Volume 12, No. 2, March-April 2021



International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

USING RPM – GMM FOR TPS INTERPOLATION IN MCC ALGORITHM

Nguyen Thi Huu Phuong Department of Software Engineering, Faculty of Information Technology Hanoi University of Mining and Geology Hanoi, Vietnam

Abstract: The author mentions the TPS interpolation improvement using RPM - GMM, where EM is used to initialize the parameters for the Gaussian model. Using EM algorithm in model calculation in TPS-RPM will help solve the symmetry point in this point matching process. With the idea of using Robust point matching (RPM), this is the optimal search technique in the spatial transformation of the point cloud. RPM is very powerful in removing noise and exceptions. This is an algorithm that uses iteration with 2 steps of calculating probability and updating is quite similar to EM algorithm. Therefore, the authors have researched and tested the TPS interpolation based on RPM-GMM. With the test data set in Bac Ninh city of Quang Ninh province, the classification results with MCC when using TPS-RPMGMM compared to MCC version 2.1 announced in 2018 achieved higher results and non-ground class are classified into more detail: buildings and vegetation. *Keywords:* MCC, RPM, EM, LiDAR point cloud, TPS interpolation

I. INTRODUCTION

MCC (Multiscale Curvature Classification) is an algorithm built by two authors Jeffrey S. Evans and Andrew T. Hudak in 2007. This is the algorithm used to classify LiDAR data in a forest environment. Algorithm classify LiDAR point cloud into two classes: ground and non-ground. MCC is an algorithm based on the multiscale repeating principle to classify the LiDAR reflection signal that exceeds the threshold of surface curvature. Multiscale approach will determine the deviation of the points to be classified with the average surface and gradually remove the points on the surface from the ground group [1].

LiDAR data includes surface elevation measurements and was collected through aerial topographic surveys. The file format used to capture and store LiDAR data is a simple text file and is called "x, y, and z", where x is the longitude, y is the latitude and z is the altitude, These points are saved in .LAS or .LAZ file format [2]. LiDAR technology is able to collect information about the Earth's shape and characteristics through its solar cells. Each point in the geode contains a large amount of information which is used to create a 3D model of the Earth's surface as well as the objects on the ground, when color information is available.[3].

MCC algorithm begins by determining whether the points are on the single or last reflected laser as well as the first reflection from the plant. Next, the MCC calculates the average surface from the characteristic ground points using a Thin Plate Spline (TPS) interpolation, and then the MCC corrects the loop through a filter core. TPS is an interpolation that allows the assessment of the state between points, compliance with the input data, and control of how far the sample points affect the estimated surface [1]. The size of the filter is calculated by the value of the parameter NPS (Nominal Point Spacing) or the ratio s, usually using 3 sizes: 0.5NPS, NPS, and 1.5NPS.

Implementing the MCC algorithm the authors define a vevtor Z (s) containing the coordinate value of all LiDAR points. This vector will be used in a raster surface iteration using TPS iteration with a proportional resolution λ . A 3x3

filter goes through all the filters to find a new vector x (s). The scale domain 1 is the iteration model set with the model parameters running until convergence. λ and t will be user defined [1]. In the iterations of the algorithm, the threshold curvature (t) is the parameter that most influences the results, because this is the parameter used to compare and points to the ground or non-ground class.

On research related to MCC algorithm in LiDAR point cloud classification in Vietnam, researches on developing algorithms to classify the lidar point cloud published are still limited.

In the world, published studies on MCC in LiDAR data classification, most notably the study of authors Jeffrey S. Evans and Andrew T. Hudak using MCC method in classifying LiDAR feedback signal in the forest environment. The MCC algorithm introduced by the authors uses TPS iteration to remove the points that are not in the ground group so that DEM of the investigation area can be established. TPS is an iteration that allows to assess the state between points, compliance with input data, and control how far away the sample points affect the estimated surface [1]. To evaluate the algorithm, the authors used datasets obtained from the North of Idaho. The results show that the MCC minimizes the error, while still exists a high ratio of ground points and high reliability of the provided points. Meanwhile, the authors Wade T.Tinkham and Hongyu Huang had a comparative study on the efficiency of two algorithms MCC and BCAL in LiDAR data classification to establish DTM. When evaluating the overall performance of MCC and BCAL with each different resolution, the threshold of error is different. At a resolution of 1m, ANOVA showed no significant difference between MCC and BCAL. At a resolution of 0.5m, BCAL is superior to MCC with different types of coverage. After that, the authors compare and create DTM of the two algorithms without difference. But when combining the two algorithms MCC and BCAL, the DTM generation has higher accuracy. After the experiment, the authors showed that the BCAL algorithm is suitable for areas with dense point density, continuous vegetation. In places where the slope is constantly changing MCC will have higher accuracy [4]. In the study [5], the MCC-RGB method is

