# AUTOMATIC DRAWING FOR TRAFFIC MARKING WITH MMS LIDAR INTENSITY

G. Takahashi [a] , H. Takeda [a], Y. Shimano [a]

[a] Spatial Information Division, Kokusai Kogyo Co., Ltd., Tokyo, Japan - (genki_takahashi, hiroshi1_takeda, yuichi_shimano)@kk-grp.jp

**Commission V, WG V/5**

**KEY WORDS:** MMS, LIDAR, Intensity, Traffic marking, Mapping, Hough transform, Median filter

**ABSTRACT:**

Upgrading the database of CYBER JAPAN has been strategically promoted because the “Basic Act on Promotion of Utilization of Geographical Information,” was enacted in May 2007. In particular, there is a high demand for road information that comprises a framework in this database. Therefore, road inventory mapping work has to be accurate and eliminate variation caused by individual human operators. Further, the large number of traffic markings that are periodically maintained and possibly changed require an efficient method for updating spatial data. Currently, we apply manual photogrammetry drawing for mapping traffic markings. However, this method is not sufficiently efficient in terms of the required productivity, and data variation can arise from individual operators. In contrast, Mobile Mapping Systems (MMS) and high-density Laser Imaging Detection and Ranging (LIDAR) scanners are rapidly gaining popularity. The aim in this study is to build an efficient method for automatically drawing traffic markings using MMS LIDAR data. The key idea in this method is extracting lines using a Hough transform strategically focused on changes in local reflection intensity along scan lines. However, also note that this method processes every traffic marking. In this paper, we discuss a highly accurate and non-human-operator-dependent method that applies the following steps: (1) Binarizing LIDAR points by intensity and extracting higher intensity points; (2) Generating a Triangulated Irregular Network (TIN) from higher intensity points; (3) Deleting arcs by length and generating outline polygons on the TIN; (4) Generating buffers from the outline polygons; (5) Extracting points from the buffers using the original LIDAR points; (6) Extracting local-intensity-changing points along scan lines using the extracted points; (7) Extracting lines from intensity-changing points through a Hough transform; and (8) Connecting lines to generate automated traffic marking mapping data.

## 1. INTRODUCTION

Recently, it has become increasingly important to use Mobile Mapping Systems (MMS) sensor data in various applications. We can extract information from around the roads using MMS LIDAR data. Therefore, in many cases, we have recently begun drawing traffic markings using MMS LIDAR data (Bisheng, 2012). Automated methods aid in minimizing costs and obtaining highly accurate results. Manual processing includes human errors, subjectivity errors, and low reproducibility. However, due to the lack of technology, we occasionally cannot use automatic methods in drawing traffic markings.

Based on these conditions, we set our ultimate goal as automatic drawing of all traffic markings using MMS LIDAR data. In such drawing work, it is necessary to draw the shape of traffic markings accurately and with the fewest vertices. Thus, one of the phases of our study is to try to extract traffic markings that have straight lines. Images of traffic markings are shown in Figure 1; most traffic markings consist of combinations of straight lines.

In this study, we use MX-8, which is a system developed by Trimble (Trimble websites, 2012). The exterior of MX-8 is shown in Figure 2. An example of LIDAR data collected by MX-8 is shown in Figure 3. MX-8 has two VQ250 LIDAR scanners developed by RIEGL that can acquire X, Y, and Z data, along with intensity information (RIEGL websites, 2012). The specifications of the LIDAR scanner are listed in Table 1.

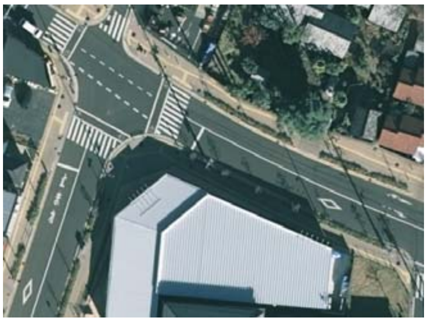


Figure 1. Traffic marking in Japan

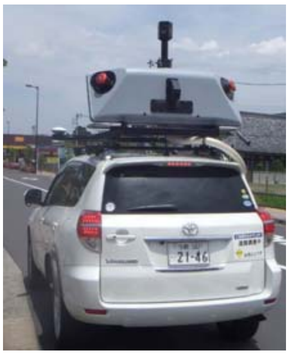


Figure 2. MX-8 exterior

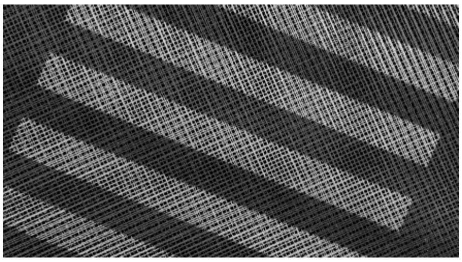


Figure 3. Example of LIDAR data by MX-8 (Crosswalk)

| Model number | VQ250 |
|---|---|
| Maker | RIEGL |
| Scan angle | 360 degrees |
| Max LIDAR pulse | 300 kHz |
| Scanning rate | 100 Hz |
| Distance measurement range | 300 m (Max) |
| Measurable distance | 75 m(10% Intensity)<br>200 m(80% Intensity) |
| System error | 10 mm(150 m Distance) |
| Angular resolution capability | 0.001 degrees |
| Distance between two points (5 m point) | 7 ~ 15 cm(traveling direction )<br>2 cm(right angle direction) |

Table 1. Specification of the LIDAR scanner

## 2. LIMITATIONS WITH THE CONVENTIONAL METHODS

### 2.1 Problems with manual drawing

Presently, we apply only manual photogrammetry drawing for mapping traffic markings. In transitioning from this, we have two matters to consider.

