AUTOMATIC EXTRACTION AND RECOGNITION OF ROAD MARKINGS BASED ON VEHICLE LASER POINT CLOUD

YAO Lianbi[1],*, QIN Changcai[1] , CHEN Qichao[1] , WU Hangbin[1], ZHANG Shaohua[1]

[1] College of Survey and Geo-Informatics Tongji University, Shanghai 200092, China - (lianbi, changcai.qin, mailboxchen, hb, shaohua)@tongji.edu.cn

KEY WORDS: point cloud; point cloud slice; slice projection; inverse distance weighted interpolation; adaptive threshold segmentation; Euclidean clustering;

ABSTRACT:

At present, automatic driving technology has become one of the development direction of the future intelligent transportation system. The high-precision map, which is an important supplement of the on-board sensors under the condition of shielding or the restriction of observation distance, provides a priori information for high-precision positioning and path planning of the automatic driving with the level of L3 and above. The position and semantic information of the road markings, such as the absolute coordinates of the solid line and the broken line, are the basic components of the high-precision map. At present, point cloud data are still one of the most important data source of the high-precision map. So, how to get road markings information from original point clouds automatically deserve study. In this paper, point cloud is sliced by the mileage of the road, then each slice is projected onto respective vertical section. Random Sample Consensus (RANSAC) algorithm is applied to establish road surface buffer area. Finally, moving window filtering is used to extract road surface point cloud from road surface buffer area. On this basis, the road surface point cloud image is transformed into raster image with a certain resolution by using the method of inverse distance weighted interpolation, and the grid image is converted into binary image by using the method of adaptive threshold segmentation based on the integral graph. Then the method of the Euclidean clustering is used to extract the road markings point cloud from the binary image. Characteristic attribute detection is applied to recognize solid line marking from all clusters. Deep learning network framework pointnet++ is applied to recognize remain road markings including guideline, broken line, straight arrow, and right turn arrow.

1. INTRODUCTION

Image processing is applied to get road markings in early studies(Li et al., 2012; Chen et al., 2015; Rebut J et al., 2004; Noda et al., 2011; Foucher et al., 2011). However, the changes of imaging environment such as tree shadow, car shelter, will cause difficulties to the image processing inevitably. Road marking extraction based on laser point cloud depends upon the intensity differences of the intensity of point clouds. Road marking has higher reflection rate compared to the asphalt. Therefor, road marking can be extracted based on intensity differences between road marking and asphalt road surface(Yao et al., 2018; Yang et al., 2012). A. Hervieu, B, et, al. (2015) use ortho-image which is computed by vertical projection of georeferenced point clouds to extract road markings. Guan et al. (2014) extracted road marking based on the point cloud density differences. Kumar et al. (2014) used road surface position to figure out segmentation threshold between road marking and its surrounding road surface. Wang et al. (2015) adopted self-adaption binary segmentation to extract road marking form raster image, and use deep learning to recognize road marking. But this literature only studies pedestrian crossing and arrow marking. Yu et al. (2015) divided point cloud into several parts along the vertical direction of the car driving trace, then Otsu threshold segmentation was applied to extract road marking in each part. Guan et al. (2015) firstly utilized inverse distance weight interpolation to transform point cloud into raster image, then Weighted neighborhood difference histogram was applied to determine intensity segmentation threshold so as to transform the raster image into binary image, finally two dimensional multiscale tensor voting was applied to extracted road marking from binary image.

In this paper, point cloud is sliced by the mileage of the road, then each slice is projected onto its own vertical section. Random Sample Consensus (RANSAC) algorithm is applied to establish road surface buffer area. Finally, moving window filtering is utilized for the sake of extracting road surface point cloud from road surface buffer area. On this basis, the road surface point cloud image is transformed into raster image with a certain resolution by using the method of inverse distance weighted interpolation, and the grid image is converted into binary image though using the method of adaptive threshold segmentation based on the integral graph. Then the method of the Euclidean clustering is used to generate road marking clusters from original road marking point cloud which are restored from binary image. However, these road markings clusters still does not include any semantic information. For the purpose of getting semantic information of these road marking clusters, characteristic attribute detection is applied to recognize solid line marking from all clusters. Deep learning network framework pointnet++(Charles R, et al., 2017) is applied to recognize remain road markings including guideline, broken line, straight arrow, and right turn arrow.