introduced, the classification update steps use the support vector machine classifier (SVM) to distinguish the vegetation and the ground point from the derived features, from point color. The algorithm identifies low ground zero points, especially in high-density photovoltaic data generated from drone surveys, an emerging data source that can challenge directions. The authors found that color-based grading updates eliminate fallen trees, low canopy, and brush, often requiring less repetitions than the standard MCC method on large, confidential datasets. extreme high. This shows that simple machine learning techniques can enhance point cloud data filtering for difficult geomorphologic applications such as soil micro-tube imaging or point grading on plant slopes. With the study [6], the authors compared the four algorithms WLS (Weighted Linear Least Squares), MCC, PMF (Progressive Morphological Filter), PTIN (Progressive Triangulated Irregular Network) in the classification of ground points. Over the course of the experiment, the four algorithms properly removed the ground objects in the forest environment, because the terrain models represented very small divergences. In which, algorithm MCC and WLS are used for aviation LiDAR data (ALS - Airborne LiDAR Scanning) and provide accurate terrain number modeling (DTM).

TPS is a spline-based technology for interpolation and data smoothing. TPS is an interpolation that finds a "minimal flexed" smooth surface passing through all given points. The result of the interpolation is still the height of the surface, the number of limit points is still dependent on the number of control points, the more control points it takes to run the surface interpolation. From this problem, the authors made improvements to the TPS to reduce the number of control points while ensuring the accuracy of the classification results. The TPS iteration improvement studies have been published by the study authors in articles [7], [8], [9], [10], [11], [12] that when Reducing the number of TPS control points will be more efficient and smoother, the results are proven with different data sets.

From there, the authors proposed the idea to reduce the number of TPS control points and using RPM - GMM define TPS interpolation.

II. METHOGOLOGY

Based on the studies of TPS interpolation and MCC algorithm in part 1, the author proposes a method to classify the LiDAR point cloud with MCC has shown in figure 1.



The implementation steps of the TPS iteration improvement process based on TPS-RPM algorithm using EM algorithm (TPS-RPMGMM) are performed as follows:

A. To find the value of s, t author used the NPS concept to search for s, t values that match the data set.

To find NPS for the data set, the author has researched and applied the principle of the Voronoi curve to find the point density and NPS value for it. A Voronoi graph usually divides the data set into regions, each consisting of points closer to a certain point than others, and so we can say that each point in the point cloud will have a certain distance. with a certain point. To find the Voronoi graph point cloud based on Fortune scan line algorithm (from bottom to top), the algorithm is performed as follows:

We have a set of points $P = \{p1, p2, ..., pn\}$

- Step 1: Locate points based on point coordinates $(\boldsymbol{x},\,\boldsymbol{y})$ - symbol s

- Step 2: Define a transformation * (P) with * (P) = (px, py + d (P)), define the sweep line T, Rs is the area surrounding position s, Bij is the boundary area between the two positions si, sj. Consider sm as positions with the minimum value of the y-coordinate in terms of x. Set Q = P - {sm}, create vertical borders with symbols B0s1, s2, B0s2, s3,..., B0sm-1, sm. The scan line T is a set of values {* (Rs1), B0s1, s2, * (Rs2), B0s2, s3,..., * (Rsm-1), B0sm-1, sm}

- Step 3: if si is position in * (P) use transform * P to find the frequency of occurrence of region Rs in T. If si belongs to T create a new boundary B and remove any point is located in the boundary area, and add any points at the intersection of the upper and lower areas of the scan line.

- Step 4: If si is the vertex of Voronoi, consider si position to the left or right of the scanning line, create new boundary B and remove or add points in the boundary area according to the position of the point.

- Step 5: Record these locations and create a path to partition the location

- Step 6: Save these borders in the scan line to create a Voronoi graph

After finding the Voronoi plot of the LiDAR point cloud, the NPS is calculated according to the point density and the distribution of points in space according to the following steps:

- Arrange the distance and density values from low to high
- In which, the distance (Point Spacing PS) and point density (Point Density PD) are calculated according to the following formulas:

$$PD = 1 / (cell Voronoi)$$

$$PS = MAX(\Delta p_{ki} \in P_k)$$

In which, Δp_{ki} - the distance between two points, P_k - the neighborhood point set of pk. Calculate the percentile of the PS and PD values, and assign this percentile value = NPS.