The first is faithfully extracting and accurately drawing traffic markings. The *Road Traffic Act* defines the size of traffic markings in Japan (*Road Traffic Act* websites, 2009). Therefore, if traffic markings conform to the act, it should be easy to extract them through template matching. In many cases, however, the real size of a traffic marking does not overlap the defining template. Thus, the template matching method is not suitable for road inventory map work because it requires high accuracy and reflection of the state in practice, accounting for number of vertices and possible error in the road marking on the part of the road administrator.

The second is the need to draw traffic markings with the fewest number of vertices. For example, when we draw the traffic marking shown in Figure 3, we should only use four vertices, and when we draw the traffic marking in Figure 4, we should only use seven vertices. When an operator draws traffic markings using MMS LIDAR intensity data, the intensity threshold value is set subjectively. This subjectivity brings the possibility of variation in quality when using manual drawing.

### 2.2 Problems with camera images

Table 2 compares various approaches of extracting traffic markings. With approach [2], camera image, we introduce a significant cost owing to the need of creating orthoimages, and therefore, this approach is not suited to the aim of cost reduction. Furthermore, the existence of quality issues such as overexposure, underexposure, and halation are significantly dependent on weather conditions.

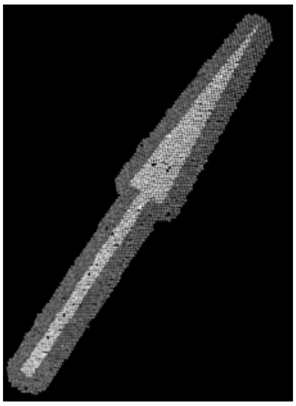


Figure 4. Traffic marking with seven vertices

| | [1] LIDAR with intensity | [2] Camera image | [3]LIDAR with RGB from camera image |
|---|---|---|---|
| [A]pre-cost | Good | Poor | Poor |
| [B]night | Good | Poor | Poor |
| [C]density | Poor | Good | Poor |
| [D]including error | Good | Poor | Poor |
| [E]accuracy (C+D) | Good | Good | Poor |

Table 2. Comparison table of sensor

With approach [3], LIDAR data with RGB from a camera image, we introduce two sources of error: the LIDAR scanner’s misalignment and the camera image’s misalignment. These two errors affect to the accuracy. Therefore, Approach [1] is a much better option for automation as a result.

Finally, with approach [1], LIDAR with intensity, we do not have the additional sources of error or the costly pre-processing incurred by the other two methods, and LIDAR scanners can obtain data even when there is no sun. However, LIDAR points are sensitive to noise due to the greater discreteness as compared to a camera image. Therefore, we have to eliminate this noise before processing.

From the above comparison, approach [1] is more advantageous in many ways than the other approaches. However, MMS LIDAR data has a problem from its disposition, as we explain in the next section.

### 2.3 Problems with MMS LIDAR data

In general, MMS LIDAR data includes noise as shown in Figure 5. These noise points have high intensity values within traffic markings, and so should be removed. Additionally, the intensity distribution of MMS LIDAR data locally changes due to fading, dirt, and differences in incidence angles. Therefore, we should consider locality about these possibility.

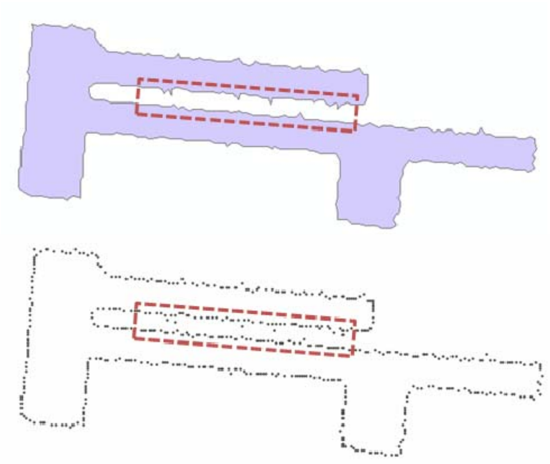


Figure 5. Noise points obtained using the conventional method

## 3. METHODOLOGY

### 3.1 Flow chart

This study's flow chart is shown in Figure 6.