2. METHODOLOGY

In this paper, the process of extraction and recognition of road markings are divided into three steps: road surface point clouds extraction, road marking point clouds extraction and road marking point clouds recognition. The flowchart of the proposed method is shown in figure(1).
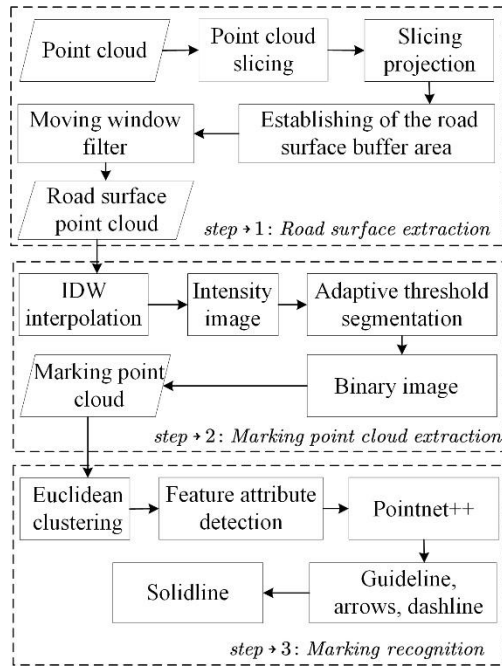
---

* Corresponding author

Figure 1. Flowchart of proposed method

Figure 2. The position of the slicing point

2.1 Point Cloud Slicing

This paper slices point clouds based on the mileage of the road coupled with the position and orientation of the Mobile Mapping System. As the trace is slowly changing along the elevation direction, this paper only considers the plane information of the driving trace. Suppose that the coordinate of the any point on the driving trace is $Tr_i(X_i, Y_i)$, then the mileage value of this point can be calculated by the equation (1).

$$S_i = \sum_{k=1}^{i} \sqrt{(X_k - X_{k-1})^2 + (Y_k - Y_{k-1})^2} \qquad (1)$$

Suppose that the corresponding coordinate azimuth of the point Tr_i equals to α_i, and the length of each slice along the road direction equals to D, then the mileage of the slicing point would be equal to $(n-1)D + \frac{D}{2}$. Suppose that $c_n(X'_n, Y'_n)$ represents the position of the slicing point, then the coordinate of the point c_n can be determined by the equation (2).

$$\begin{cases} X'_n = X_{i_0} + \Delta_n cos(\alpha_{i_0}) \\ Y'_n = Y_{i_0} + \Delta_n sin(\alpha_{i_0}) \end{cases}, n = 1,2,3 \dots \qquad (2)$$

where $\Delta_n = \min\left\{\left|S_i - (n-1)D - \frac{D}{2}\right| \middle| i = 1,2,3 \dots\right\}$

The parameter Δ_n represents the distance between slicing position and the nearest point adjacent with the slicing position on the trace(see figure(2)). The subscript i_0 is determined by the parameter Δ_n.
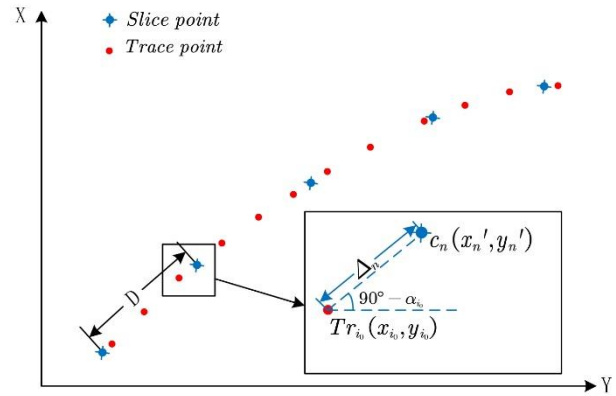
This paper utilizes the weighted sum of the two points which are adjacent to slicing point on the trace to calculate the azimuth of the slice point, which is determined by the equation(3).

$$\alpha'_n = \begin{cases} \frac{\Delta_n \alpha_{i_0}}{S_{i_0+1} - S_{i_0}} + \left(1 - \frac{\Delta_n}{S_{i_0+1} - S_{i_0}}\right)\alpha_{i_0+1}, \Delta \le 0 \\ \frac{\Delta_n \alpha_{i_0}}{S_{i_0} - S_{i_0-1}} + \left(1 - \frac{\Delta_n}{S_{i_0} - S_{i_0-1}}\right)\alpha_{i_0-1}, \Delta > 0 \end{cases} \qquad (3)$$

where $\Delta = S_{i_0} - (n-1)D - \frac{D}{2}$

$c_n(X'_n, Y'_n)$, α'_n, and D determines the position of the slice, the direction of the slice and the length of the slice respectively. For the purpose of slicing the point cloud into several slices, the distance between point cloud and slicing section has to be calculated. When the distance is less than half of length of a slice, the point clouds are bound to lie on the same slice. As is shown in figure(3), The symbol L represents the projection of the slicing section on the XOY plane. The symbol T represents the normal of the slicing section. The symbol P represents the coordinates of some point cloud. The parameter d represents the distance between the slicing section and the point cloud. Then the equation(4) can be determined by relationship between the vector $\overrightarrow{c_n P}$ and the normal vector T. The slicing result of the original point clouds is shown in figure(4) when the length of a slice is assigned to 2m.

