

## LIDAR-INCORPORATED TRAFFIC SIGN DETECTION FROM VIDEO LOG IMAGES OF MOBILE MAPPING SYSTEM

Y. Li<sup>a</sup>, J. Fan<sup>a</sup>, Y. Huang<sup>a,\*</sup>, Z. Chen<sup>a</sup>

<sup>a</sup> School of Remote Sensing and Information Engineering, Wuhan University, 129 Luoyu Rd,  
Wuhan, China, 430079 -

(Liyang1273, hycwhu, zyichen)@whu.edu.cn, fanjiawuda@126.com

Commission I, WG I/Va

**KEY WORDS:** Traffic Sign Detection, Mobile Mapping System, RANSAC, Random Forest, Camshift, Kalman Filtering

### ABSTRACT:

Mobile Mapping System (MMS) simultaneously collects the Lidar points and video log images in a scenario with the laser profiler and digital camera. Besides the textural details of video log images, it also captures the 3D geometric shape of point cloud. It is widely used to survey the street view and roadside transportation infrastructure, such as traffic sign, guardrail, etc., in many transportation agencies. Although many literature on traffic sign detection are available, they only focus on either Lidar or imagery data of traffic sign. Based on the well-calibrated extrinsic parameters of MMS, 3D Lidar points are, the first time, incorporated into 2D video log images to enhance the detection of traffic sign both physically and visually. Based on the local elevation, the 3D pavement area is first located. Within a certain distance and height of the pavement, points of the overhead and roadside traffic signs can be obtained according to the setup specification of traffic signs in different transportation agencies. The 3D candidate planes of traffic signs are then fitted using the RANSAC plane-fitting of those points. By projecting the candidate planes onto the image, Regions of Interest (ROIs) of traffic signs are found physically with the geometric constraints between laser profiling and camera imaging. The Random forest learning of the visual color and shape features of traffic signs is adopted to validate the sign ROIs from the video log images. The sequential occurrence of a traffic sign among consecutive video log images are defined by the geometric constraint of the imaging geometry and GPS movement. Candidate ROIs are predicted in this temporal context to double-check the salient traffic sign among video log images. The proposed algorithm is tested on a diverse set of scenarios on the interstate highway G-4 near Beijing, China under varying lighting conditions and occlusions. Experimental results show the proposed algorithm enhances the rate of detecting traffic signs with the incorporation of the 3D planar constraint of their Lidar points. It is promising for the robust and large-scale survey of most transportation infrastructure with the application of MMS.

### 1. INTRODUCTION

The improvement of Intelligent Transportation System is not only beneficial to our daily transportation but also gathering more intelligence in predicting the possible risk of driving. Traffic sign plays a significant role in regulating and controlling traffic activities, and ensures a safe and smooth traffic. The accurate detection and localization information of traffic signs are necessary for many intelligent transportation-related applications, like those systems of autonomous driving and driver assistance. As a result, automated traffic sign detection and recognition techniques are crucial for the transportation agencies to update the traffic sign inventory timely, and to improve the traffic quality and safety quickly.

Traditionally, most of the traffic sign detections are based on textual and color details of video log images. But only images can hardly achieve precise detection results. It is challenging to cope with the complex texture and color corruption in the urban environment. With the laser profiler and digital camera, mobile mapping system (MMS) provides an effective way for acquiring very dense point clouds as well as road video log images in a scenario. Mobile laser scanning (MLS) has proven to be very efficient in acquiring very dense point clouds (over 800 points per square meter) along road corridors. Integrated in a mobile mapping system, the data acquired by laser scanners can be used to robustly capture the geometry of the road environment and be the basis for the recognition of a wide range of objects.

