Ryszard BRATUŚ, Paweł MUSIALIK, Marcin PROCHASKA, **Antoni RZONCA**
DEPHOS SOFTWARE LTD.
63 Rodziny Poganów St., 32-080 Zabierzów, Poland

Development of LiDAR Data Classification Algorithms based on Parallel Computing using nVidia CUDA Technology

Abstract

The paper presents an innovative data classification approach based on parallel computing performed on a GPGPU (General-Purpose Graphics Processing Unit). The results shown in this paper were obtained in the course of a European Commission-funded project: "Research on large-scale storage, sharing and processing of spatial laser data", which concentrated on LIDAR data storage and sharing via databases and the application of parallel computing using nVidia CUDA technology. The paper describes the general requirements of nVidia CUDA technology application in massive LiDAR data processing. The studied point cloud data structure fulfills these requirements in most potential cases. A unique organization of the processing procedure is necessary. An innovative approach based on rapid parallel computing and analysis of each point's normal vector to examine point cloud geometry within a classification process is described in this paper. The presented algorithm called LiMON classifies points into basic classes defined in LAS format: ground, buildings, vegetation, low points. The specific stages of the classification process are presented. The efficiency and correctness of LiMON were compared with popular program called Terrascan. The correctness of the results was tested in quantitive and qualitative ways. The test of quality was executed on specific objects, that are usually difficult for classification algorithms. The quantitive test used various environment types: forest, agricultural area, village, town. Reference clouds were obtained via two different methods: (1) automatic classification using Terrascan, (2) manually corrected clouds classified by Terrascan. The following coefficients for quantitive testing of classification correctness were calculated: Type 1 Error, Type 2 Error, Kappa, Total Error. The results shown in the paper present the use of parallel computing on a GPGPU as an attractive route for point cloud data processing.

Keywords: point cloud classification, parallel computing, normal vectors.

1. Introduction

One of the most important elements of point cloud processing is classification. An innovative point cloud classification algorithm was proposed as part of a project run by the DEPHOS Software Company: "Research on large-scale storage, sharing and processing of spatial laser data." The project ran in the years 2012-2015 and was financed by the European Union. The general outcomes of the project were described in a paper by (Będkowski et al. 2015). The new approach to point cloud classification is based on the use of parallel computing with graphics processors as well as the use of advanced testing techniques of the geometry of the base cloud in order to assess which points belong to which classes in accordance with the LAS system. The algorithm used by DEPHOS Software is based on the following types of points: (1) so-called low points, (2) terrain, (3) buildings, (4) low, medium, and high vegetation (BĘDKOWSKI et al. 2015). evaluated the algorithm in terms of efficiency and output quality by comparing the algorithm-classified data with data classified using Terrasolid software as well as manual verification.

2. Parallel processing and point cloud processing

Parallel processing using graphics processors based on nVidia CUDA technology allows for efficient processing of scanning data (Będkowski et al. 2015). This paper presents the outcomes of the use of this technology with an array of different processors, with most processors experiencing significant improvement thanks to the use of nVidia CUDA and appropriate algorithm architecture. One of these algorithms, created for the purpose of assessing computing power in graphics processors designed to process point clouds, was the classification of points. Most analytical operations can be completed in parallel. The software works using commands recorded in a driving macro. Computations performed in the process of classification that run successfully in parallel mode include the loading and saving of LAS files, both 2D and 3D decomposition of data, calculation of normal vectors, identification of ground points, ground expansion, and the classification of buildings and vegetation.

3. Existing point cloud classification techniques and technologies

3.1. Classification methods

The classification of point clouds produced by airborne laser scanning (ALS) is one of the most vital technical issues associated with this method of measurement. Ever since ALS data have become available, the assignment of points to specific classes has become a condition for their full utilization.

A variety of different point cloud classification methods can be found in the research literature and these methods are systematized in a variety of ways (Sithole, Vosselman, 2004; Borkowski 2005; Ural, Shan, 2016). A classification system that covers almost all available methods and provides the most current information on the subject of classification is discussed by Meng et al. (2010) who place methods into several categories: segmentation/cluster, morphology, directorial scanning, contour, TIN, interpolation. The paper discusses methods in detail for the lower level of the classification system (key methods section):

- Segmentation/cluster method class includes segmentation based on the smoothness constraint, segmentation-based classification, segment-based terrain interpolation.
- Morphology class includes dual rank filter based on dilation and erosion, morphologic filter based on geodesic dilation, progressive morphologic filter.
- Directorial canning class includes bidirectional labeling and hybrid multi-directional ground filtering.
- Contour class includes active contour and active shape model as well as active shape model based on the energy function.
- TIN class includes local curvatures of point measurements and the adaptive TIN model.
- Interpolation class includes the following: iterative robust interpolation, multiscale curvature algorithm based on TPS interpolation, facet model, linear prediction.