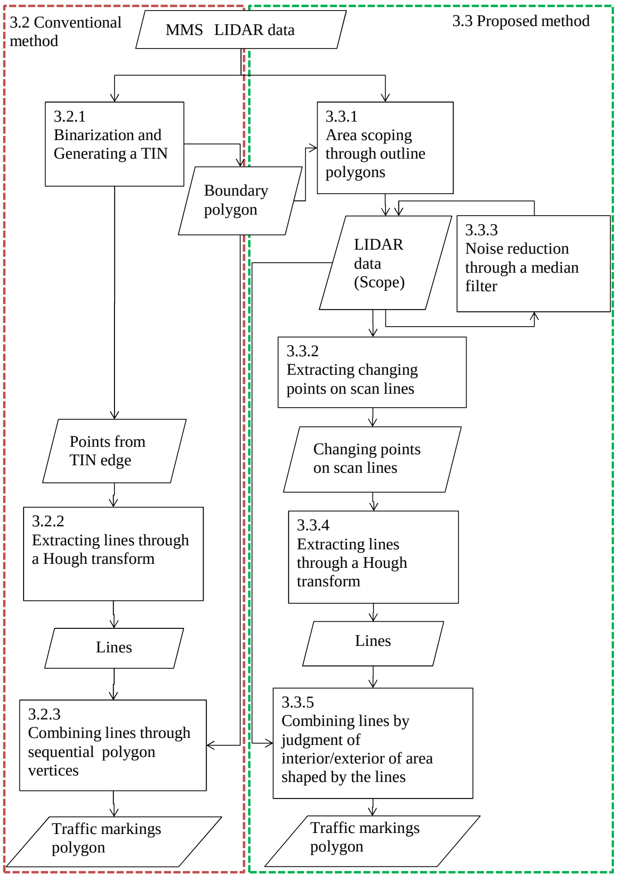

3.2 Conventional method
MMS LIDAR data
3.3 Proposed method
3.2.1 Binarization and Generating a TIN
Boundary polygon
3.3.1 Area scoping through outline polygons
LIDAR data (Scope)
3.3.3 Noise reduction through a median filter
3.3.2 Extracting changing points on scan lines
Changing points on scan lines
Points from TIN edge
3.2.2 Extracting lines through a Hough transform
3.3.4 Extracting lines through a Hough transform
Lines
Lines
3.2.3 Combining lines through sequential polygon vertices
3.3.5 Combining lines by judgment of interior/exterior of area shaped by the lines
Traffic markings polygon
Traffic markings polygon

Figure 6. Flow chart

Our previous method, "binarization method," is shown on the left side of Figure 6. This method consists of binarizing LIDAR points according to intensity and extracted points, generating a TIN from the higher intensity points, deleting arcs by length, generating outline polygons on the TIN, generating buffers from the outline polygon, and extracting lines from TIN nodes by a Hough transform. Finally, we extract traffic markings from the lines by continuously the outline polygon. These details are explained in 3.2.

The improved method, "local change detection," is shown on the right side of Figure 6. This method involves extracting points by outline buffers made through the binarization method from the original LIDAR points, extracting local-intensity-changing points along scan lines from the extracted points, and extracting lines from the intensity-changing points through a Hough transform. Finally, we connect the lines to generate traffic marking automated mapping data. Details are explained in 3.3.

### 3.2 Conventional method

**3.2.1 Binarization and Generating a TIN**: First, because there is a high intensity value within traffic markings, we try to extract a traffic marking polygon through LIDAR intensity binarization. The intensity value distribution around the traffic marking is shown in Figure 7. We consider extracting the traffic marking area through setting the appropriate intensity threshold because there is a clear difference between the inside and outside. We use the same intensity threshold in all areas.

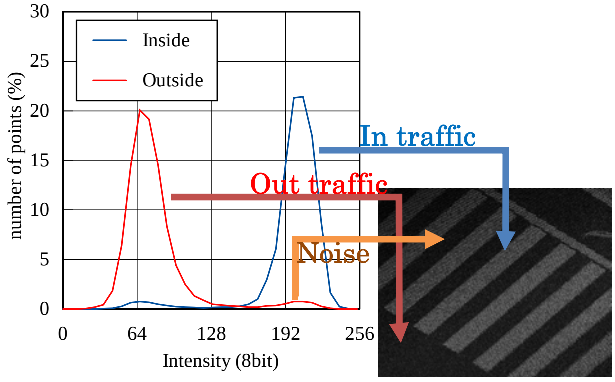


Figure 7. Intensity distribution around traffic marking

Next, we create a TIN from the extracted points and divide into domains by TIN edge line lengths; these lengths are the same value in all areas. We create a polygon from the outline of each TIN, as shown in Figure 8, and we use this polygon in another method shown later.

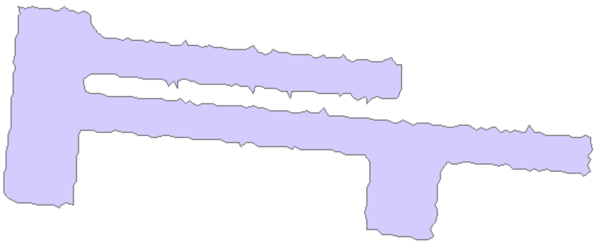


Figure 8. Polygon from TIN outline

**3.2.2 Extracting lines through a Hough transform**: We can regard the polygon shape as the edge of the traffic marking, and so we extract approximate lines of the traffic marking from the points of the polygon vertices using a Hough transform, as shown in Figure 9.