Traffic sign detection can be classified into color-based, shape-based, or both. Different color spaces have been used, for instance, HSI-HSV (Fleyeh, 2006, Gomez-Moreno et al., 2010), YUV (Shadeed et al., 2003) or Gaussian color model (Li et al., 2015) as a visual feature to define a traffic sign region. Shape features have also been studied, such as Hough Transform (Barile et al., 2008), Local Contour Pattern (Landesa-Vazquez et al., 2010), or Local Binary Patterns (Liu et al., 2014). There are many intense efforts in computer vision (Heng et al., 2011, Crandall et al., 2011, Heng et al., 2011) focus on image-based 3D reconstruction at large-scale internet imagery. For example, image-based 3D point clouds and semantic texton forests are used to segment and recognize the highway assets (Golparvar-Fard et al., 2012). The color-coded point clouds and the geo-registered video frames are integrated together which enables a user to conduct visual walk through and query different categories of assets. Semantic texton forests and SVM (Support Vector Machine), proposed by (Golparvar-Fard et al., 2012, Yang et al., 2015) are used to recognize the traffic signs. (Riveiro et al., 2015) is focused on the detection and classification of retro-reflective vertical traffic signs for their function (danger, give way, prohibition/obligation, and indication) from mobile laser scanning data by considering geometric and radiometric information. (Yu et al., 2016) achieve recognition task by using Gaussian-Bernoulli deep Boltzmann machine-based hierarchical classifier on 2-D images. They focus on the detection of vertical traffic signs in 3D point clouds acquired by a LYNX Mobile Mapper system, comprised of laser scanning and RGB cameras (Soiln et al., 2016).

\*Corresponding author

However, to take full advantage of using MMS, they must be used in optimal applications. Unlike other literatures, the 3D point cloud will play a special role in traffic sign detection. In this study, we focus on the detection and tracking of traffic signs using 3D Lidar points and 2D video log images by considering the imaging geometry and GPS movement. The global strategy for detection and tracking by filtering the noise point through setting distance and elevation thresholds, the rough area of traffic sign can be obtained. Then we project the candidate planes fitted by RANSAC onto the image, regions of Interest (ROIs) of traffic signs are localized in the video log images. At this stage, Random Forest is adopted to detect the traffic signs among these ROIs. Finally, a tracking algorithm is proposed to analyze this temporal context by combining the Camshift and Kalman filtering together.

## 2. TRAFFIC SIGN LOCALIZATION

The proposed traffic sign detection algorithm first focuses on using the distance and elevation information to segment the traffic signs from the point cloud. By filtering the noise point through setting distance and elevation thresholds, the rough area of traffic sign can be obtained. Then we project the candidate planes fitted by RANSAC onto the image, regions of Interest (ROIs) of traffic signs are localized in the video log images.

### 2.1 Pre-processing

We need to handle laser point clouds to reduce the amounts of data to process or it will be time-consuming. The 3D data we acquire includes pavements, traffic signs, buildings, billboards, it is important to extract the local interest information instead of the whole data. Those points consist mainly of two types: one is on the ground such as traffic signs, billboards, trees and guardrails; the other is ground segments like pavements, lane markings. Considering the data of each photo contains the trajectory position of the vehicle, the range containing the road as well as the traffic sign can be located temporally from a large scenario. Within a certain distance and height of the pavement boundary and plane respectively, points of the overhead and roadside traffic signs can then be obtained through the setup specification of traffic signs.

Since the point cloud data contains huge information and involves in thousands of pictures like Figure.3(a), we need to set values for distance and elevation to obtain necessary points for single image. The classification starts with the distance from the 3D points to the sensor is computed, and points further than 20 meters are filtered out. At this case, it is obvious that the remaining points contain a set of points that belong to traffic signs, but they still include points on the floor, on lane markings and on nearby buildings, as Figure.3(b) Among the whole points, traffic sign points are mostly in higher altitude compared to other belongings. According to the Chinese road traffic laws, the traffic sign is placed on the edge of the road shoulder, and its height is 2 to 2.5 meter away from the road. We reserve those points whose elevation is 1.8 meters higher than the average, the general road plane's height. As we can see from the Figure.1(c), only the information of the traffic sign where the MMS is traveling will be kept for further processing.