Today the subject of point cloud classification is associated with the identification of urban sites (Rottensteiner et al. 2014), data segmentation (Lari et al., 2011), and semantic data analysis (Niemeyer et al., 2014). The latest methods utilize neural networks and advanced analyses of full waveform data (Zhou et al. 2016). In addition, a classification method based on the use of normal vectors (Jeong, Lee 2016) was introduced at the 2016 ISPRS Congress in Prague. This method is being developed at the University of Seoul, independently of DEPHOS Software.

3.2. Operating rules for a typical algorithm

A typical point classification approach is based on several geometric principles resulting from the very nature of points as representatives of a given class. The software most commonly used to process scanning data is called Terrascan – made by Terrasolid (2016). However, the algorithm used in this software is based on an iterative adjustment of the triangle network (Axelsson 1999, 2000) and makes certain mistakes, which then have to be manually corrected. Typical classification errors affect points found on bridges, overpasses, high trees that overlook the roofs of all sorts of buildings and low structures, points found on the walls of buildings, balconies, and small elements of rooftops such as antennas and chimneys (Sithole, Vosselman 2002; Meng et al. 2010).

The classification algorithm is produced on the basis of a set of consecutive commands or macros. Once points in overlapping areas found between arrays are classified, a sequence of commands with parameters is used to classify erroneous points, terrain, built structures, and vegetation.

4. Method description

The main idea behind the presented method is utilizing parallel computing for classification purposes. A 3D point cloud consists of a large number of similar entities, which makes computations done for these entities particularly good targets for parallelization. The current version of the method yields a classification for the following classes: ground, low/med/high vegetation, buildings, low points. The algorithm follows these steps:

- Normal vector calculation,
- Height index calculation for each point,
- Ground initialization,
- Ground growth,
- Building initialization,
- Building growth,
- Vegetation classification.

To achieve accurate results, these steps must be followed in the given order.

4.1. Regular grid decomposition

One of the key problems in parallel computing is choosing a reliable method of dividing the data set into suitable subsets. Each subset is then used by one computation thread. Too large data sets tend to significantly decrease performance up to a point where they nullify all the advantages gained from parallelism. On the other hand, too small subsets may lead to a situation where a single computation thread will contain too little information, and will yield inaccurate results.

For the task of classifying aerial 3D point clouds, we have chosen a modified version of the Regular Grid Decomposition (RGD) method described in Będkowski et al. 2013. This method divides 3D point clouds into N x N x N bins, where N is the number of bins in each direction. Each point of the cloud is assigned to the bin it falls into and the number of neighboring bins. These bins are then used for calculations related to that point. By changing the number of bins, we can adjust the subsets used by each thread.

This approach proved to be successful for 3D point clouds gathered by a ground laser scanner. Aerial 3D point clouds, however, differ from ground point clouds in density and general scale of the area covered – large differences in size in each dimension. In order to cope with this problem, RGD was used instead of a constant number of bins – constant bin size in each dimension.

4.2. Normal vector calculation

In the first step of the classification process, we calculate a normal vector for each point. The method used is parallel implementation of PCA/SVD similar to the one described in Będkowski et al. (2013). At this point a preliminary classification in 3 support classes is performed: linear, planar, spherical. Vector calculations occur only if the number of points in the neighborhood exceeds a threshold value.

4.3. Height index calculation

In the second classification step, each point is assigned a height index value in range 0-1. The RGD for this calculation is done in 2D – bins are only created in the XY plane, despite the height. A point is assigned a value of zero if it is the lowest point in its bin, and a value of one if it is the highest point.

4.4. Ground initialization

This step finds the "seed" points ground. The main assumption used herein is that the lowest local point in a given area must be the ground. Every point whose height index is lower than a threshold value is considered a seed for the ground. To avoid including low points, only points with normal vectors are considered in this step.

4.5. Ground growth

In the ground growth step, the ground is iteratively expanded basing on the nearest neighborhood of points. This is done as follows:

- For each point P, we find the nearest ground point.
- If the ground point is closer than X, we check if P lies on the plane created by the ground point and ground point's normal vector.
- If the condition from the previous point is met, then we check additional conditions such as maximum allowed height difference, whether P is the last reflection point, whether the normal vector of P is roughly similar of the direction with the normal vector of the ground plane.

This step may be executed multiple times using different parameters as well as different numbers of iterations in order to analyze various types of scanned areas.

At the end of this step, we perform a finalization of the search for low points. An average estimation model is created for each bin that includes ground. Hence, every point that lies lower than the model – beyond a certain threshold value – is considered a low point.

4.6. Building initialization

The building initialization step can only be done after ground points have been identified. This step searches the subset of planar points for those that meet certain conditions:

- Are they at least H m above the ground?
- Feature a low number of non-last reflection points in the nearest neighborhood.
- Do not possess any ground points in the nearest neighborhood.