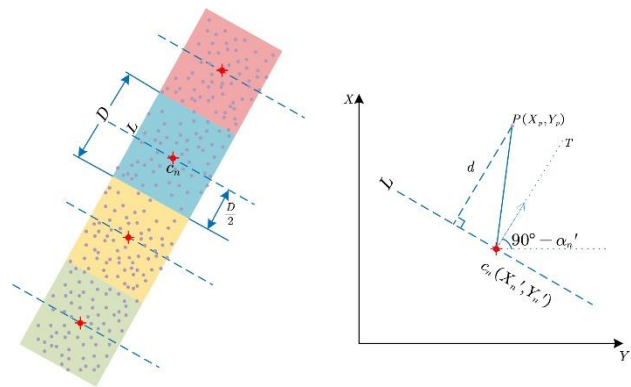$$d = \left|(X_p - X'_n)\cos(\alpha'_n) + (Y_p - Y'_n)\sin(\alpha'_n)\right| \qquad (4)$$

Figure 3. The diagrammatic sketch of slicing point cloud

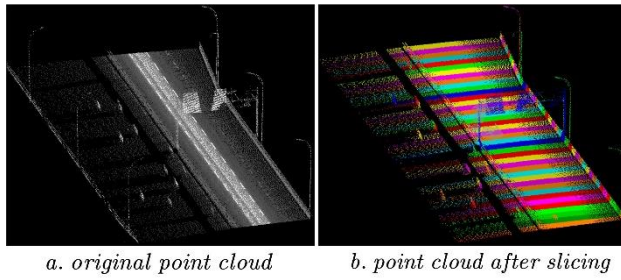*a. original point cloud*   *b. point cloud after slicing*

Figure 4. The comparing between the original point cloud and
the slicing point cloud

2.2 Slicing projection

Considering the trace direction is parallel to the road direction at
most time, this paper used trace section to replace the road section.
Suppose that the point $P_i(X_i, Y_i, Z_i)$ represents point cloud which
lie on one slice, (Xc, Yc) represents the coordinate of the slicing
point, and the parameter α represents the azimuth of the trace
point(see figure(5)), then the coordinate of the point $P_i(X_i, Y_i, Z_i)$
after projection can be determined by the equation(5).

$$
\begin{cases}
Px_i = Xc + ((X_i - Xc)\sin^2\alpha - (Y_i - Yc)\sin\alpha\cos\alpha) \\
Py_i = Yc - ((X_i - Xc)\sin\alpha\cos\alpha - (Y_i - Yc)\cos^2\alpha) \\
Pz_i = Z_i
\end{cases} \quad (5)
$$

For the purpose of convenience of processing point clouds, the
three-dimensional point (Px_i, Py_i, Pz_i) need to be
transformed into two-dimensional point (u_i, v_i). Suppose that
$(\min(Px_i), \min(Py_i))$ represents the origin of the two-
dimensional coordinate system, Then (u_i, v_i) can be
determined by the equation(6). Figure(6) indicates a real scene
of the point cloud projection.

$$
\begin{cases}
u_i = \sqrt{(Px_i - \min(Px_i))^2 + (Py_i - \min(Py_i))^2} \\
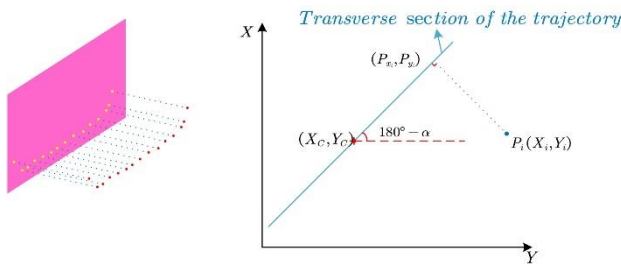v_i = Pz_i
\end{cases} \quad (6)
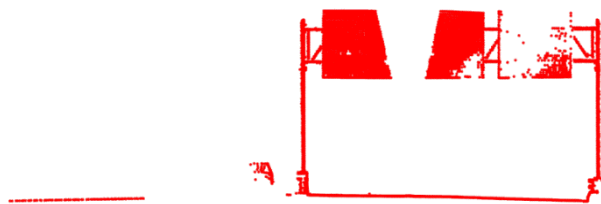$$

