ISSN 1009 - 3095 *Journal of Zhejiang University*(*SCIENCE*) V.2, No.4, P.355 - 394, Oct. - Dec., 2001
http://www.periodicals.com.cn; http://www.zju.edu.cn;
http://lib.zju.edu.cn/English; http://www.zjupress.com

OBSTACLE DETECTION BY ALV USING TWO 2D LASER RANGE FINDERS*

XIANG Zhi-yu(项志宇), LIU Ji-lin(刘济林),
GU Wei-kang(顾伟康), TIAN Yuan(田　元)

(*Department of Information and Electronic Engineering, Zhejiang University, Hangzhou 310027, China*)

Received Oct.21, 2000; revision accepted Mar.8, 2001

Abstract: This paper describes an effective method of obstacle detection by ALV (Autonomous Land Vehicle) equipped with two 2D laser range finders (LRF) installed at different locations of the ALV to obtain comprehensive information on the environment. The data processing includes two main steps: (1) data-processing of the current sample; (2) fusion of the former range data and the current one. The rough description of the ALV's environment via the four sub-steps (Data Filter, Obstacle Extraction, Obstacle Merging, Distinguishing Obstacle from Road-Edge) was not reliable enough for our control system. To overcome the shortcoming of the 2D LRF and the motion noise of the ALV, a Kalman filter was used to estimate the position of the obstacles; then the data of the two LRFs were collated to obtain the height and width of the obstacles. Experiment results attested the feasibility of the detection system.

Key words: obstacle detection, obstacle avoidance, 2D laser range finder, Kalman Filter, autonomous land vehicle(ALV)
Document code: A **CLC number:** TP242.6

INTRODUCTION

Obstacle avoidance is an essential task of the ALV. To achieve this, a quick and reliable detection of obstacles is very important. Several classes of active sensors can be used for this purpose.

A laser range finder (LRF) can suitably be used to detect obstacles ahead of the ALV. 3D LRFs often used in previous applications (Ying et al., 1998; Bonlanger, 1992; Dunlay, 1986; Herbert et al., 1986) could yield relatively comprehensive information, but are very costly because of their complexity (Zhang et al., 1997). What is more, its obvious shortcomings (such as bulky and cumbersome body, and low sampling rate) restrict its further application.

Compared with 3D LRF, 2D LRF has higher sampling rate, less system errors, a great advantage in realtime signal processing; and its relatively lower price and higher reliability makes it widely used in various indoor mobile robots.

Most 2D LRF applications focus on map building for mobile robots in indoor structural environment (Kwon et al., 1997; Vandorpe et al. 1996; Gonzalez et al., 1994) and localizing mobile robots with a prior map and the current scans of LRF (Mashavan et al., 1998; Gonzales et al., 1992). LRFs are used in this study to achieve obstacle detection and roadside recognition in a non-structural outdoor environment. We hope we can dynamically update the environmental map in realtime by fully utilizing the 2D LRF's character of high sampling rate. To gain sufficient information, we mount two LRFs at different places. We pay attention to the updating process of the local map. Firstly, we process the current LRF data to obtain a rough description of the obstacles' distribution in the current obstacle map. Then Kalman filter technique is adopted to track the obstacles and get a better estimation of the obstacles' position. Finally, by collating data provided by the two LRFs, the width and the height of the obstacle can be obtained.

* Project supported by Ninth-Five primary research foundation of National Defence of China

LRFs SYSTEM

Fig. 1 shows two LRFs' installed on the ALV. The 2D LRFs were both commercial eye-safe laser scanners (LMS210 series made by Sick Optic Electronic).

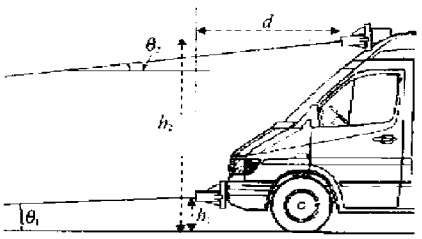

Fig.1 The LRFs and ALV

The ALV's obstacle detection system contained a top LRF with 100° scanning angle and 0.25° resolution; and a bottom LRF with 100° scanning angle and 0.5° resolution. Thus compared with the top LRF, the bottom one takes half the time for a complete scanning, 40 ms. The effective range of both LRFs was 50 m. The parameters in Fig.1 are as follows:

$$\theta_1 < 1°;\ \theta_2 = 5.7°;\ h_1 = 40\ \text{cm};$$
$$h_2 = 290\ \text{cm};\ d = 210\ \text{cm}.$$

1. Different function of the two LRFs

The bottom LRF can detect obstacles rapidly and reliably within its effective region due to its higher sampling rate. On the other hand, the top LRF can easily find obstacles farther away due to its higher resolution and its view is relatively free from being shutoff as it is mounted in higher location. As we know, we cannot get the height of the obstacles with the bottom LRF whose scanning plane parallels the ground, therefore another important task of the top LRF is to obtain the approximate height of obstacles. Fig.2 shows the relationship between the top LRF and the obstacle.