4.7. Building growth

Building growth uses points found in the previous step in order to expand buildings. As with ground growth, the algorithm iteratively checks certain conditions based on nearest building points:

- Is there no ground under the checked point (within a given radius)?
- Are there any other building points within that given radius?
- Does the checked point fit in the plane of the nearest building point?
- Is the number of non-last deflection points lower than the given threshold value?

This condition is forfeited for the last few iterations to include the edges of buildings.

This step ends with a special building closure sub-step that takes care of adding walls to the rooftops.

4.8. Vegetation classification

Vegetation classification steps check all unclassified points and attempt to assign one of three vegetation classes to them. Unlike in the case of buildings, in this step, each positively classified point has to have ground points underneath. The distinction between low, medium, and high vegetation is made based on distance from the ground. The conditions that need to be checked are:

- Does the given point have ground underneath?
- Are there no building points within a given radius?
- “Most” (defined by a parameter) neighbors do not fit the plane created by the checked point?
- Are there vegatation points in the neighborhood? This does not apply to the first iteration.

The final step is aimed at dividing points between ground and low vegetation and concentrates on points near the ground level. Flat areas are divided into sectors. For each sector, a planar model of low vegetation and ground is created based on already classified points. Unclassified points are then assigned to one of these classes based on the distance from these models. Each model has a weight assigned (parameter) that allows to choose which class should be favored in this step.

5. Testing the algorithm

Two types of tests were conducted in order to evaluate the studied algorithm. The first test was an efficiency test and the second test was a functional accuracy test, which was a quantitative test where appropriate coefficients were computed that served as indicators of agreement with data and the model method. In addition, qualitative analysis was used to manually compare selected profiles of point clouds at locations that generate typical classification errors.

5.1. Efficiency test

Terrascan software (made by Terrasolid) was used as a reference algorithm due to its popularity. Test data were selected in four distinct areas: (1) woodland areas, (2) agricultural areas, (3) rural built-up areas, (4) urban and industrial areas. Ten data blocks were prepared for each category used in the ISOK project, which is a project that covers all of Poland, with a point density ranging from 4 points per square meter in agricultural and woodland areas to 12 points per square meter in cities. The data were classified using reference software as well as the LiMON algorithm and processing times (without data loading times and result saving times) were recorded. The computer used to test classification times using Terrascan software included an AMD FX4300 QuadCore 3.8 GHz processor and 8 GB of RAM. The parallel computing process was run on a computer equipped with an Intel i5 processor and an nVidia Titan graphics card with 6 GB of RAM. The configuration of the two computers was different for the processes being compared due to optimal processing conditions in classic array-type processing mode versus parallel processing mode.

Figure 1 to 4 show the results of the efficiency test in the form of a dependence of processing time (in seconds) on the number of points in selected files belonging to specific land use categories. In most cases, the LiMON algorithm is characterized by shorter processing times than Terrascan. Yet, it is important to note that process optimization and the optimization of the number of iterations will lead to shorter classification times.

5.2. Quantitative accuracy test

The quantitative test consisted of a calculation of parameters expressing agreement between the tested point clouds and reference clouds. The parameters of agreement calculated in this case were: Type 1 Error, Type 2 Error, Total Error, and the Kappa Coefficient. Type 1 Error is an error where ground points are classified as some other type of entity. Type 2 Error is an error where points belonging to other entities are classified to be ground (Marmol, Jachimski 2004; Sithole, Vosselman, 2002, 2003). Total Error is calculated for every class (LU et al., 2008). Finally, the Kappa Coefficient serves as a parameter that quantitatively describes the degree of agreement between classification results using two or more methods (Silván-Cárdenas, Wang, 2006).

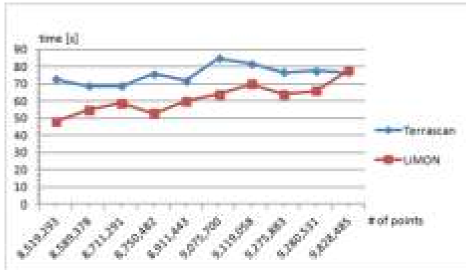


Fig. 1. Test results for files for agricultural areas show processing times (in seconds) in relation to the number of points

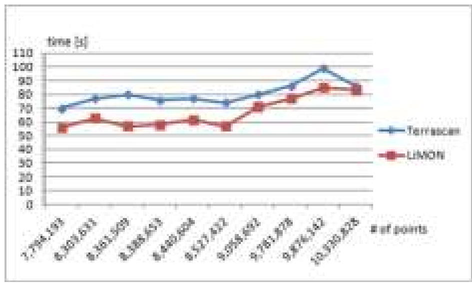


Fig. 2. Test results for files for rural areas show processing times (in seconds) in relation to the number of points

