer 2 relative with it non-use in Dataset 1

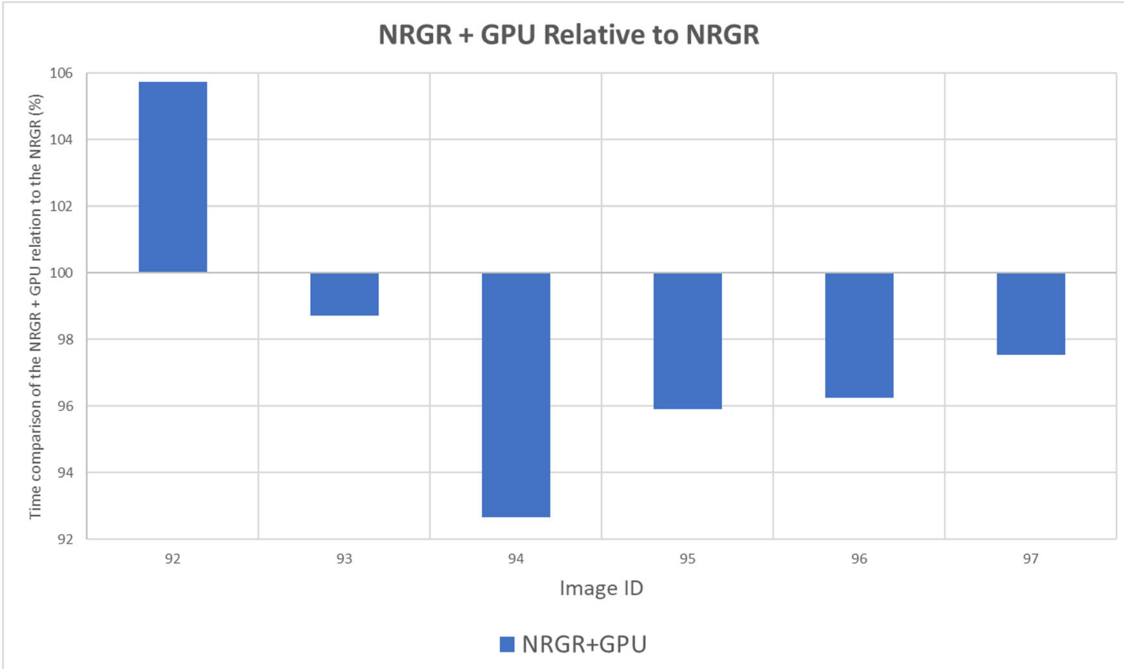

Fig. 15 Comparison of the proposed NRGR approach applied to the GPU of computer 2 relative with it non-use in Dataset 2

algorithm corroborates with the processing times obtained and previously presented for both algorithms. This fact suggests that there is significant gain and speed-up related to the time processing provided by the NRGR algorithm.

The results of this study show that the quality of the cartographic features extraction achieved using the NRGR algorithm is identical to the quality achieved by the RGR algorithm used in (Cardim et al. 2014). The results also show that the NRGR algorithm is less complex and more efficient than the RGR algorithm used by (Cardim et al. 2014). Therefore, the NRGR algorithm proposed in this study provides advantages in comparison to the RGR algorithm previously applied in (Cardim et al. 2014).

As a problem with the results of our study, it was not possible to obtain results from an implementation of the NRGR algorithm using C language. The reason is that it was hard to implement the NRGR using C language because of the absence of an easy interface of communication with GPUs in the library of the C language. Therefore, the results presented and compared in this study did not take results from an implementation of the NRGR algorithm using C language in consideration. However, the results presented in this study are acceptable to assess the quality, the complexity and the efficiency of all versions of the algorithms investigated here. Future work will try to obtain results from an implementation of the NRGR algorithm using C language and compare them with the results achieved in this study. Another problem with results was caused by the time that Matlab consumes to apply the RGR on high-resolution images. Because this time was too long, the results related to the RGR presented in this study refer to the implementation of the algorithm using C Language. Despite this, the results presented in this study indicate that the NRGR overcomes the efficiency of the RGR algorithm to extract road networks in high-resolution images from remote sensing.