### 2.2 RANSAC

After the first pre-processing step, the remaining point clouds contain mainly the road sign. The 3D candidate planes of traffic signs are then fitted using the RANSAC plane-fitting strategy

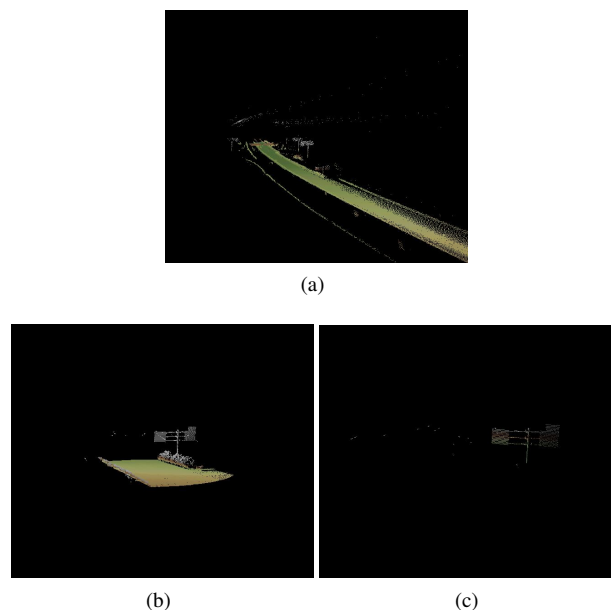


Figure 1: Different results after pre-processing. (a) Original point clouds; (b) The remaining points after certain distance is set; (c) Only the points around the traffic sign is reserved.

among those points.

The RANSAC algorithm is a learning technique to estimate parameters of a mathematical model from a set of observed data contains both inliers and outliers. Inliers can be explained by a model with a particular set of parameter values, while outliers do not fit that model in any circumstance. The voting scheme is used in RANSAC to find the optimal fitting result. The implementation of this voting scheme is based on two assumptions: the entire data we are observing consists of both inliers and outliers, and a process which can optimally estimate the available parameters of the chosen model from the inliers. The input to the RANSAC algorithm is a set of observed data values including point clouds produced from previous step, a way of fitting some kind of model to the observations, and some confidence parameters. The RANSAC algorithm is essentially composed of two steps that are iteratively repeated:

- Firstly, a sample subset containing minimal data items is randomly selected from the input dataset. We can get the fitting model and the corresponding model parameters by computing the elements of this sample subset. The cardinality of the sample subset is the smallest sufficient to determine the model parameters.
- Secondly, for all the points in our data, the distance to the fitting model is computed. A data element will be considered as an outlier if its distance is larger than the threshold we set before the procedure.

The set of inliers obtained for the fitting model is called consensus set. This procedure is repeated a fixed number of times, each time producing either a model which is rejected because too few points are part of the consensus set, or a refined model together with a corresponding consensus set size. In the latter case, we keep the refined model if its consensus set is larger than the previously saved model. The estimated model which contains the most inliers is considered as the optimal model, final figure is shown as Figure.2.

An advantage of RANSAC is its ability to estimate the model parameters robustly. But, RANSAC can only do the estimation of the parameter model we want for a particular data set, i.e. if the data contains traffic sign and greenbelts, the plane fitted by RANSAC may fail to find either one.

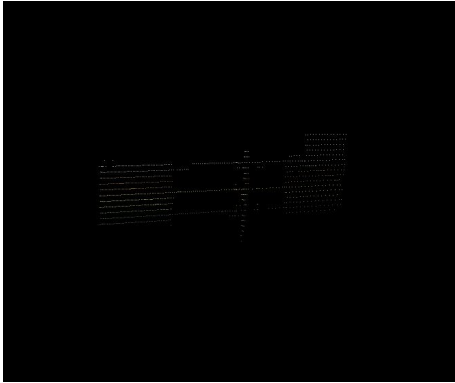


Figure 2: The plane result fitted by RANSAC.

### 2.3 Projection

During the third phase of our procedure, the projection of traffic signs plane is accomplished by using co-linearity equation, via shooting rays from the image to the 3D geometry. Traditional road sign detection is carried out using 2D image only, this may would not guarantee high accuracy. Also, the spatial resolution of a point cloud is not enough to recognize traffic sign. The best source of information that the MMS provides for the recognition task are RGB cameras, whose internal calibration and external orientation parameters with respect to the vehicle are known. The relative registration between laser point clouds and array C-CD images is achieved by using POS data and relative position of each sensor.