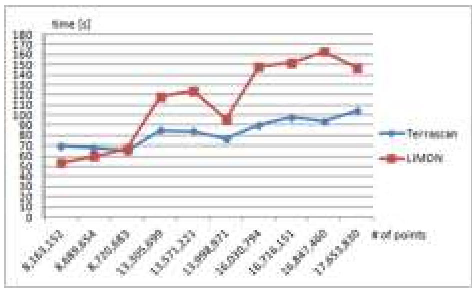


Fig. 3. Test results for files for woodland areas show processing times (in seconds) in relation to the number of points

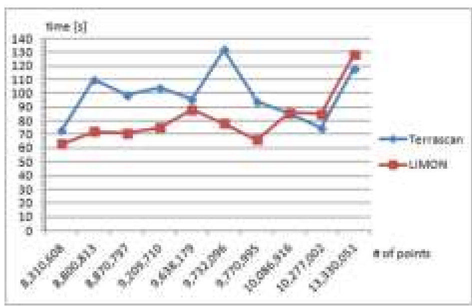


Fig. 4. Test results for files for urban areas show processing times (in seconds) in relation to the number of points

Tests were conducted on data samples provided by the ISOK project. A total of 15 data blocks with a total surface area of 1.2 km^2 were selected as the samples. Each data sample featured different characteristics. Point clouds were classified using LiMON software (dataset no. 1) and Terrascan software. Manual corrections were made for dataset no. 2. No manual corrections were made for dataset no. 3; only Terrascan was used in the classification process. Dataset no. 2 was treated as a reference dataset.

The next step consisted of 2 types of comparative analyses. The first analysis compared dataset no. 1 (classified using LiMON) with dataset no. 2 (reference data).

Each coefficient described earlier was calculated: Type 1 Error, Type 2 Error, Kappa, Total Error. The resulting values are shown on graphs provided in the paper. Fig. 5 lists the kappa coefficients for each studied class in relation to the studied data samples (1 to 15). It is readily apparent from the data that the greatest degree of agreement can be observed for high vegetation, while the largest discrepancies can be observed for buildings. Lowest mean values of kappa were calculated for low vegetation and the ground.

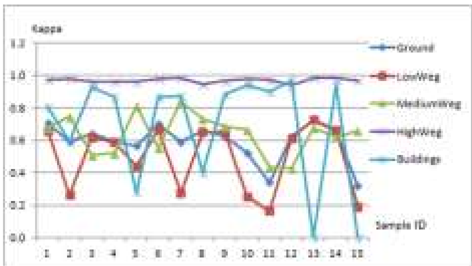


Fig. 5. Kappa coefficients calculated for datasets no. 1 and 2

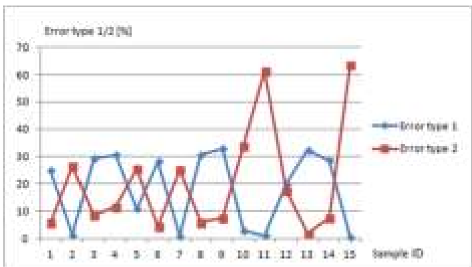


Fig. 6. Errors no. 1 and 2 (%) for datasets no. 1 and 2

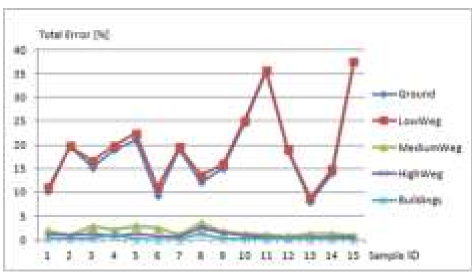


Fig. 7. Total Error (%) for datasets no. 1 and 2

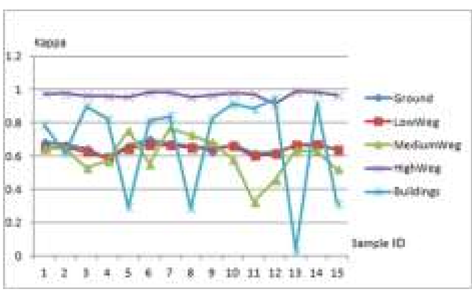


Fig. 8. Kappa coefficients calculated for datasets no. 1 and 3

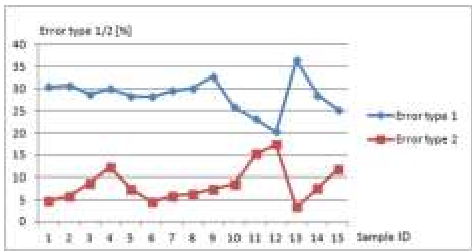


Fig. 9. Type 1 Errors (%) and Type 2 Errors (%) for datasets no. 1 and 3

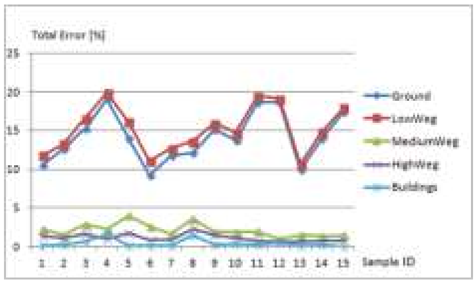


Fig. 10. Total Error (%) for datasets no. 1 and 3

Fig. 6 shows Errors no. 1 and 2 calculated only for the ground class in relation to other classes due to the definition of these errors, as provided earlier in the paper.