B. The TPS iteration relies on the control point to make the surface independent from external influences

The TPS iteration relies on the control point to make the surface independent from external influences. In addition, we can consider the TPS as an approximation for the interpolated surface of the control points and make this surface as close as possible to the actual surface.

Suppose we have a set of points $P = \{P1, P2, ..., Pn\}$, to find the set of control points of P we can do:

There is matrix CT^0 containing all points Pi belonging to P which are the control points of P

Divide CT0 into a series of CTk matrices, with the coupling of control point pk level k with control point pk + 1 of the k level + 1 is calculated by the formula:

$$p_{k+1} = CT_k p_k$$

When k = n, record the points in the matrix CT_k , call them the control points

C. - Initialize parameters for GMM model using EM:

 μ (Zi, Θ), Σ , δ , β (schedule parameter) - β is initialized with a value of approximately 0. Because with the point match problem, the match point parameter is a potential variable to be found. And the interpolation is the parameterization based on the point-matching parameter - Θ , so instead of initializing the basic parameters for the model as usual, we need to add the value of this match parameter. The parameter Θ is represented by the likelihood function:

$$\odot = \arg \max_{\theta} L(\theta) = \arg \max_{\theta} \sum_{i=1}^{n} \log \left(P(Z_i | \theta) \right)$$

with $p(s_i|\theta)$ as the probability distribution function is shown by GMM model by the formula: $P(Z_i) = \sum_{k=1}^{K} \prod_k Gaussian(Z_i|(\mu, \odot), \Sigma_{k_i}, Ck)^{\beta}$

During the calculation and in order to be able to estimate the point match parameters, it is necessary to find the symmetry between the observed data and the model data. This is necessary when the points of symmetry in the observed model are often lost data and it can be retrieved in the implicit variable model with the EM algorithm [15]. And many studies have proven that, EM can converge to the maximum of local capacity based on observed datasets. Therefore, using the EM algorithm in calculating the model in TPS-RPM will help solve the symmetry point in this point matching process.

The implementation steps of the TPS iteration improvement process based on TPS-RPM algorithm using EM algorithm are performed as follows:

- Step 1: Initialize the parameters for GMM model: μ (Zi, O), Σ , δ , β . Because with the point match problem, the match point parameter is a potential variable that needs to be found. And the interpolation is the parameterization based on the point-matching parameter - O, so instead of initializing the basic parameters for the model as usual, we need to add the value of this match parameter.

The parameter $\boldsymbol{\Theta}$ is represented by the likelihood function:

$$\odot = \arg \max_{\theta} L(\theta) = \arg \max_{\theta} \sum_{i=1}^{n} \log \left(P(Z_i | \theta) \right)$$

In which, $p(s_i|\theta)$ as probability distribution function is represented by GMM model according to the formula:

 $P(Z_i) = \sum_{k=1}^{K} \prod_k Gaussian(Z_i | (\mu, \odot), \Sigma_{k, j} \operatorname{Ck})^{\beta}$

Where k is the number of model elements, β is the scheduling parameter, $\sum_k \Pi_k = 1, Z_i$ the height of the point

- Step 2: Calculate the probability function based on the initialized parameters of the model

- Step 3: Update parameters for the model as in Chapter 3 formula, variable parameters according to the formula:

$$\odot^{i+1} = \arg \max_{\bigcirc} \sum_{i=1}^{n} \log \left(P(Z_i, Y_i, \odot, |\theta) \right)$$

Covariance matrix according to the formula:

$$\Sigma^{i+1} = \arg \max_{\Sigma} \log P(Y_i, Z_i, \odot^{i+1}, \mu_i, k, \beta)$$

- Check the convergence of the algorithm with threshold is:

$$\varepsilon = \sqrt{\frac{1}{n} \sum_{i=1}^{n} r_i^2} * 10^{-8}$$

D. Using TPS in MCC for LiDAR point cloud classification:

After the definition of TPS and Z (s), it is classified as with the original MCC algorithm. A 3x3 filter goes through all the filters to find a new vector x (s). The scale domain 1 is the repeating model that is set with the parameters of the running model until convergence. A curvature tolerance t will be added to x (s), and the points will be layered into the non-ground layer using the condition: *If* Z(s) > c *then non-ground point*

III. EXPERIMENT

A. Dataset

To test the accuracy of the proposed method, the author selected a densely populated area of Quang Ninh province, with 3.047.656 points. This is a densely populated area, with an area of about 10 km2, high population density, roads and many trees.