Figure 5. Projection of the point cloud

Figure 6. The real scene of the projection of the point cloud

2.3 The Establishing of Road Surface Buffer Area

This paper establishes section buffer area for the purpose of
filtering the noise points such as trees, building and road lamp.
Firstly, pseudo scan points are generated by dividing points on
the projection plane into several small intervals which are
$[0, W), [W, 2W), [2W, 3W) \cdots, [(k-1)W, kW)$ according to
designative interval W. Suppose that (u_i^k, v_i^k) represents the
points which lie in the k-th interval, then $((k-1)W, \min(u_i^k))$
is the coordinates of pseudo scan points generated from the k-th
interval. The RANSAC algorithm is applied to detect straight line
l. Finally, the points which lie in buffer area can be extracted by
establishing buffer area of the straight line. The straight line l is
assigned to be the center of the buffer area. Assign the parameter
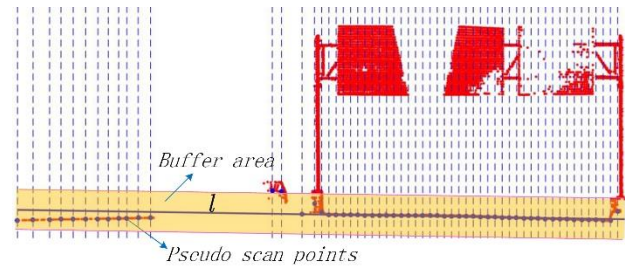W to be 0.05m, then the buffer area of the straight line is shown
in figure(7).

Figure 7. The extraction of the buffer area of the road surface

2.4 Moving Window Filter

This paper uses moving window filter to crop point clouds which
lie inside the buffer area. As is shown in figure(8-a). Firstly, the
point clouds are divided into several intervals according to a
certain interval W_{window}. The minimum bounding rectangle of
the point clouds which lie inside each interval need to be
calculated. Let $Rect_i(h_i, dh_i)$ represent the minimum bounding
rectangle, and h_i、dh_i represent the elevation of the higher edge
and the elevation of the $Rect_i$ respectively. Let
$roadThickness$ represent a certain threshold value, when the
equation(7) is satisfied, $Rect_i$ is marked as the key window(the
window which is marked as red color in figure(8)).

$$
dh_i > roadThickness \quad (7)
$$

Secondly, the key window need to be cropped for the purpose of
extracting road surface point clouds. There are two situations
when the point clouds are cropped. One is that the key window
such as $Rect_1$、$Rect_9$, lies on the edge of one section, as is shown
in figure(8-a). Take $Rect_1$ as an example. Suppose that point set
$Set_1(x_k, y_k)$ are included in $Rect_1$, and $Set_1'(x_j, y_j)$
represents a subset where the points satisfied the equation $y_k >
h_2$, then let the crop boundary to be $Cutx(Cutx = max(x_j))$,
when the points in Set_1 satisfy the equation(8), the points are
recognized to be road surface point clouds, as is shown in
figure(8-b).

$$
x_k > Cutx, y_k \leq h_2 \quad (8)
$$

The other situation is the key window lie in the interior of the road. Take $Rect_5$ as an example. Suppose that the point clouds of $Set_5(x_k, y_k)$ are included in $Rect_5$, then the point clouds which satisfied the equation(9) are recognized as road surface point clouds, as is shown in figure(8-c).

$$y_k \le h_4 + \frac{(h_6 - h_4)(x_k - 4W_{window})}{W_{window}} \qquad (9)$$

Figure(9) shows the result of the road surface point clouds which are extracted by moving window filter when the width of the moving window is set to be 0.4m.
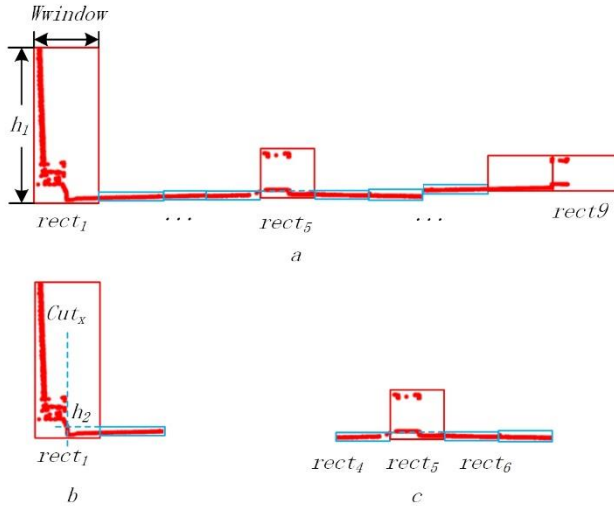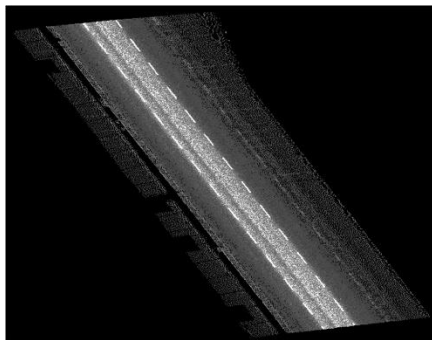