This graph shows errors that marginally exceed 30%, except for sample no. 11 and 15, where a substantial number of points were classified as ground.

The Total Error coefficient is shown for each studied class on Fig. 7.

The results shown here confirm outcomes produced via calculations of Type 1 and Type 2 Errors. The above analysis is designed to employ a relatively broad dataset to show the degree of agreement between results produced by the LiMON algorithm and a reference algorithm.

The second analysis concerned a practical comparison of raw (unadjusted) classification results produced by both LiMON and Terrascan software. This second analysis along with qualitative analysis is designed to show whether LiMON can be used in production at this stage of software development.

Figures 8, 9, and 10 show the following coefficients: Kappa, Type 1 Error, Type 2 Error, Total Error.

The analysis of kappa values shows significant similarity to an analogous result in the first analysis on datasets no. 1 and 2. Substantial dispersal of agreement values was observed for the building class. The highest dispersal range was noted for high vegetation, while the lowest for medium vegetation. The pattern followed by ground curves and low vegetation curves is quite similar. This diagram can be used to conclude that some buildings can become classified as medium vegetation and vice versa. The solution of this problem is provided by qualitative analysis later on in the paper.

The graph for errors 1 and 2 exhibit a large disproportion, with many more ground points being classified erroneously as other classes (Type 1 Error) than vice versa. In the context of Fig. 8 (kappa), the ground was classified in the majority of cases of a Type 1 Error as low vegetation. This type of conclusion may be drawn given the similarity of the kappa graph for ground and low vegetation.

Total Error assumes higher values for the ground and low vegetation compared with other classes. The pattern for these two classes is quite similar due in part to the agreement on which point is really ground and which point is really low vegetation. In some cases, even an experienced observer may have a serious problem distinguishing the two in the process of manual point cloud classification. This also confirms the conclusion on Type 1 Errors.

5.3. Quality test

Quality testing included an array of comparisons between cloud profiles classified using Terrascan and LiMON versus RGB clouds, which was designed to yield a more accurate interpretation. The work of other researchers as well as our own previous production work was used to determine entity type and configuration for entities that tend to be problematic in the classification process. Our own work in this area is based on statistical analysis – which errors need to be corrected and where these errors are located following automated classification using Terrascan.

Four difficult types of entities were identified in the study area:

Chimneys and rooftop antennas

Figure 11a shows a vertical cross section of a cloud classified using LiMON, while Figure 11b shows the same section classified using Terrascan.

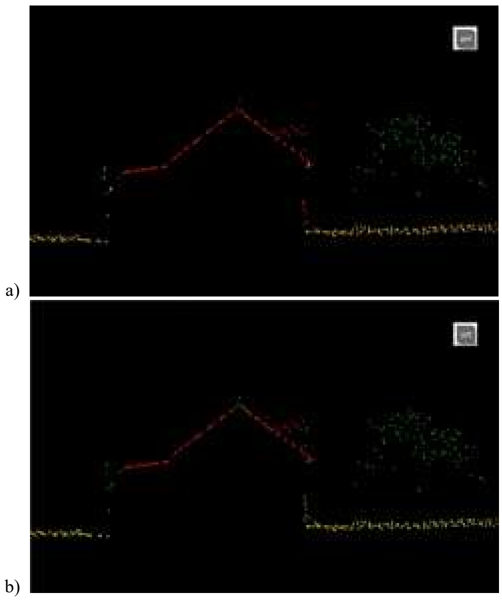


Fig. 11. Cross section of a single-family house – cloud classified using: (a) LiMON, (b) Terrascan

LiMON properly classified the ridgeline of the roof and did not classify points (white points on the right) whose identity is difficult to ascertain.