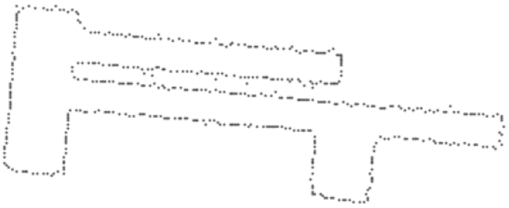


Figure 9. Points from polygon vertices

We wish to convert the LIDAR data into a traffic marking polygon with the fewest vertices. In general, a Hough transform (Chris, 2004) (Davies, 1988) (Paul, 1959) (Rabbani, 2005) (Richard, 1972) is used for extracting lines from a camera image. In this study, we expand the Hough transform to finding polygon vertices from LIDAR data and trying to extract the approximate lines around a traffic marking. We use a Hough transform for object detection in our image through the steps as follows.

Step 1) Vote to cell in rho-theta plane

Step 2) Extract approximate line from the cell with maximum value

Step 3) Make a buffer polygon from the line and eliminate points within the polygon

Step 4) Repeat until the number of remaining points is under the threshold

The extraction result is shown in Figure 10.

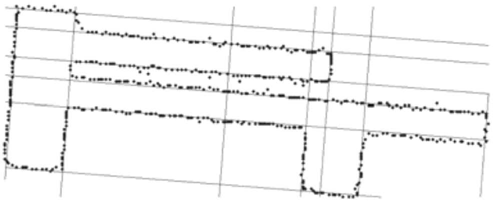


Figure 10. Lineaments by Hough transform

**3.2.3 Combining lines through sequential polygon vertices**: With this method, we draw traffic marking polygons by combining lines. Convex polygons like those shown in Figure 3 are easy to make in this way by connecting the intersections of lines. On the other hand, lines from concave polygons like those shown in Figure 4 have two or more non-consecutive vertices, and so the polygon is not shaped by ordered vertices; this problem also occurs in convex polygons with additional approximate lines. Because of this, we have to consider lines and their connectivity. Therefore, we combine lines by vertex grouping, as vertices have sequential IDs along the polygon outline.

We describe this method as follows with Figure 11; a simple illustration of combining lines. Points are outline vertices and lines 1, 2, and 3 are extracted lines. Line 1 has points 1, 2, 3, and 9, line 2 has points 4 and 5, and line 3 has points 6, 7, 8 and 9. Point 1, 2 and 3 add to group 1A, point 9 adds to group 1B, point 4 and 5 add to group 2A, and points 6, 7, 8 and 9 add to group 3A, in consideration of continuity of the point IDs. Point 3 from group 1A and point 4 from group 2A are close in ID, so we can make an intersection point between line 1 and line 2. In the same way, point 5 from group 2A and point 6 from group 3A are close in ID, and so we can make an intersection point between line 2 and line 3. On the other hand, though point9 from group 1B and point 9 from group 3A are close in ID, line 1 and line 3 have no intersection point because group 3A also has other points. We continue until all groups consist only of discontinuous points, at which point we complete the shape. This result is shown in Figure 12.

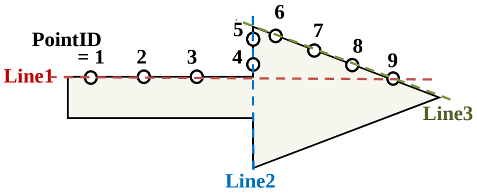


Figure 11. Illustration of line combining

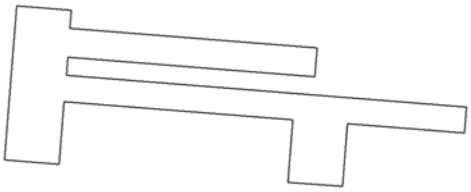


Figure 12. Extracted traffic marking

### 3.3 Proposed method

The conventional method described above is better than our previous method at extracting all traffic markings in all areas. However, this method has a robustness problem because it uses the same threshold for both TIN arc length and intensity. To solve this problem, we try to extract local-intensity-changing points along scan lines. Through this, we can expect to extract traffic marking more robustly, as this method does not require a single global threshold value.

**3.3.1 Area scoping through outline polygons**: The result of extracting lines in a large area is shown in Figure 13. Each line is affected by other traffic markings, and so we cannot extract proper lines.

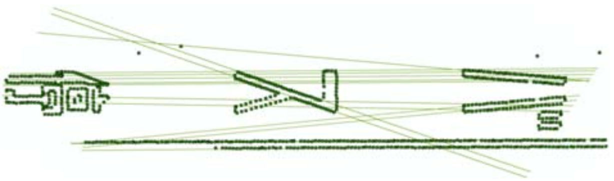


Figure 13. Extracting lines with large area

The binarization method can extract each traffic marking from all areas because it has an extracting outline step. Therefore, we perform an area scoping process that includes extracting outline polygons through binarization and clipping out the polygon from the original LIDAR points. If we include this process in our method before extracting local-intensity-changing points, we can extract lines for every traffic marking.