Figure 8. Moving window filter

Figure 9. Road surface point clouds

2.5 Road Surface Point Cloud Rasterization

This paper uses Inverse Distance Weighted(IDW) to transform the three-dimensional point clouds into the two-dimensional raster image. Let $P_i(X_i,Y_i,Z_i,I_i)$ represent the point cloud, where I_i is the intensity of the laser reflection. Let R_{pixel} represent the resolution of the raster image. Let R_{IDW} represent interpolation radius. When the point P_i is projected onto the two-dimensional plane, the corresponding point after projection is (X_i,Y_i,I_i). When the following four equations $minX = min(X_i)$, $maxX = max(X_i)$, $minY = min(Y_i)$, and $maxY = min(Y_i)$ are satisfied, the size of the raster image is to be determined by the equation(10).

$$\begin{cases} Width = \left[\frac{maxX - minX}{R_{pixel}}\right] \\ Height = \left[\frac{maxY - minY}{R_{pixel}}\right] \end{cases} \qquad (10)$$

The point clouds(P_k) which lie interior of interpolation radius are determined by the position of the interpolation point $Pixel_{(m,n)}$ and the interpolation radius(see equation(11) and figure(10)).

$$\{P_k(X_k, Y_k, I_k)|R_{IDW} > d_{(m,n,k)}\} \qquad (11)$$

where

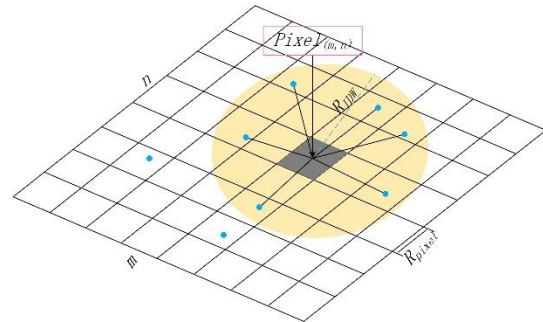$$d_{(m,n,k)} = \sqrt{(X_k - (minX + \frac{(2m-1)R_{pixel}}{2}))^2 + (Y_k - (minY + \frac{(2n-1)R_{pixel}}{2}))^2}$$

Figure 10. Inverse distance weighted interpolation

$d_{(m,n,k)}$ represents the distance between the point P_k and the pixel center $Pixel_{(m,n)}$. Then the intensity value of the interpolation point is determined by the equation(12), where $1/d_{(m,n,k)}$ is assigned to be the weight.

$$I_{(m,n)} = \frac{\sum \frac{I_k}{d_{(m,n,k)}}}{\sum \frac{1}{d_{(m,n,k)}}} \qquad (12)$$

Let $minI = min(I_i)$, $maxI = max(I_i)$, then the pixel value after normalization is determined by the equation(13).

$$G_{(m,n)} = \frac{I_{(m,n)} - minI}{maxI - minI} \qquad (13)$$

Let 0.05m be the pixel resolution and 0.15m as the interpolation radius, figure(8) is transformed into figure(11) using IDW.
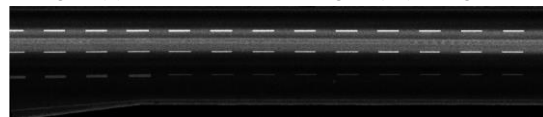

Figure 11. Intensity image

2.6 The Adaptive Threshold Segmentation

The adaptive threshold segmentation based on integral image is proposed by Bradley and Roth (2007). The integral image refers to the image that each pixel value is the sum of top left rectangular area of original image and it is convenient to calculate pixel value sum of arbitrary rectangular area of original image by integral image. The adaptive threshold of each pixel is determined by the average of its surrounding pixels' grey value. Gaussian filter is applied to process intensity image, and let the original pixel be $p_i(x_i, y_i)$, corresponding grey value be $I_{(x_i,y_i)}$

and integral image be $intI_{(x_i,y_i)}$. We define two parameters s being background radius and t being segmentation coefficient. Then the sum of the grey value of pixels which lie in a rectangular area, the centre of which is p_i and the length of which is $2s + 1$ (see Figure 12), is calculated as follows:

$$sum_i = intI_{(x_i+s,y_i+s)} + intI_{(x_i-s-1,y_i-s-1)} - intI_{(x_i-s-1,y_i+s)} - intI_{(x_i+s,y_i-s-1)} \quad (14)$$

The image boundary shall prevail when the rectangular area is beyond it. The binary image is as follows:

$$I'_{(x_i,y_i)} = \begin{cases} 1, & I_{(x_i,y_i)} > \frac{sum_i * t}{(2s+1)^2} \\ 0, & others \end{cases} \quad (15)$$
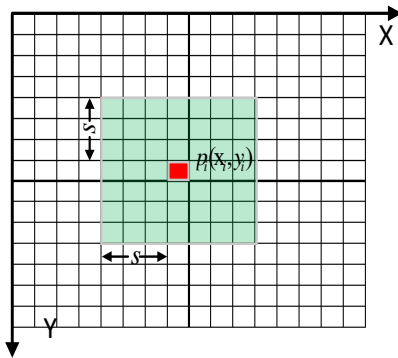