Roofline of a residential building

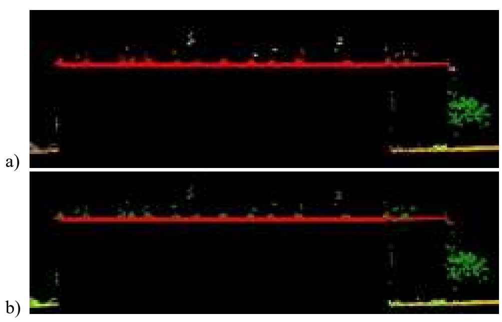


Fig. 12. Longitudinal cross section through a residential building – cloud classified using: (a) LiMON, (b) Terrascan

Visible chimneys and antennas were classified by Terrascan as high vegetation, while LiMON was more accurate in classifying these as elements of a building or elements that cannot be unequivocally placed in a specific class (white points). This represents a major advantage of the algorithm based on the analysis of normal vectors.

Walls of a building

Another example of points that are difficult to classify is walls of buildings. Figures 13 and 14 indicate that the LiMON algorithm is more accurate in classifying points on the walls of buildings as building points and not high and medium vegetation points.

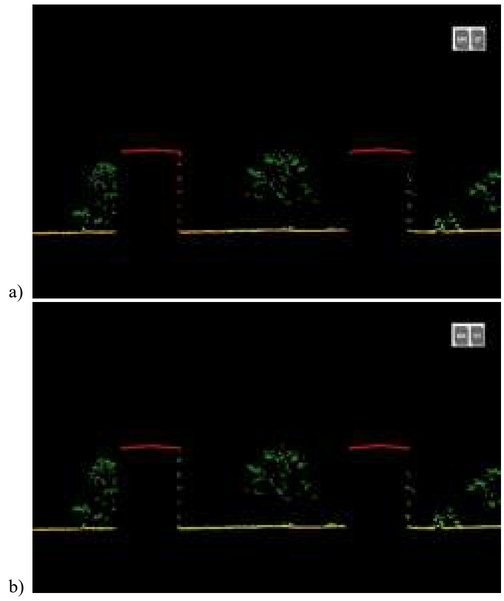


Fig. 13. Cross section of a residential building – cloud classified using: (a) LiMON, (b) Terrascan

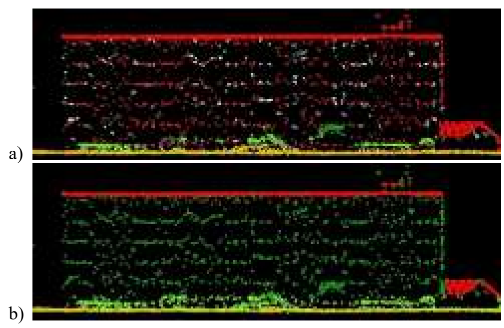
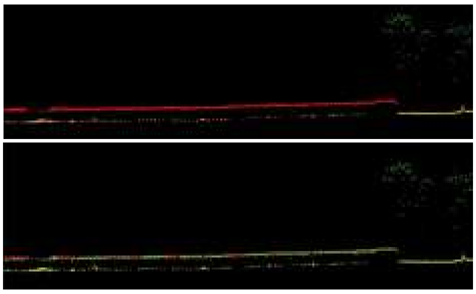


Fig. 14. View of the wall of a residential building - cloud classified using: (a) LiMON, (b) Terrascan

Roofs of low built structures such as garages

Typical built structures that are not tall also constitute a challenge for classification algorithms. Figures 15 and 16 show (respectively) a longitudinal cross section and a regular cross section through a group of garages characterized by a height that is not typical for buildings (2.5 m). Terrascan classifies the roof areas of such low buildings are vegetation.



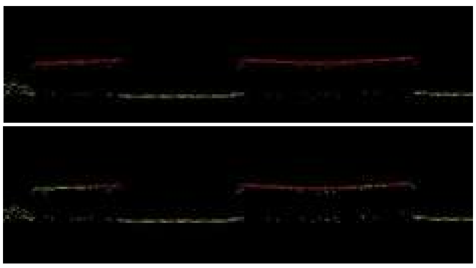
a)

b)

Fig. 15. Longitudinal cross section of a group of garages – cloud classified using: (a) LiMON, (b) Terrascan.

Trees with a flat, dense crown

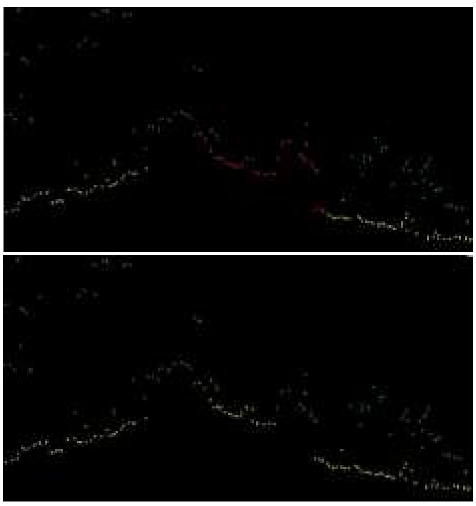
In the case of trees with a very dense crown, no reflections reach the ground, as laser impulses are not able to penetrate through to its surface. If these tree crowns happen to not differ substantially in terms of overall geometry, then neither software program yields the expected result, with Terrascan yielding slightly more erroneous classifications (Fig. 17).