Let  $x, y$  refer to a coordinate system with the  $x$ -axis and  $y$ -axis in the sensor plane. Denote the coordinates of the point P on the object by  $x_p, y_p, z_p$ , the coordinates of the image point of P on the sensor plane by  $x$  and  $y$  and the coordinates of the projection (optical) center by  $x_0, y_0, z_0$ . As a consequence of the projection method there is the same fixed ratio between  $x - x_0$  and  $x_0 - x_p$ ,  $y - y_0$  and  $y_0 - y_p$  the distance of the projection center to the sensor plane  $z_0 = c$  and  $z_p - z_0$ . Hence:

$$x - x_0 = -\lambda(x_p - x_0) \quad (1)$$

$$y - y_0 = -\lambda(y_p - y_0) \quad (2)$$

$$c = \lambda(z_p - z_0) \quad (3)$$

Solving for  $\lambda$  in the last equation and entering it in the others yields:

$$x - x_0 = -c \frac{x_p - x_0}{z_p - z_0} \quad (4)$$

$$y - y_0 = -c \frac{y_p - y_0}{z_p - z_0} \quad (5)$$

The point P is normally given in some coordinate system "outside" the camera by the coordinates  $X, Y$  and  $Z$ , and the projection center by  $X_0, Y_0, Z_0$ . These coordinates may be transformed through a rotation and a translation to the system on the camera. The translation doesn't influence the differences of the coordinates, and the rotation, often called camera transformation, is given by a  $3 \times 3$ -matrix  $\mathcal{R}$ , transforming  $(X - X_0, Y - Y_0, Z -$

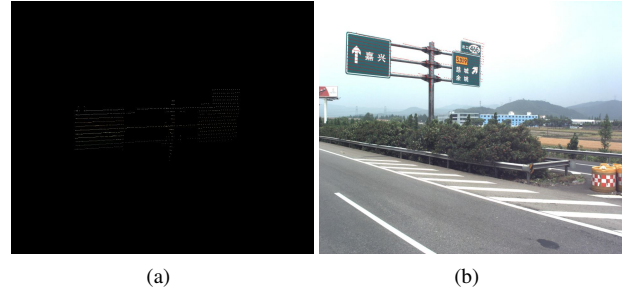


Figure 3: Projection result.

$Z_0$ ) into:

$$x_p - x_0 = R_{11}(X - X_0) + R_{21}(Y - Y_0) + R_{31}(Z - Z_0) \quad (6)$$

$$y_p - y_0 = R_{12}(X - X_0) + R_{22}(Y - Y_0) + R_{32}(Z - Z_0) \quad (7)$$

and

$$z_p - z_0 = R_{13}(X - X_0) + R_{23}(Y - Y_0) + R_{33}(Z - Z_0) \quad (8)$$

Substitution of these expressions, leads to a set of two equations, known as the collinearity equations:

$$x - x_0 = -c \frac{R_{11}(X - X_0) + R_{21}(Y - Y_0) + R_{31}(Z - Z_0)}{R_{13}(X - X_0) + R_{23}(Y - Y_0) + R_{33}(Z - Z_0)} \quad (9)$$

$$y - y_0 = -c \frac{R_{12}(X - X_0) + R_{22}(Y - Y_0) + R_{32}(Z - Z_0)}{R_{13}(X - X_0) + R_{23}(Y - Y_0) + R_{33}(Z - Z_0)} \quad (10)$$

The most obvious use of these equations is for images recorded by a camera. The projection process can be described by transformations from object space  $(X, Y, Z)$  to image coordinates  $(x, y)$ . It indicates that the image point (on the sensor plate of the camera), the observed point (on the object) and the projection center of the camera were aligned when the picture was taken. After projection, the candidate region is located shown as Figure.3(b)

### 3. TRAFFIC SIGN DETECTION

By projecting the candidate planes onto the image, Regions of Interest (ROIs) of traffic signs are found physically with the geometric constraints between laser profiling and camera imaging. At this stage, Random Forest is adopted to detect the traffic signs among these ROIs.