Figure 12. Adaptive threshold segmentation

We selected four segments of 60 m data samples and generated intensity image with the parameter R_{IDW} being 0.15 m and the parameter R_{pixel} being 0.05 m (see Figure 13). Suppose that *TP*(true positive), *FN*(false negative), and *FP*(false positive) represent the counts of the pixels which are correctly, defectively, redundantly extracted respectively in the process of automatic extraction of road markings. The manual annotation result is assessment reference(Yao et al., 2018) and *Fscore*(Cheng et al., 2017) is calculated by using the method proposed by Cheng. The intensity image is processed with adaptive threshold method by different parameters of *s* and *t* according the equation(16~18).

$$completeness = \frac{TP}{TP+FN} \quad (16)$$

$$correctness = \frac{TP}{TP+FP} \quad (17)$$

$$Fscore = \frac{2*completeness*correctness}{completeness+correctness} \quad (18)$$

Figure 14 is the extracted marking of samples point cloud data with their own max *Fscore*. Table 1 is the evaluation of extracted marking of samples.
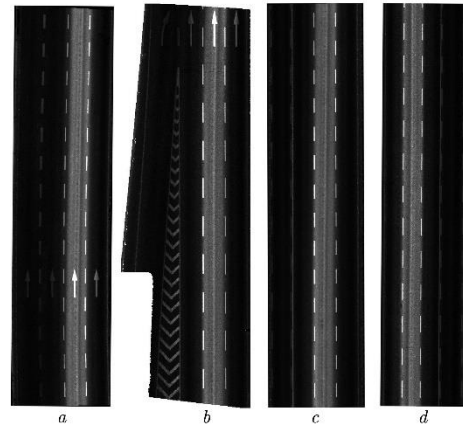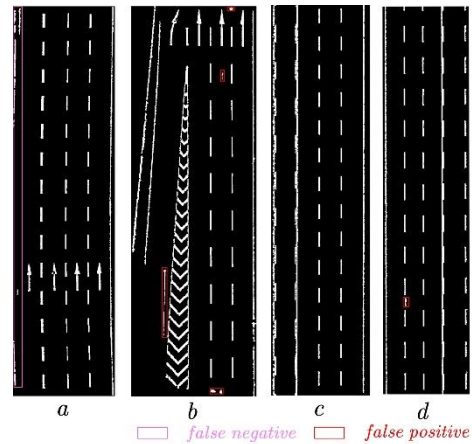

Figure 13. Raster image

Figure 14. Adaptive threshold segmentation

area	(s,t)	completeness	correctness	F_score
a	(11, 1.27)	0.83	0.89	0.86
b	(12, 1.29)	0.89	0.87	0.88
c	(11, 1.27)	0.95	0.81	0.87
d	(10, 1.27)	0.79	0.87	0.83

Table 1. Evaluation of extracted marking of samples

2.7 Point Cloud Euclidean Clustering

Road surface point clouds can be restored from the binary image according to the foreground pixels. Point cloud clustering means that the points which belong to the same category(though the category may be unknown after clustering) are clustered into one cluster according to a certain principle. This paper use Euclidean clustering to divide the point clouds into different clusters based on the different Euclidean distances among point clouds. Let $P = \{P_i(X_i, Y_i) | i=1, 2, 3\cdots\}$ represent point set which are to be clustered, parameter D represent the cluster radius. The concrete steps of Euclidean clustering is as follows:

(1). Take any point p from the point set P, and put the point p into the point set *cluster*.

(2). Let C represent the circle, the center of which is the point p, and the radius of which is D. Put the points which lie inside the circle C into the *cluster* defined by the step one(see figure 15-a).

(3). The former step is repeated by centering on point p which comes from the *cluster* point by point until there are no new

points which are added into the *cluster*(see figure 15-b and figure 15-c).

(4). Start a new *cluster* by repeating from step one to step three using remain points which come from the point set *P*(see figure 15-d).