**3.3.2 Extracting changing points on scan lines**: Next, we try to extract traffic markings through local change detection, as the LIDAR data is sequential. Details of this method are as follows.

1) We set a changing point where the difference between the target point's intensity and the previous point's intensity is more than some threshold.
2) The coordinates of the changing point are between the target point and the previous point.
3) If the target point is far from the previous point, we do not find a changing point, because it is on the outline of the buffer polygon.

**3.3.3 Noise reduction through a median filter**: We cannot use these points without filtering because of noise. Before we can apply a Hough transform, we have to first remove these noise points with the proper method. In this process, we use a median filter for this purpose. A median filter for removing noise in image processing sets each pixel value to the median in a two-dimensional neighborhood mesh (e.g., a 3 by 3 mesh). This allows us to remove spike noise, such as in LIDAR point data. We apply a median filter one-dimensionally for local changing points and a few points around them along the scanning lines.

**3.3.4 Extracting lines through a Hough transform**: We extract lines from changing points through the same method described in 3.2.2.

**3.3.5 Combining lines by judgment of interior/exterior of area shaped by the lines**: This step includes creating areas from extracted lines, clipping LIDAR data from the area, detecting the interior/exterior of traffic markings by the average of each clipping point's intensity value, and forming the traffic marking by dissolving inside areas.

## 4. EXPERIMENT

Input LIDAR data is as follows.

1) Single sensor
2) Single course
3) People and cars are removed from data

### 4.1 Comparison of binarization method and local change detection method

We try to extract various traffic markings by the previous method, binarization, and the new method, local change detection, and compare the results. Target objects are shown in Table 3.

| Markings | Number of vertices | Complicated |
|---|---|---|
| Single arrow | 7 | simple, convex |
| Stop | 16 | complicate, concave |
| Crosswalk (slanting) | 4 | simple, convex |
| Crosswalk notification | 8 | normal, concave |
| Bus stop | 10 | normal, concave |

Table 3. Target objects information

### 4.2 Comparison of filter size

We use a median filter to remove noise. It is important to properly set the filter size for obtaining high accuracy. We tested automatic drawing with filtering using a median size of 1, 3, 5, 7, and 9, and compared the results.

### 4.3 Quantitative comparison

**4.3.1 Comparison of manual and automatic drawing**: We expect a stable and objective drawing from this automated extraction. Therefore, we quantitatively compare manual and automatic drawing through drawing a traffic marking with each method as follows and estimating the standard deviation of these areas.

1) Automatic extraction with random noise
2) Manual extraction repetition

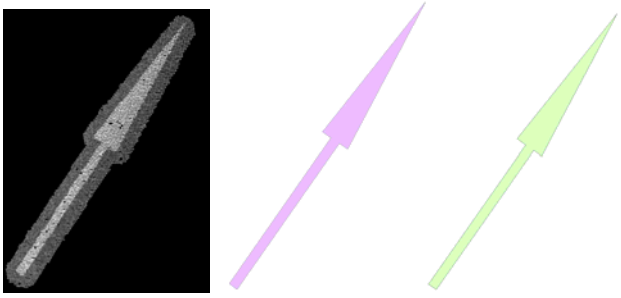
**4.3.2 Comparison of binarization method and local change detection method**: We quantitatively compare the previous and new methods by estimating the area size of some traffic markings, assuming that in this comparison the manual drawing is correct.

## 5. RESULTS AND CONSIDERATION

### 5.1 Comparison of binarization method and local change detection method

We show the resulting extracted traffic markings which shown in 4.1 by previous method; binarization and new one; local change detection as follows.

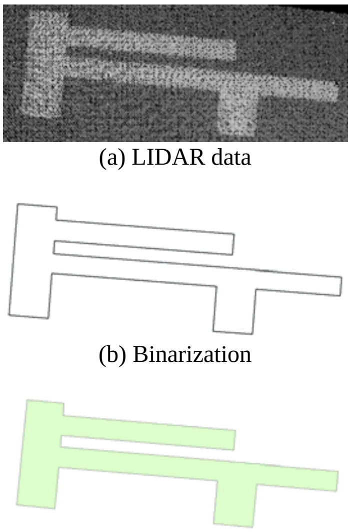
**5.1.1 Single arrow**: This traffic marking shows a direction for cars. Extracting result is shown in Figure 14. (a) is the LIDAR data with intensity, (b) is the polygon produced by the binarization method, and (c) is the polygon produced by the local change detection method. Both methods can exactly extract this traffic marking.



(a) LIDAR data (b) Binarization (c) Local changing

Figure 14. Single arrow

**5.1.2 Stop**: This traffic marking is built from Chinese characters that means "stop". The extraction result is shown in Figure 15. Both methods can again exactly extract the traffic marking.