a)

b)

Fig. 16. Section of a group of garages – cloud classified using: (a) LiMON, (b) Terrascan



a)

b)

Fig. 17. Cross section of dense tree cover – cloud classified using: (a) LiMON, (b) Terrascan

In summary, the tests described above were able to show that LiMON software produced higher quality results compared with classification results produced by a reference software program. Visual control of the studied entities confirmed the results of the quantitative tests described in this study. However, the most common type of error produced by the Terrascan algorithm or the erroneous classification of points on roofs and walls of buildings as well as on garage roofs was eliminated in the algorithm used by LiMON thanks to the use of normal vectors. This is very important from the point of view of production practices in large surface area projects such as Poland's national-scale ISOK project. It also substantially limits costs associated with the need to manually edit point clouds classified using the software most commonly used by businesses or Terrascan.

6. Conclusions

The paper describes an innovative approach to point cloud classification. A brief review of other methods is provided in the introduction, with the described method resting on two bases: (1) IT base or parallel processing on graphics cards using nVidia CUDA technology, (2) mathematical base or the innovative use of normal vectors to interpret scanned points.

The paper presents the proposed and implemented classification method in comparison with the major achievements in this area of research. It also describes the results of three different tests used to test the proposed algorithm and a reference algorithm for comparative purposes.

The first test was an efficiency test, which showed that the LiMON algorithm is a better solution than the reference algorithm, but without any major breakthrough. LiMON yields shorter processing times, but this advantage is not substantial enough to warrant the purchase of this particular software, as opposed to other types of software.

The second test was a quantitative test of classification accuracy, which was conducted in the form of two comparative analyses based on parameters commonly used to evaluate classification algorithms. The first analysis was designed to determine the absolute accuracy of the classification with reference to classification data produced by Terrascan and manual classification. The second analysis was designed to show the quantitative discrepancy between results not subjected to manual editing and produced by both LiMON and Terrascan.

The result of the first analysis shows a satisfactory level of agreement between the data produced. The largest discrepancies were noted for buildings (large kappa range for different data blocks), while relatively low discrepancies were noted for vegetation and the ground. This is also confirmed by the values of Type 1 Errors and Type 2 Errors. The calculated value of Total Error confirms the link between potential discrepancies between the ground and low resolution, which is highly likely to be true. One typical example is that of a plowed field (15 cm furrows) and a field unevenly overgrown with grass or some type of crop. This type of situation yields a high likelihood of error and results must be treated with a dose of caution regardless of the method used.

The second analysis showed that raw classification results confirm the outcome above, implying that the largest amount of uncertainty is associated with the classification of points in the ground and low vegetation categories (see Total Error). A certain degree of differentiation appears for medium vegetation, but the largest discrepancy range between results for specific data blocks has been shown for the class designated "buildings." These conclusions may be drawn most readily based on diagrams of kappa values.

The third test was fairly typical and simple in terms of methods, but rooted in practical experience, and focused on the identification of places where errors tend to appear and then need to be manually corrected following automated classification using Terrascan software. The result of this test was very good for the new algorithm.

Further research and more work on the new algorithm are needed and will focus on the problem identified in the case of high vegetation, which does not produce a reflection off the ground. Additional work will also be done on the optimization of the code in the pursuit of greater efficiency.

Rapid implementation and commercialization of the results presented herein are planned.

The research was carried out as part of the following European Union project: "Research on large scale storage, sharing and processing of spatial laser data" no. UDA-POIG.01.04.00-12-125/11-00.

7. References

[1] Axelsson P.: Processing of laser scanner data - algorithms and applications. ISPRS Journal of Photogrammetry and Remote Sensing, 54(2), 1999, s. 138–147, 1999.

[2] Axelsson P.: DEM generation from laser scanner data using adaptive TIN models. The International Archives of the Photogrammetry and Remote Sensing, 33 (B4/1), pp. 110–117, 2000.

[3] Będkowski J., Bratuś R., Prochaska M., Rzonca A.: Use of parallel computing in mass processing of laser data. Archiwum Fotogrametrii, Kartografii i Teledetekcji 1, 45-59, Poland, 2015.

[4] Będkowski J., Majek K., Nuechter A.: General purpose computing on graphics processing units for robotic applications. Journal of Software Engineering for Robotics, 4(1), pp. 23-33, 2013.

[5] Borkowski A.: Filtracja danych lotniczego skaningu laserowego z wykorzystaniem metody aktywnych powierzchni. Roczniki Geomatyki, Tom III, Zeszyt 4, s. 35-42, Poland, 2005.