When the parameter D is assigned to be 0.5m, the clustering result of the point clouds of figure(14) is shown in figure(16).
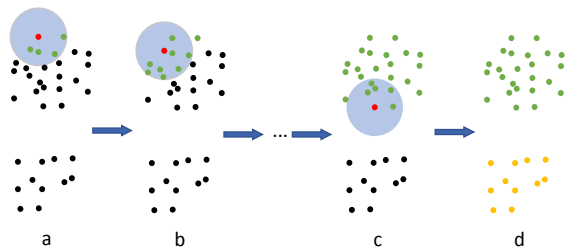

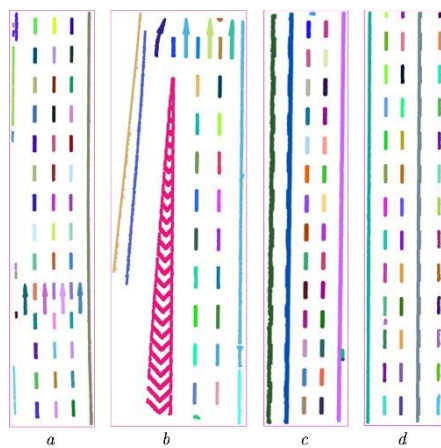Figure 15. The process of point cloud Euclidean clustering

Figure 16. The result of Euclidean clustering

2.8 Road Marking Semantic Recognition

Separate cluster is generated after the discrete point clouds are clustered into point cloud cluster. The minimum bounding rectangle(MBR) of each cluster need to be calculated in order to get some attribute characteristics of each cluster. Solid line can be detected from all the clusters according to one or more attribute characteristics of the MBR of each cluster. Other road markings such as broken line, guideline, straight arrow, and right turn arrow can be recognized by using the deep learning network framework pointnet++.

2.8.1 Solid line Recognition: Suppose that a represents the longer side of the MBR, and b represents the shorter side of the MBR. In order to reveal geometric characteristics of the marking cluster, some attribute characteristics of the MBR of each separate cluster are defined as follows:

Length-width ratio(β). The length-width ratio reflects how similar the MBR is to the square. The parameter β is determined by the equation(19).

$$\beta = \frac{a}{b} \qquad (19)$$

When the parameter a and the parameter β is assigned a certain threshold value($a > 7\text{m}, \beta > 20$), the solidline can be

detected from all the extracted clusters(see figure(17)).

2.8.2 Recognition of Other Road Markings: The solid line can be discarded after it was recognized. The remain clusters here included guideline, straight arrow and right turn arrow. In early studies, Most of deep learning network framework in the realm of point clouds like pointnet++, pointcnn(Li et al., 2018), KPconv(Thomas et al., 2019), et, al., were applied to indoor point clouds. There were also some network framework which were used in outdoor point clouds for the purpose of semantic recognition of the outdoor objects such as trees, buildings, and road lamp. But there were seldom network framework which were applied to semantic recognition of the road markings. In this paper, The deep learning network pointnet++ was used to recognize remain clusters. The result of recognition was shown in figure(17). Different kinds of road markings were marked as different color. For example, all the broken lines which were correctly recognized were marked as blue color. There was one broken line which was erroneously recognized to be straight arrow. We can see that all the straight arrow and one right turn arrow was correctly recognized. One guideline was correctly recognized. From all separate broken lines, only one broken line was erroneously recognized. Some broken lines which lie very close to the solid line were erroneously recognized to be solid line. The reason was that when the method of Euclidean clustering was applied to discrete point clouds, these broken lines which lay very close to the solid line were clustered together with solid line. For the purpose of evaluating the proposed method in this paper quantitatively, 5km of the road were selected. The detected length of the solid line was 3.6km, So the detection rate was 0.72. The evaluation of other kinds of road markings was listed in table2.
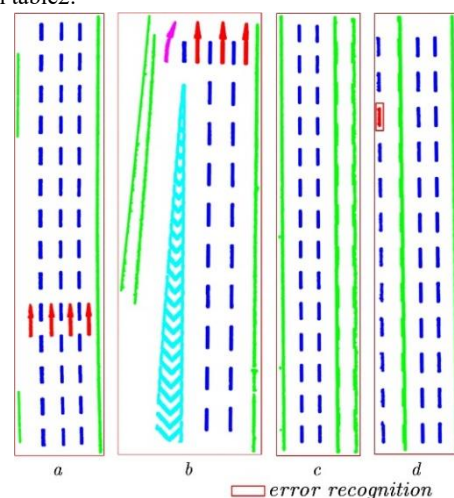

Figure 17. Semantic recognition result of each cluster

category	completeness	correctness	F_score
Broken line	0.992	0.995	0.994
Straight arrow	0.909	0.967	0.937
Left arrow	1.00	1.00	1.00
Right arrow	1.00	1.00	1.00
Guideline	1.00	1.00	1.00

Table 2. Evaluation of marking recognition

3. CONCLUSIONS AND DISCUSSIONS

This paper extracted road surface point clouds through slicing original point clouds into separate slice along the road direction.