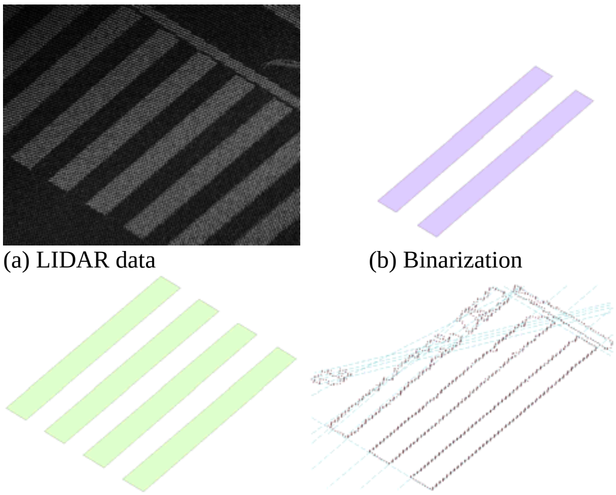
(a) LIDAR data

(b) Binarization

(c) Local changing

Figure 15. Stop

**5.1.3 Crosswalk (slanting)**: The extraction result is shown in Figure 16. We tested our methods on a slanting crosswalk with four regions as shown in (a). The change detection method can extract four regions as shown in (c). On the other hand, binarization method can extract only two regions as shown in (b), as there is sparse LIDAR data in the upper-left side of (a), so as shown in (d) we cannot extract enough points and lines; the binarization method cannot extract an outline polygon in the sparse LIDAR data area because it uses a single threshold for TIN arc length.

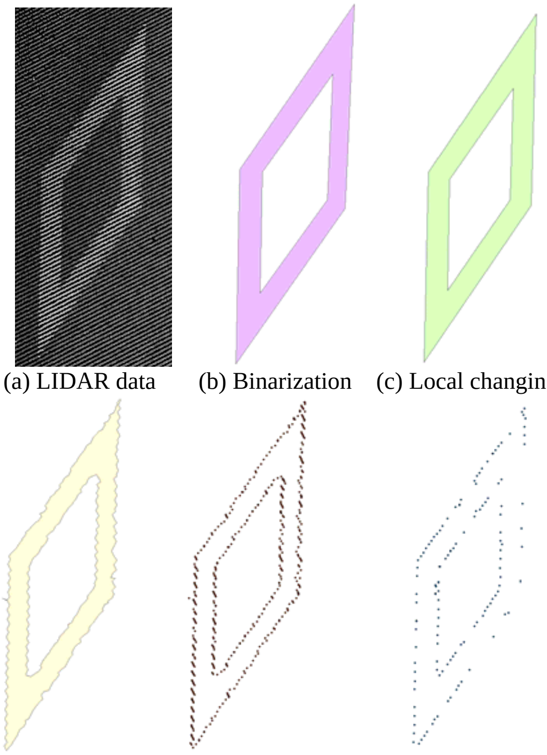


(a) LIDAR data (b) Binarization

(c) Local changing (d) Points and lines by binarization

Figure 16. Crosswalk (slanting)

**5.1.4 Crosswalk notification**:
This traffic marking means "crosswalk ahead". The extraction result is shown in Figure 17. Both methods can exactly extract the traffic marking, but we can see that the inner outline and edge points as shown in (d) and (e), include an acute angle which is rounded and less accurate with the conventional method. However, the local changing points are not rounded by our improved method.

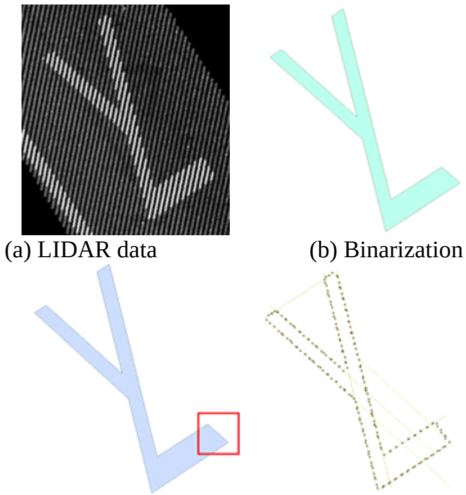


(a) LIDAR data (b) Binarization (c) Local changing

(d) TIN outline (Binarization) (e) TIN edge points (Binarization) (f) Local changing points (Local changing)

Figure 17. Crosswalk notification

**5.1.5 Bus stop**: This traffic marking means "bus stop". The extraction result is shown in Figure 18. As shown in (b) and (c), both methods have failed when compared to the LIDAR data shown in (a). Both methods cannot draw short side properly; in both, it is not vertical to the front and rear lines. Looking at the points and lines of local changing, as shown in (d), it can be seen that the vertical line is not be extracted properly. This indicates that we have a problem with extraction of short sides. This problem will be solved by properly splitting the interval between vertices.