Conclusions

The application of the GR algorithm has been described in the scientific literature to a variety of different purposes. The advantages of using the GR algorithm are that the algorithm provides a semi-automatic extractor of interest targets from digital images and requires just few samples of the interest targets to perform the extraction. The GR algorithm has been more frequently used to process small-sized images because the algorithm needs considerable processing time when using its original idea, which is based on a recursive search, to process large-scale images. This fact explains why we named the aforementioned algorithm as RGR in this study. Aiming at obtaining a faster GR algorithm than the RGR one, this study introduced the NRGR algorithm. The NRGR is named nonrecursive because it do not use the recursive search concept to extract road networks. In this study, the NRGR was applied to extract road networks, such as highways, urban and rural road networks, in high-resolution images from remote sensing.

To the knowledge of the authors, our proposal of the NRGR algorithm, which works with parallel programming, is the first of its kind. Our results are identical to the results achieved by (Cardim et al. 2014) regarding quality of the cartographic features extraction. Moreover, our algorithm is less complex and faster than the GR algorithms used in (Cardim et al. 2014). Therefore, our proposal broadens the advantages of using the GR algorithm.

This study contributes by presenting a novel GR algorithm (NRGR) which is faster than the RGR to detect interest targets in digital images. The NRGR was tested using two datasets with a total of 97 images and was capable to perform the features detection with less than 30% of the time used by the RGR for practically all the tested images. In addition, in some testes cases the time used by the NRGR was less than 10% of the time used by the RGR. Moreover, the presented experiments helped to evaluate the efficiency of the joint use of Graphical Processor Unit platforms (GPUs) with the execution of the proposed algorithm. Furthermore, the application of the NRGR succeed in maintaining the quality of the extractions in the resulting images when compared to other approaches.

We have confidence that the NRGR algorithm deals with significant quantity of data enabling the use of entire remote sensing images. Because the NRGR overcomes these two challenges this algorithm is much less computationally demanding and time-consuming than the RGR one.

At present, the NRGR detected road networks with efficiency in this study. The application of the NRGR to extract other interest features than road networks was beyond the scope of this study. Future work should investigate if the NRGR algorithm can also be applied to extract other interest features than only road networks efficiently.

The NRGR algorithm can be used as software or hardware-software system's co-design solutions. Moreover, based on this study, the NRGR can be applied by Cartography to develop maps of road networks, no matter if the maps focus on highways, urban or rural road networks. In future, the NRGR could be used to improve even more its applicability helping Cartography to develop maps that could include other feature of interest.

Acknowledgments The authors would like to thank São Paulo Research Foundation – FAPESP (Projects 2014/24392-8 and 2016/04553-2) for the financial support given to the development of this work. The Vaihingen dataset was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF) (Cramer 2010): http://www.ifp.uni- stuttgart.de/dgpf/DKEP-Allg.html.

References

Alali MI, Mhaidat KM, Aljarrah IA (2013). Implementing image processing algorithms in FPGA hardware. IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT) 1-5. https://doi.org/10.1109/AEECT.2013.6716446

Ali M, Clausi D (2001) Using the Canny edge detector for feature extraction and enhancement of remote sensing images. IEEE 2001 International geoscience and remote sensing symposium (IGARSS) 5:2298–2300. https://doi.org/10.1109/IGARSS.2001.977981

Bartovsky J, Holik M, Kraus V, Krutina A, Salom R, Georgiev V (2012). Overview of recent advances in hardware implementation of mathematical morphology. 20th telecommunications forum (TELFOR) 642-645. https://doi.org/10.1109/TELFOR.2012.6419292

Benkrid K, Crookes D, Smith J, Benkrid A (2001). High level programming for FPGA based image and video processing using hardware skeletons. IEEE symposium on field-programmable custom computing machines (FCCM'01) 219-226

Cardim GP, Azevedo SC, Silva EA, Dias MA, Bravo I (2016). Improvement of semiautomatic methodology for cartographic features extraction using high-resolution remote sensing images. XVII Simposion internacional SELPER 2016

Cardim, GP, Silva EA, Dias MA, Bravo I (2014) Semiautomatic Methodology for Cartographic Features Extraction Using High-Resolution Remote Sensing Images. Revista UD Y LA GEOMÁTICA, 8:53–58