IDW was used to transform road surface point clouds into road raster image with three-dimensional coordinate and the intensity of the point clouds. Then Adaptive threshold segmentation was applied into the raster image for the purpose of transforming raster image into binary image. After we get the binary image, Euclidean clustering was utilized to cluster discrete point cloud into separate point cloud cluster. In order to get geometric feature of the separate point cloud cluster, the MBR of each point cloud cluster was calculated. Finally, the solid line was detected by restricting the length of the longer side and the length-width ratio of the MBR of each point cloud cluster. The remain road markings including the broken line, the straight arrow, the right turn arrow and the guideline were recognized through the deep learning network framework pointnet++. The result indicates that all the solid line were detected correctly. All the broken line except one and those which lay very close to the solid line were recognized correctly, All the arrows and the guideline was recognized correctly. But not all kinds of road markings have been studied in this paper. Future work will be focused on other road markings such as pedestrian crossing and crosswalk notice sign.

ACKNOWLEDGEMENTS

REFERENCES

A. Hervieu, B. Soheilian, M. Brédif, 2015. Road marking extraction using a Model&Data-driven RJ-MCMC. ISPRS Annals of Photogrammetry Remote Sensing and Spatial Information Sciences, II-3/W4, pp.47-54, PIA 2015, Munich, Germany, March 2015.

Bradley D, Roth G, 2007. Adaptive thresholding using the integral image[J]. Journal of graphics tools, 12(2): 13-21.

Charles R. Qi, Li Yi, Hao Su, Leonidas J. Guibas, 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA.

Chen T, Chen Z, Shi Q, Xinming Huang, 2015. Road marking detection and classification using machine learning algorithms[C]. Intelligent Vehicles Symposium (IV). IEEE, 2015: 617-621.

Cheng M, Zhang H, Wang C, Jonathan Li, 2017. Extraction and Classification of Road Markings Using Mobile Laser Scanning Point Clouds[J]. IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing, 10(3):1182-1196.

Fischler, M.A. and Bolles, R.C, 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM , 24(6): 381–395.

Guan H, Li J, Yu Y, Ji Z, Wang C, 2015. Using mobile LiDAR data for rapidly updating road markings[J]. IEEE Transactions on Intelligent Transportation Systems, 16(5): 2457-2466.

Haiyan Guan, Jonathan Li, Yongtao Yu, Cheng Wang, Michael Chapman, Bisheng Yang, 2014. Using mobile laser scanning data for automated extraction of road markings[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 87: 93-107.R

Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, B. Marcotegui, François Goulette, Leonidas J. Guibas, 2019. KPConv: Flexible and Deformable Convolution for Point Clouds[C]. IEEE International Conference on Computer Vision, arXiv:1904.08889.

Kumar P, McElhinney C P, Lewis P, Timothy McCarthy, 2014. Automated road markings extraction from mobile laser scanning data[J]. International Journal of Applied Earth Observation and Geoinformation, 32: 125-137.

Li Z, Cai Z, Xie J, Ren X, 2012. Road markings extraction based on threshold segmentation[C]. Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on. IEEE: 1924-1928.

Masafumi Noda, Tomokazu Takahashi, Daisuke Deguchi, Ichiro Ide, H. Murase, Yoshiko Kojima, Takashi Naito, 2011. Detection of Road Markings Recorded in In-Vehicle Camera Images by Using Position-Dependent Classifiers[J]. IEEJ Transactions on Industry Applications, 131(4), 466-474.

Philippe Foucher, Yazid SEBSADJI, Jean-Philippe Tarel, Philippe Nicolle, 2011. Detection and recognition of urban road markings using images[C]. International IEEE Conference on Intelligent Transportation Systems. 14: 1747-1752.

Rebut J, Bensrhair A, Toulminet G, 2004. Image segmentation and pattern recognition for road marking analysis[C]. Industrial Electronics, 2004 IEEE International Symposium on. IEEE, 1: 727-732.

Wang, H., Luo, H., Wen, C., Cheng, J., Li, P., Chen, Y., Wang, C., Li, J., 2015. Road boundaries detection based on local normal saliency from mobile laser scanning data[J]. IEEE Geosci. Rem. Sens. Lett. 12, 2085–2089.

Wellner P D, 1993. Adaptive Thresholding for the DigitalDesk[J]. Xerox.

Yang B, Fang L, Li Q, Jonathan Li, 2012. Automated extraction of road markings from mobile LiDAR point clouds[J]. Photogrammetric Engineering & Remote Sensing, 78(4): 331-338.

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, Baoquan Chen, 2018. Pointcnn: Convolution on x-transformed points[C]. In Advances in Neural Information Processing Systems, 820–830.

Yao, L., Chen, Q., Qin, C., Wu, H., and Zhang, S., 2018: Automatic Extraction of Road Markings from Mobile Laser-Point Cloud Using Intensity Data, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLII-3, 2113–2119, https://doi.org/10.5194/isprs-archives-XLII-3-2113-2018, 2018.

Yu Y, Li J, Guan H, Fukai Jia, Cheng Wang, 2015. Learning hierarchical features for automated extraction of road markings from 3-D mobile LiDAR point clouds[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8(2): 709-726.