Collected data will be conducted to remove the noise. Due to the influence of the external environment or the laser rangefinder malfunction, the resulting point clouds always contain noise points, including high and low margins. Both of these boundaries may affect the assumed distribution when displayed; especially the low margins can have a great influence on the final filter result. Therefore, these boundaries are removed during data processing. Removal methods are based on their height. The author used the algorithm k-NN (k Nearest Neighbor) and according to the research of author Z.Hui et al to remove the boundary point by comparing its altitude with that of the neighboring points. Proximity, which seeks to remove points. The point removed is the point where the altitude change is too large before and after comparing it with the k nearest neighbor [13].

The selected point is the amplitude if their altitude changes greater than the threshold that can be automatically computed according to the equation [14]:

C

$$\begin{cases} Z_{th} = 3 * Z_{std} \\ Z_{std} = \sqrt{\frac{1}{k} \sum_{k=1}^{k} (Z_k - Z_{mean})} \\ Z_{mean} = \frac{1}{k} \sum_{k=1}^{k} Z_k \end{cases}$$

In which, Zth is the threshold to detect the peripheral point; Ztd is the normal distribution of the neighborhood point; Zmean is the mean altitude of the neighboring points; Zk th height of the kth point - the point under consideration of the periphery. The point cloud after noise removal is shown in Figure 2.



Figure 2. LiDAR point cloud after noise removal

B. Experimetn result

 Selection of parameters s, t for MCC based on NPS is carried out with the step of creating Voronoi graph for the data set shown in Figure 3, the distribution of data point is shown in Figure 4 and PD of data in Figure 5.





Figure 4. Distribution of points

Х	Y	Z	Z	
420996.5	4467690	140.3	140.3	0.005153
420999.6	4467690	141.8	141.8	0.00493
420999.1	4467689	143.17	143.17	0.004729
420995.9	4467690	143.73	143.73	0.004648
420995.8	4467689	139.49	139.49	0.005274
420993	4467690	148.41	148.41	0.003989
420986.4	4467690	140.9	140.9	0.005063
420983.2	4467690	139.12	139.12	0.005329
420980.2	4467690	142	142	0.0049
420949.9	4467690	128.33	128.33	0.006967
420981.9	4467690	139.18	139.18	0.00532
420985.2	4467690	140.73	140.73	0.005088
420991.8	4467690	151.53	151.53	0.003576
420994.7	4467689	143.98	143.98	0.004611
420997.8	4467689	142.86	142.86	0.004774
420998.2	4467688	142.29	142.29	0.004858
420995.2	4467689	143.84	143.84	0.004632
420995	4467688	139.17	139.17	0.005322
420992.2	4467689	148.57	148.57	0.003968
420992	4467689	141.66	141.66	0.00495

Figure 5. PD of dataset

Sorting the value of PD from large to small to take its percentile, we have the value of NPS = 3.02, so we choose the value s = 3 (according to section 2.3.2), the value t = 1 / 2s = 0.5.

Select the number of control points for TPS: with a matrix CT0 with 943,844 rows, divided into k = 3.047.656 rows, through the tests the condition $p_{k+1} = CT_k p_k$, we have 107,365

points satisfy the conditions. Thus, the number of control points of TPS will be chosen to be 107,365, we will define TPS through these 107,365 control points.

Perform classification with s = 3, t = 0.5 and defined TPS. Results shown in Table 1.

Table I. Classification result with proposed MCC

Class	Ground	Building	Vegetation	Unclassified
Number	1.725.320	232.477	889.323	200.536
of				
points				

After grading, 1,725,320 points were classified into the ground layer. The algorithm ran with 12 iterations at SD1, 9 iterations at SD2, and 7 iterations at SD3. To evaluate the accuracy of the proposed algorithm, the student compared the results of the MCC algorithm (Evans). The results are shown in Table 2.

 Table II.
 Comparison of classification results between proposed MCC and MCC (Evans)

	Precision	Recall	F1	Convergence
Proposed				
МСС	93,2%	92,8%	92.9995	0.00025
МСС				
(Evans)	92%	92,4%	92.1995	0.0023

The test results are evaluated on Precision, Recall, F1 measures, showing that the proposed MCC algorithm has better classification results, the run time is shortened, this shows the number of control points selected. In addition, two parameters of the proposed MCC-TPSRPMMM are selected automatically according to the input data set, which will reduce the errors made by the user's choice in the sorting process and the convergence of the algorithm is better than the original MCC algorithm.