Based on the blob feature of (Vicen-Bueno et al., 2005), a blob of  $24 \times 24$  pixels for each component (R, G, and B) of each ROI of the video log image is sent to random forest classifier. The total dimension of the input vector is 51 nodes. They consists of 3 normalized average maximum pixel values, MR, MG and MB, 24 inputs from the vertical histogram ( $vh$ ) and 24 inputs from the horizontal histogram ( $hh$ ).

Random forests, introduced by Breiman and Cutler (Breiman, 2001), are an ensemble learning method for classification. It is operated by constructing a multiple of decision trees at training time and outputting the class. Each tree in the ensemble is built from a random sample of the original data with replacement from the original training data. The main steps of random forest are as follows:

- Take a subset from the whole set of data (training set).
- The algorithm clusters the data in groups and subgroups. If you would draw lines between the data points in a subgroup,

and lines that connect subgroups into group etc. the structure would look somewhat like a tree. This is called a decision tree. At each split or node in this cluster/tree/dendrogram variables are chosen at random by the program to judge whether data points have a close relationship or not.

- The program makes multiple trees a.k.a. a forest. Each tree is different because for each split in a tree, variables are chosen at random.
- Then the rest of the dataset (not the training set) is used to predict which tree in the forests makes the best classification of the data points (in the dataset the right classification is known).
- The tree with the most predictive power is shown as output by the algorithm.

The random forests achieve state-of-the-art performance in many multi-class classification applications. A further advantage is that they are fast to build, easy to implement in a distributed computing environment

#### 4. TRAFFIC SIGN TRACKING

The sequential occurrence of a traffic sign among consecutive video log images are defined by the geometric constraint of the imaging geometry and GPS movement. Candidate ROIs are analyzed in this temporal context to double-check the salient traffic sign among video log images. In this section, a tracking algorithm is proposed to analyze this temporal context by combining the Camshift and Kalman filtering together. Camshift can track the moving objects quickly and robustly, using the color characteristics. Kalman filtering can predict the most probable object location in the next frame according to the geometric constraints and the updated observations in the current frame.

##### 4.1 Camshift

The Camshift algorithm is based on the color probability distribution of the target, so the changes of the object shape would not affect the result. By calculating similarity of the color probability distribution, the moving target in the current frame image location is taken as the initial location of the next frame. Its core algorithm is the Meanshift algorithm, a probability density estimation based on rapid non-parametric pattern. Meanshift looks for the maxima of a density function and Camshift is the extension of it. The flow of the algorithm is as follows:

1. Set the size of the search window(s) in the color probability distribution.
2. Calculate the zero moments:

$$\mathcal{M}_{00} = \sum_x \sum_y I(x, y) \quad (11)$$

$$\mathcal{M}_{01} = \sum_x \sum_y yI(x, y) \quad (12)$$

$$\mathcal{M}_{10} = \sum_x \sum_y xI(x, y) \quad (13)$$

Where,  $I(x, y)$  is the image element value of coordinates  $(x, y)$ , and  $x$  and  $y$  change in the scope of the search window.

3. Calculate the center of mass for the search window  $(X_c, Y_c)$ :

$$x_c = \frac{\mathcal{M}_{10}}{\mathcal{M}_{00}}, y_c = \frac{\mathcal{M}_{01}}{\mathcal{M}_{00}} \quad (14)$$

4. The size of the search window that is the function of a color probability distribution of the former search window can be re-installed as s.
5. Repeat steps 2, 3, 4 until they are constrigent (the change of the center of mass is less than the threshold value).
6. The major axis l, minor axis w and direction angle of the target can be obtained by calculating the second-order matrix.