(a) LIDAR data (b) Binarization

(c) Local changing (d) Points and lines (Local changing)

Figure 18. Bus stop

**5.1.6 Consideration**: We have attempted to extract traffic marking through two methods: binarization and local change detection. The results are shown in Table 4. With both methods, we can extract "single arrow" and "stop". However, the binarization method cannot extract traffic markings around low-density areas such as "crosswalk" because it uses the same threshold in all areas. Further, both methods failed to extract "bus stop" due to a problem extracting short sides.

| Markings | Binarization | Local change detection |
|---|---|---|
| Single arrow | Success | Success |
| Stop | Success | Success |
| Crosswalk(slanting) | Failure | Success |
| Crosswalk notification | Success | Success |
| Bus stop | Failure | Failure |

Table 4. Compare to binarization method and local change detection method in each objects

## 5.2 Comparison of different filtering sizes

Figures 20-24 show local changing points according to various filter sizes. In the case of filtering sizes of 1 and 3, as shown in Figures 20 and 21, there is much noise. In the case of a filtering size of 5, as shown in Figure 22(a), the noise is almost removed. However, in extracting traffic marking using these points, as shown in Figure 22 (b), the remaining noise results in the creation of incorrect connections.

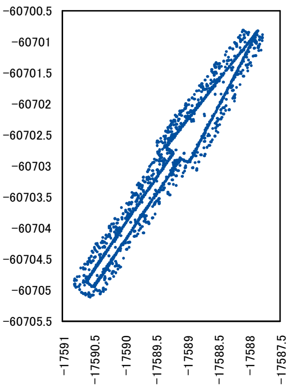


Figure 20. Filter size=1 (no filter)

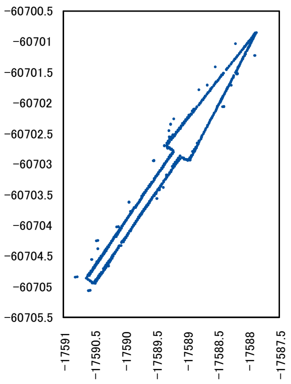
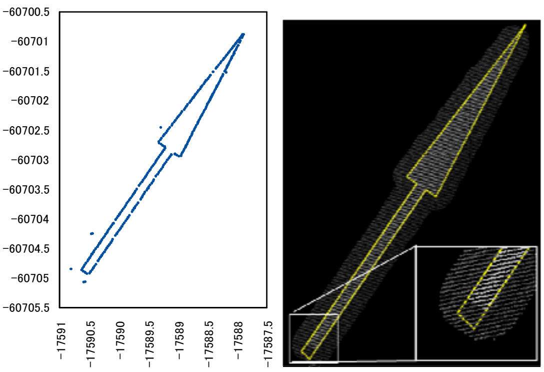
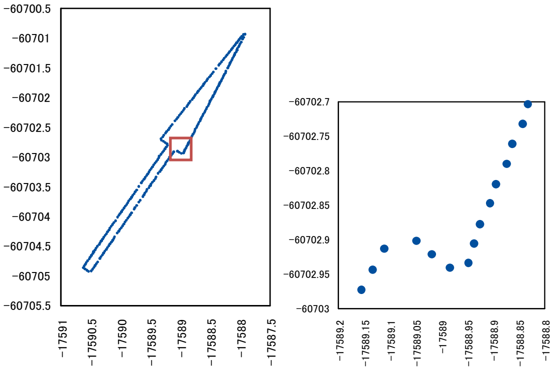


Figure 21. Filter size=3



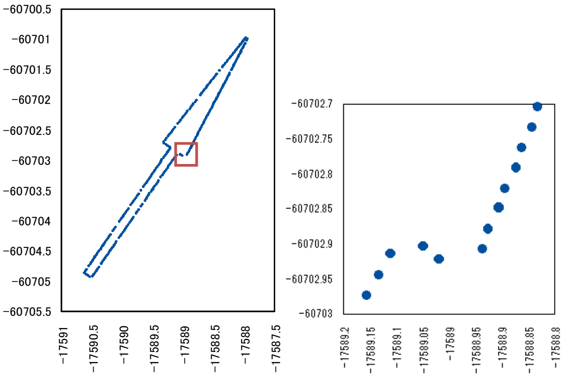
(a) Changing points (b) Drawing data

Figure 22. Filter size=5



(a) All points view (b) Expand view

Figure 23. Filter size=7



(a) All points view (b) Expand view

Figure 24. Filter size=9

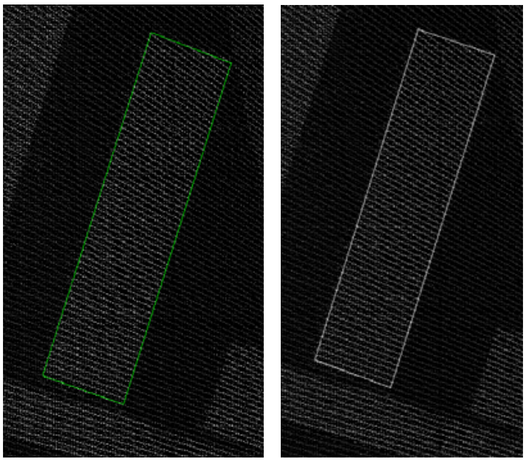
On the other hand, in the case of filtering sizes of 7 and 9, as shown in Figures 23 and 24, all noise is removed, but in the case of a filtering size of 9, necessary points are also removed as shown in Figure 24(b).