Data after classification was used by the author to establish a DEM / DSM / 3D model of the test area. The models are shown in the figure 6, 7, 8.



Figure 6. DEM with 1m resolution



Figure 7. DSM with 1m resolution



Figure 8. 3D model of point cloud

IV. CONCLUSIONS

MCC algorithm is an algorithm that has been researched and applied by scientists around the world to the problem of classifying the scattered LiDAR point cloud. The algorithm uses TPS interpolation to classify points in the ground layer or not on the ground. This is the commonly used interpolation to model scattering data. However, the interpolation accuracy and time depend on the number of control points of the surface. In the paper, the authors have proposed to improve the iteration based on RPM - GMM to find the number of control points and surface interpolation in order to reduce the interpolation runtime while retaining the accuracy of the algorithm. With TPS-RPMGMM, the authors tested on the data set in Quang Ninh for better accuracy and runtime than MCC (Evans) algorithm published in 2018. However, with TPS- RPMGMM still needs more research to help the algorithm have better convergence.

V. ACKNOWLEDGMENT

. The research is sponsored by the project of issuing CT. 2019.01.07

VI. REFERENCES

- [1] ESRI, "ArcGIS for Desktop," 2016. [Online]. Available: http://desktop.arcgis.com/en/arcmap/10.3/managedata/las-dataset/what-is-lidar-data-.htm. [Accessed 12 6 2019].
- [2] Jeffrey S.Evans, Andrew T.Hudak, "A multiscale curvature algorithm for classifying discrete return LiDAR in forested environments," IEEE, vol. 45, pp. 1029-1038, 2007.
- [3] Nguyễn Thị Hữu Phương, Đặng Vắn Đức, Nguyễn Trường Xuân, "The MCC algorithm in LiDAR point

cloud classification," in Hội thảo Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông, Quy Nhơn, 2017.

- [4] Wade T.Tinkham, Hongyu Huang, "A comparison of two open source LiDAR surface classification algorithm," Remote Sensing, vol. 3, pp. 638-649, 2011.
- [5] Sare.R, Hilley G.E, "Supervised, color enhanced MCC of LiDAR and structure from Motion point clouds for geomorphic applications," in AGU Fall Meeting, 2019.
- [6] Carlos Alberto Silva, Carine Klauberg, Ângela Maria Klein Hentz, Ana Paula Dalla Corte, Uelison Ribeiro, Veraldo Liesenberg, "Comparing the Performance of Ground Filtering Algorithms for Terrain Modeling in a Forest Environment Using Airborne LiDAR Data," Floresta e Ambiente, vol. 25, no. 2, pp. 1-10, 2018.
- [7] Lunds University, "Matematikcentrum," [Online]. Available: http://www.maths.lth.se/matstat/kurser/fms110mas222/E M-algorithm.pdf. [Accessed 23 6 2019].
- [8] R.K. Beatson, W.E. Ong, I.Rychkov, "Faster fast evaluation of thin plate splines in two dimensions," Journal of Computational and Applied Mathematics, vol. 261, pp. 201-212, 2014.
- [9] Chuanfa Chen, Yanyan Li, Changqing Yan, Hongfei Dai, Guolin Liu, "A TPS besed feature preserving method for reducing elevation points derived from LiDAR," Remote Sensing, vol. 7, no. 9, pp. 11344-11371, 2015.
- [10] Wolfgang Keller, Andrzej Borkowski, "Thin plate spline interpolation," Journal of Geodesy, vol. 93, pp. 1251-1269, 2019.
- [11] M.Lohndorf, J.M. Melenk, "On thin plate spline interpolation," ArXiv, 2017
- [12] Penggen Cheng, Zhenyang Hui, Yuanping Xia, et al, "An improved skewness balancing filtering algorithm based on TPS interpolation," Applied Sciences, vol. 9, no. 203, pp. 1-15, 2019.
- [13] Z. Hui et al, A threshold-free filtering algorithm for airborne lidar point clouds based on expectationmaximization, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Beijing, 2018
- [14] Patrick Chazette, Julien Totems, Laurent Hespel, Jean -Sptepane Bailly, Principle and physics of the LiDAR Measurement, Optical Remote Sensing of Land Surfaces, ISTE Press Ltd Published by Elsevier Ltd, 2016.
- [15] Zhe Min, Jiaole Wang, Max Q-H Meng, "Robust generalized point cloud registration with orientational data based on Expectation Maximization," IEEE transactions on automation science and engineering, pp. 99-105, 2019