$$\mathcal{M}_{11} = \sum_x \sum_y xyI(x, y) \quad (15)$$

$$\mathcal{M}_{20} = \sum_x \sum_y x^2 I(x, y) \quad (16)$$

$$\mathcal{M}_{02} = \sum_x \sum_y y^2 I(x, y) \quad (17)$$

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}} \quad (18)$$

$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \quad (19)$$

$$\theta = \frac{1}{2} \arctan\left(\frac{2b}{a-c}\right) \quad (20)$$

Where

$$a = \frac{\mathcal{M}_{20}}{\mathcal{M}_{00}} - x_c^2, b = 2\left(\frac{\mathcal{M}_{20}}{\mathcal{M}_{00}} - x_c y_c\right), c = \frac{\mathcal{M}_{02}}{\mathcal{M}_{00}} - y_c^2 \quad (21)$$

Following the aforementioned procedures yields a window which is very likely to contain the original target. However, if the occlusion or large-scale similar color interference occur during tracking process, Camshift algorithm will fail. In order to deal with such problem, we improves the Camshift algorithm in two ways: background subtraction algorithm and kalman algorithm.

##### 4.2 Kalman filter

Before we do Camshift method, we need to judge the effects of background to the tracking. If the background color is too similar to the object, we need to use background subtraction first. It is a motion detection algorithm begins with the segmentation part where foreground or moving objects are segmented from the background. The simplest way to implement this is to take the previous image as background and take the current frames, denoted by I to compare with the background image denoted by B. Here using simple arithmetic calculations, we can segment out the objects simply by using image subtraction technique of computer vision meaning for each pixels in I, take the pixel value denoted by  $P[I]$  and subtract it with the corresponding pixels at the same position on the background image denoted as  $P[B]$ .

$$P[F] = P[I] - P[B], \quad (22)$$

Where the difference image denoted as P[F], which would only show some intensity for the pixel locations which have changed between the two frames. Then the difference image will be sent to Camshift procedure for further processing. After that, the kalman filter is adopted here to estimate the parameters of the moving

targets. The key of kalman filter is prediction and update. The state vector  $X_k = [x, y, V_x, V_y]$ , measurement vector is  $Y_k = [x, y]^T$ , Where  $x$  and  $V_x$  are the target image in the horizontal direction of the position and velocity;  $y$  and  $V_y$  are the target image in the vertical position and velocity. The state equation of the system is (23), and observation equation is (24).

$$X_{k+1} = A_k X_k + W_k \quad (23)$$

$$Y_{k+1} = H_k X_k + V_k \quad (24)$$

Where  $A_k$  is state transition matrix.  $Y_k$  is measurement state of system.  $H_k$  is observation matrix.  $W_k$  is dynamic noise corresponding to state vector, and  $V_k$  is measurement noise corresponding to observation vector. The equation of prediction and update are as follows:

Prediction-equation1:

$$X_k' = F * X_{k-1} \quad (25)$$

Prediction-equation2:

$$P_k' = F \cdot P_{k-1} \cdot F^T + Q \quad (26)$$

Kalman-gain-equation:

$$K_k' = P_k' \cdot H^T \cdot (H \cdot P_k' \cdot H^T + R)^{-1} \quad (27)$$

Update-equation1:

$$X_k = X_k' + K_k \cdot (Z_k - H \cdot X_k') \quad (28)$$

Update-equation2:

$$P_k = P_k' - K_k \cdot H \cdot P_k' \quad (29)$$

where  $F$  is state transition matrix,  $H$  is measurement matrix,  $Q$  represents process noise covariance matrix and measurement noise covariance matrix is  $R$ .

The main steps are as follows:

- Whether the background color is similar to the object, if it is, we do background subtraction first.
- Initialize a search window.
- Search the target using CamShift algorithm in the estimated range to locate the possible position of target.
- Estimate the appearing position of target in the next moment by Kalman filter.
- If the distance of Camshift predicted center and kalman's center is in the threshold, the search is seen as a success, and the observed value of Kalman filter will be regarded as the next window position.

## 5. EXPERIMENTAL RESULTS

In this section, the proposed algorithm is tested for the accuracy of traffic sign detection using the video log images and 3D point cloud of the MMS. The data is captured on May 19th 2013 in Ningbo, China and provided by Ledor Spatial company. All the detection and tracking methods are programmed with VS2010 and are executed on a PC of 2.3GHz Pentium 4 with 6GB RAM with the Windows 64 bits operating system. The size of video image is  $2448 \times 2048$ .