Cardim GP, Silva EA, Dias MA, Bravo I, Gardel A (2018) Statistical evaluation and analysis of road extraction methodologies using a unique dataset from remote sensing. Remote Sens 10:620. https://doi.org/10.3390/rs10040620

Cheng G, Han J, Lu X (2017) Remote sensing image scene classification: benchmark and state of the art. Proc IEEE 105:1865–1883. https://doi.org/10.1109/JPROC.2017.2675998

Cramer M (2010) The DGPF-test on digital airborne camera evaluation – overview and test design. Photogrammetrie – Fernerkundung – Geoinformation 2:73–82. https://doi.org/10.1127/1432-8364/2010/0041

Déforges O, Normand N, Babel M (2013) Fast recursive grayscale morphology operators: from the algorithm to the pipeline architecture. J Real-Time Image Proc 8:143–152. https://doi.org/10.1007/s11554-010-0171-8

Herumurti D, Uchimura K, Koutaki G, Uemura T (2012). Grid seeded region growing with mixed ART for road extraction on DSM data. IEEE international conference on signal processing, communication and computing (ICSPCC) 613-617. https://doi.org/10.1109/ICSPCC.2012.6335689

Herumurti D, Uchimura K, Koutaki G, Uemura T (2013). Urban road extraction based on Hough transform and region growing. IEEE 19th Korea-Japan joint workshop on frontiers of computer vision 220-224. https://doi.org/10.1109/FCV.2013.6485491

Jeon B, Jang J, Hong K (2000) Road detection in SAR images using genetic algorithm with region growing concept. Proceedings 2000 IEEE international conference on image processing 2:688–691. https://doi.org/10.1109/ICIP.2000.899802

Kaur A, Singh R (2015) Various methods of road extraction from satellite images: a review. International journal of research 2:1025–1032

Lu P, Du K, Yu W, Wang R, Deng Y, Balz T (2014) A new region growing-based method for road network extraction and its application on different resolution SAR images. IEEE Journal of selected topics in applied Earth observations and remote sensing 7:4772–4783. https://doi.org/10.1109/JSTARS.2014.2340394

Mohammadzadeh A, Tavakoli A, Zoej MJV (2006) Road extraction based on fuzzy logic pan-sharpened IKONOS images. Photogramm Rec 21:44–60

Plaza A, Du Q, Chang Y, King RL (2011) High performance computing for hyperspectral remote sensing. IEEE Journal of selected topics in applied Earth observations and remote sensing 4:528–544. https://doi.org/10.1109/JSTARS.2010.2095495

Santiago VS, Silva EA, Nogueira JR, Leonardi F (2012) Application of PDE and mathematical morphology in the extraction of cartographic features. Journal of Earth science and engineering 2:287–291

Senthilkumar B, Umamaheswari G, Karthik J (2010). A novel region growing segmentation algorithm for the detection of breast cancer.

IEEE international conference on computational intelligence and computing research 1-4. https://doi.org/10.1109/ICCIC.2010.5705725

Toscani LV, Veloso PAS (2002) Complexidade de algoritmos: análise, projeto e métodos. Porto Alegre, Editora Sagra Luzzatto

Wang J, Shan C (2012). Extract different types of roads based on mathematical morphology. 5th international congress on image and signal processing (CISP2012) 505-509. https://doi.org/10.1109/CISP.2012.6469717

Wang W, Yang N, Zhang Y, Wang F, Cao T, Eklund P (2016) A review of road extraction from remote sensing images. Journal of traffic and transportation engineering 3:271–282. https://doi.org/10.1016/j.jtte.2016.05.005

Xiaolin L, Yufeng J, Yan G, Xiaoxue F, Weixing L (2018). Unstructured road detection based on region growing. IEEE Chinese control and decision conference (CCDC) 3451–3456. https://doi.org/10.1109/CCDC.2018.8407720

Zhu H, Cao Y, Zhou Z, Gong M (2012). Parallel multi-temporal remote sensing image change detection on GPU. IEEE 26th international parallel and distributed processing symposium workshops & PhD forum 1898-1904. https://doi.org/10.1109/IPDPSW.2012.234