[6] Jeong J., Lee I.: Classification of lidar data for generating a high-precision roadway map. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B3, 2016.

[7] Lari Z., Habib A., Kwak E.: An adaptive approach for segmentation of 3d laser point cloud. ISPRS Workshop Laser Scanning 2011 Calgary, Canada 29 – 31 August 2011.

[8] Lu W.L., Little J.J., Sheffer A., Fu H.: Deforestation: extracting 3d bare-earth surface from airborne LiDAR data. In: Proceeding of: CRV 2008, Fifth Canadian Conference on Computer and Robot Vision, 28–30 May 2008, Windsor, Ontario, Canada, pp. 203–210, 2008.

[9] Marmol U., Jachimski J.: A FFT based method of filtering airborne laser scanner data. Int. Archives of Photogrammetry and Remote Sensing, ISSN 1682-1750, Vol. XXXV, part B3, 2004.

[10] Meng X., Currit N., Zhao K.: Ground filtering algorithms for airborne lidar data: a review of critical issues. Remote Sens 2010, 2, 833–860, 2010.

[11] Niemeyer J., Rottensteiner F. Soergel U.: Contextual classification of lidar data and building object detection in urban areas. ISPRS Journal of Photogrammetry and Remote Sensing, 87(2014), pp. 152-165., 2014.

[12] Rottensteiner F., Sohn G., Gerke M., Wegner J.D., Breitkopf U., Jung J.: Results of the ISPRS benchmark on urban object detection and 3D building reconstruction, ISPRS Journal of Photogrammetry and Remote Sensing 93 (2014) 256–271, 2014.

[13] Silván-Cárdenas J. L.,Wang L.: A multi-resolution approach for filtering LiDAR altimetry data, ISPRS J. Photogramm. Remote Sens., vol. 61, no. 1, pp. 11–22, 2006.

[14] Sithole, G., Vosselman, G.: Comparison of filtering algorithms. In ISPRS Commission III, Symposium 2002 September 9 - 13, 2002, Graz, Austria, 2002.

[15] Sithole G., Vosselman G.: Report ISPRS: Comparison of filters, http://www.isprs.org/commission3/wg3, 2003.

[16] Sithole, G., Vosselman, G.: Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 59 (1–2), 85–101, 2004.

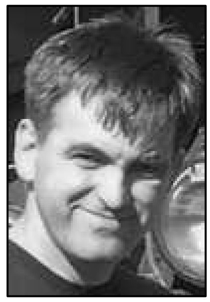
[17] Terrasolid: TerraScan User's Guide, 2006.

[18] Ural S., Shan J.: A min-cut based filter for airborne lidar data, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B3, 2016.

[19] Zhou M., Li C.R., Ma L., Guan H.C.: Land cover classification from full-waveform lidar data based on support vector machines. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B3, 2016.

Received: 24.08.2016 *Paper reviewed* *Accepted: 03.10.2016*

Ryszard BRATUŚ, MSc, eng.



A graduate of the Faculty of Mining Surveying and Environmental Engineering, AGH in Cracow (1995); speciality: industrial surveying, geoinformatics. In professional and academical work develops software for geodesy, surveying, photogrammetry. Specialization in Geographic Information Systems, dedicated for municipalities, mining and marine industries. Interested in numerical processing of large datasets, spatial algorithms and spatial analysis.

e-mail: ryszard.bratus@dephos.com

Paweł MUSIALIK, MSc, eng.

A graduate of Faculty of Mechatronics, Warsaw University of Technology (2009), specialty: parallel processing, robotics, artificial intelligence, mission planning for unmanned vehicles. Currently independent researcher.

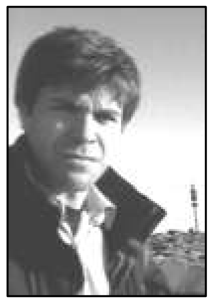
e-mail: pjmusialik@gmail.com

Marcin PROCHASKA, MSc, eng.

MSc Eng. degree in Geodesy and Cartography of the Faculty of Mining Surveying and Environmental Engineering, AGH in Cracow (AGH). He is founder of ADRAM Mapping System and DEPHOS Software Ltd., R&D director of DEPHOS Group. He has 20 years of experience in software design and development for mapping, digital photogrammetry and laser scanning.

e-mail: mprochaska@dephos.com

Antoni RZONCA, PhD, eng.



PhD in technical sciences (2008), a graduate of the Faculty of Mining Surveying and Environmental Engineering, AGH in Cracow (2001), specialty: geoinformatics, photogrammetry and remote sensing, adjunct in the Department of Geoinformatics, Photogrammetry and Environmental Remote Sensing, experienced in surveying of historical sites, aerial mission planning and quality control of photogrammetric and scanning data, and development of software for photogrammetry and laser scanning

e-mail: arzonca@dephos.com