The experiments demonstrated the detection and tracking of the traffic sign. For most traffic sign which only appears in few images, we project our 3D point cloud onto the first image involving the traffic sign. After projecting the fitted planes onto the image, regions of interest of traffic signs are localized in the image. Then we use random forest to detect our traffic sign, the algorithm obtained a higher accuracy than traditional methods, which improves our detection efficiency quickly. Then we initialize the search window for tracking according to the ROIs. The background of the video log image is similar to what we need to track, thus the search window we set is larger than ROIs but share the same center. By using Camshift and Kalman filter, our tracking outputs are shown as Figure 4. In most cases, we can track the object successfully. One reason is the well-fused 3D point cloud with the video log image, through which we can narrow the search region in our images and eliminate the noise interference effectively. Besides, the RANSAC algorithm helps us screening out some noise points by fitting a cluster of point into a plane of a limited tolerance. If we track the object on a consecutive images successfully, for instance, in the first four images, we can track the object in three or more images, a conclusion can be made that we definitely detect a traffic sign.

However, when traffic signs are mixed up together with trees, our pre-processing sometimes may fail to localize the traffic sign, as shown in Figure.5. The traffic sign sits in the bush, our RANSAC fitting of planes can not achieve high accuracy with the disturbance of those points of trees, as shown in Figure.6. The contrast between our traffic sign and background becomes low in Figure.7, where we can't predict the right position and track the sign timely, as shown in Figure.8.



Figure 5: The traffic sign sits in the bush.

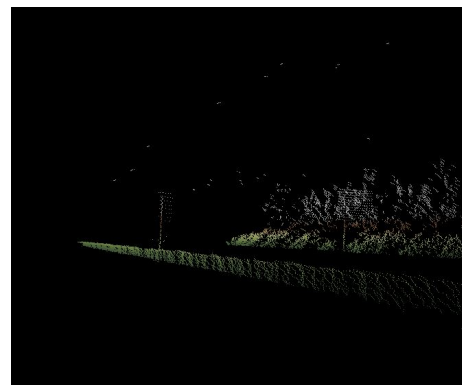


Figure 6: The pre-processing result.





Figure 4: Final results of our algorithm. Our method has been tracked in the possible position according to geometric constraint of the imaging geometry and GPS movement. doi:10.5194/isprsarchives-XLI-B1-661-2016



Figure 7: The background of the video log image is similar to what we need to track.



Figure 8: The predicted position of the traffic sign is deviated from the right position.

## 6. CONCLUSION AND RECOMMENDATIONS

Mobile Mapping System (MMS) collects the Lidar points by the laser profiler, as well as the video log images in a scenario with the digital camera. However, most literature focuses on the textural/color details of video log images and the 3D geometric shape of point cloud separately. In this paper, 3D Lidar points are incorporated into 2D video log images to enhance the detection of traffic sign both physically and visually. By filtering the noise point through setting distance and elevation thresholds, the rough area of traffic sign can be obtained. Then we project the candidate planes fitted by RANSAC onto the image, regions of Interest (ROIs) of traffic signs are localized and sent to random forest classifier. Candidate ROIs will be tracked using the Camshift and Kalman filter among consecutive video log images. The proposed algorithm is tested on video logs with the size of  $2448 \times 2048$ . Experimental results show the proposed algorithm can not only detect traffic signs but also track the next possible occurrence of candidate ROIs. It is promising for automatic identification of traffic signs.

Although the proposed algorithm demonstrates its capability for traffic sign detection, we recommend that:

- Reduce interference of billboards using the rule, which is located regularly alongside the pavement and larger than most road signs.
- Increase the exposure in areas with poor weather conditions, which will refine the candidate ROIs in tracking the similar color.

- Innovative formulation and application of such process as Camshift and Kalman filter.