When we unnecessarily use a large filter, the corners of traffic markings and narrow change areas are missing. As a result of the above analysis, we determine that a filtering size of 7, the minimum size that removes almost all noise, is optimum for noise reduction.

## 5.3 Quantitative comparison

**5.3.1 Repeatability of manual and automatic**: Automatic and manual drawing images are shown in Figure 25, and the result of comparing the quantitative reproducibility of these is shown in Table 5. This table shows the standard deviation of area. Though it includes randomizing noise, the standard deviation of area under automatic drawing is still less than under manual drawing. From the above, this shows that the automatic method has high repeatability.

**5.3.2 Comparison of binarization and local change detection**: We try to compare the differences of traffic marking areas from manual drawing about automatic drawing (binarization method) and automatic drawing (local changing method). This result is shown in Figure 26.



(a) Manual (b) Auto

Figure 25. Compare to extract result

| | Manual | Auto |
|---|---|---|
| Standard deviation ($m^2$) | 0.0049 | 0.0044 |

Table 5. Comparison of manual and automatic methods through standard deviation of area

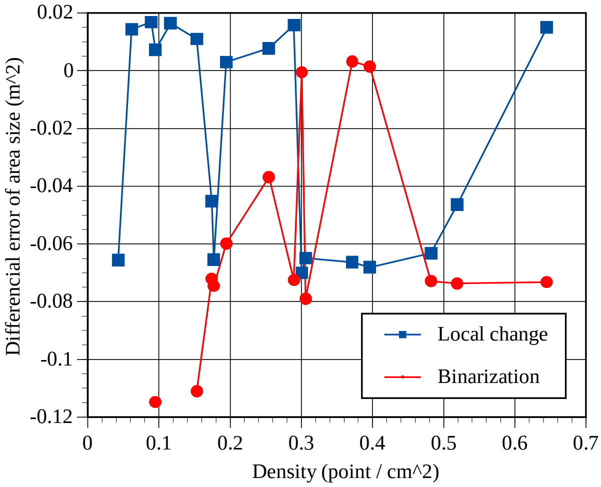


Figure 26. Area size comparison

The result shows that the binarization method is worse than other methods. Especially, areas under the binarization method are smaller than probable values, due to what we believe is the TIN length threshold having the same value in all areas. Furthermore, the result shows that we cannot extract some low density polygons through the binarization method by the same reason.

In comparison, we can see that results from the local changing method is consistent with manual drawing, thus quantitatively demonstrating the high accuracy of our improved method.

## 6. CONCLUSIONS

In this paper, we discussed automatic traffic marking extraction using MMS high-density LIDAR data. The conclusions are as follows.

- Our method can extract traffic markings made up only of straight lines but with complex shapes, for drawing with minimum vertices. On the other hand, it had a problem with extracting short sides.

- We evaluated the quantitative reproducibility of automatic drawings as compared to manual from standard deviation of the area of extracted traffic markings. Even when including random noise, the reproducibility of automatic drawing was as high as manual.

- We compared traffic marking size through binarization and local changing methods. We found that the local changing method is more accurate than the binarization method.

## 7. FUTURE WORK

Future work is as follows.

- On the technical side, it is necessary to support curved sections. When we use a generalized Hough transform, we can expect to need to extract traffic marking with curves. We also can expect a need to improve drawing accuracy.

- We may generalize to automated data extraction when MMS has been run multiple times. On the advanced side, if we perform reconciliation on this extraction data, we can expect to use the resulting drawing data for reference points.

## 8. REFERENCES

Bishen. Y, et al., Automated extraction of road markings from mobile LIDAR point clouds, *Photogrammetric Engineering & Remote Sensing*, Vol. 78(4), 2012, pp. 331-338.

Davies, E. R., 1988, Application of the generalized Hough transformation to corner detection, *IEEE proceedings*, Vol. 135, Pt. E, No.1.

J. Chris McGlone., 2004. *Manual of Photogrammetry fifth edition*, ASPRS, pp. 482-499.

Paul. H., “Machine analysis of bubble chamber pictures,” in Int. Conf. High Energy Accelerators and Instrumentation, 1959, pp. 554-556.

Rabbani, T. et al., 2005, Efficient Hough transform for automatic detection of cylinders in point clouds, *ISPRS Proceedings. Workshop Laser scanning 2005*, pp. 60-65.

Richard O. D. et al., 1972, Use of the Hough transformation to detect Lines and curves in pictures, *Graphics and Image Processing*, 15(1), pp. 11-15.

RIEGL websites, http://www.riegl-japan.co.jp/product/pdf_1/DataSheet_VQ-250_rund_25-09-2012.pdf (12 May. 2014 Access)

Road traffic act websites, http://law.e-gov.go.jp/htmldata/S35/S35HO105.html (7 Mar. 2014 Access)

Trimble websites, http://www.trimble.com/imaging/Trimble-MX8.aspx (12 May. 2014 Access)