## ACKNOWLEDGEMENTS

The authors are grateful for the support of the program of Low-cost GNSS / INS Deep-coupling System of Surveying and Mapping (Program NO 2015AA124001), State Key Technology Research and Development Program (863), China. This work was also jointly supported by the National Natural Science Foundation of China (No. 41271431), and the science and technology plan of sichuan bureau of surveying, mapping and geoinformation (No. J2014ZC02).

## REFERENCES

- Barrile, V., Meduri, G. M. and Cuzzocrea, D., 2008. Automatic recognition of road signs by hough transform. *Journal of Earth Science & Engineering*.
- Breiman, L., 2001. Random forests. *Machine learning* 45(1), pp. 5–32.
- Crandall, D., Owens, A., Snavely, N. and Huttenlocher, D., 2011. Discrete-continuous optimization for large-scale structure from motion. In: *IEEE Conference on Computer Vision & Pattern Recognition*, pp. 3001–3008.
- Fleyeh, H., 2006. Shadow and highlight invariant colour segmentation algorithm for traffic signs. In: *Cybernetics and Intelligent Systems, 2006 IEEE Conference on*, pp. 1–7.
- Golparvar-Fard, M., Balali, V. and Garza, J. M. D. L., 2012. Segmentation and recognition of highway assets using image-based 3d point clouds and semantic texton forests. *Journal of Computing in Civil Engineering*.
- Gomez-Moreno, H., Maldonado-Bascon, S., Gil-Jimenez, P. and Lafuente-Arroyo, S., 2010. Goal evaluation of segmentation algorithms for traffic sign recognition. *IEEE Transactions on Intelligent Transportation Systems* 11(4), pp. 917 – 930.
- Heng, L., Lee, G. H., Fraundorfer, F. and Pollefeys, M., 2011. Real-time photo-realistic 3d mapping for micro aerial vehicles. In: *IEEE/RSJ International Conference on Intelligent Robots & Systems IEEE/RSJ International Conference on Intelligent Robots & Systems*, pp. 4012–4019.
- Landesa-Vazquez, I., Parada-Loira, F. and Alba-Castro, J., 2010. Fast real-time multiclass traffic sign detection based on novel shape and texture descriptors. In: *Conference Record - IEEE Conference on Intelligent Transportation Systems*, pp. 1388–1395.
- Li, H., Sun, F., Liu, L. and Wang, L., 2015. A novel traffic sign detection method via color segmentation and robust shape matching. *Neurocomputing* 169, pp. 77–88.
- Liu, C., Chang, F. and Chen, Z., 2014. Rapid multiclass traffic sign detection in high-resolution images. *IEEE Transactions on Intelligent Transportation Systems* 15(6), pp. 2394–2403.
- Riveiro, B., Diaz-Vilarino, L., Conde-Carnero, B. and Soilan, M., 2015. Automatic segmentation and shape-based classification of retro-reflective traffic signs from mobile lidar data. *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing* pp. 1–9.

Shadeed, W. G., Abu-Al-Nadi, D. I. and Mismar, M. J., 2003. Road traffic sign detection in color images. In: Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference on, pp. 890 – 893 Vol.2.

Soiln, M., Riveiro, B., Martínez-Sánchez, J. and Arias, P., 2016. Traffic sign detection in mls acquired point clouds for geometric and image-based semantic inventory. *Isprs Journal of Photogrammetry & Remote Sensing* 114, pp. 92–101.

Vicen-Bueno, R., Gil-Pita, R., Jarabo-Amores, M. and Lopez-Ferreras, F., 2005. Complexity reduction in neural networks applied to traffic sign recognition tasks. In: Signal Processing Conference, 2005 13th European, IEEE, pp. 1–4.

Yang, M., Gu, X., Lu, H., Wang, C. and Ye, L., 2015. Panorama-based multilane recognition for advanced navigation map generation. *Mathematical Problems in Engineering* 2015, pp. 1–14.

Yu, Y., Li, J., Wen, C., Guan, H., Luo, H. and Wang, C., 2016. Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data. *Isprs Journal of Photogrammetry & Remote Sensing* 113, pp. 106–123.