In Fig.2, H_0 and L_0 are the mounting parameters of the top LRF, and H_1 and L_1 the height and range of the obstacle. Based on this information, we can obtain the equation:

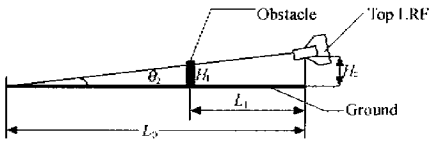

Fig.2 The relationship between top LRF and obstacle

$$H_1 = H_0 - L_1 \frac{H_0}{L_0} \tag{1}$$

Thus from Eq. (1), if the top LRF finds the obstacle when the range between the obstacle and ALV is L_1, we can conclude that the obstacle must be higher than H_1. With the running of the ALV, we can update the obstacle height in realtime until the obstacle cannot be seen by the top LRF. At that time, the approximate height of the obstacle can be acquired.

2. Calibration of range data

To directly use the result of obstacle detection in local path planning, 2D-cartesian coordinates must be set up for an obstacle map, where the plane parallels the ground, the origin point is the center of the bottom LRF, and the Y-axis of the map is the central axis of the ALV.

Only one rotation transformation is needed to convert the bottom LRF's measurement data into obstacle coordinates. If $\rho(i)$ and δ represent the range data and angle resolution respectively, α represents the angle between the central axis of the bottom LRF and the Y-axis, then the vector $\boldsymbol{X}_B = [x_b(i), y_b(i)]^T$ represents the result of transformation:

$$\begin{aligned} x_b(i) &= \rho(i) \times \cos[(40 + \alpha + i \times \delta)\pi/180] \\ y_b(i) &= \rho(i) \times \sin[(40 + \alpha + i \times \delta)\pi/180] \end{aligned} \tag{2}$$

For the top LRF, the transformation contains two steps:

1) Convert the raw range data to the plane paralelling the ground. If the slope angle θ_2 in Fig.1 is known, we have temporary vector $\boldsymbol{X}_{\text{temp}} = [x_t(i), y_t(i)]^T$ where,

$$\begin{aligned} x_t(i) &= \rho(i) \times \cos(\theta_2) \times \cos[(40 + i \times \delta)\pi/180] \\ y_t(i) &= \rho(i) \times \cos(\theta_2) \times \sin[(40 + i \times \delta)\pi/180] \end{aligned} \tag{3}$$

2) Convert the $\boldsymbol{X}_{\text{temp}}$ into coordinates of the obstacle map. As we know, the transformation between 2D coordinates is nothing but displacement and rotation. With $R = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix}$ representing the rotatory matrix, $\boldsymbol{T} = \begin{bmatrix} X_S \\ Y_S \end{bmatrix}$ representing the displacement vector, we have:

$$\boldsymbol{X}_{\text{R}} = R\boldsymbol{X}_{\text{temp}} + \boldsymbol{T} \tag{4}$$

where $\boldsymbol{X}_{\text{R}}$ is the final representation of the top LRF's range data in obstacle coordinates.

The goal of calibration is to work out the matrix R and vector $\boldsymbol{T}$. From Formula (4) we know that there are 6 parameters to be calculated. In practice, we can get several couples of $\boldsymbol{X}_{\text{R}}$ and $\boldsymbol{X}_{\text{temp}}$ acquired by the two LRFs to resolve the matrix R and vector $\boldsymbol{T}$. To improve the precision of the calibration, the least square error method with more than three couples of corresponding points is used.

PROCESSING OF CURRENT SAMPLE

To obtain the up-to-date representation of the ALV's surrounding, the processing of the current sample is necessary. It contains 4 substeps: Data filtering, Obstacle extracting, Obstacle merging and Distinguishing obstacles from brushwood on the roadsides.

1. Data filtering

Median filtering is used to reduce the noises produced by the LRF. In addition, some outside the present area data are discarded.

2. Obstacle extracting

In the obstacle map, the measured points are shown as isolated clusters within which the range between points are relatively small. We can cluster the points which lie close enough to each other by a criteria of range threshold. For real obstacles on the road, the threshold should be relatively small to keep high resolution, but for brushwood standing on roadsides, we try to cluster them into one group. Thus we give small range threshold of 80 cm to the narrow region ahead of the ALV and 150 cm to the rest region.

After clustering, the feature of each cloud of points is extracted by feature extraction module. With the processing time restricted, a complex algorithm is unpractical. So least square line fitting is used to obtain the length of segments, the orientation of segments, position of central point, size of enclosing rectangle, etc.

Let (x_i, y_i) represent the coordinates of points in a cluster which includes n measured points totally; the intermediate line parameters are (m, q) from the linear equation $y = mx + q$ (regression y to x) or (s, t) from the linear equation $x = sy + t$ (regression x to y). The choice between (m, q)- and (s, t)-parameters depends on the slope of the line in obstacle coordinates. The regression parameters R_x, R_y, R_{xx}, R_{yy}, R_{xy} are given by following expressions:

$$R_x = \sum_{i=1}^{n} x_i;\ R_y = \sum_{i=1}^{n} y_i;\ R_{xx} = \sum_{i=1}^{n} x_i^2;$$
$$R_{yy} = \sum_{i=1}^{n} y_i^2;\ R_{xy} = \sum_{i=1}^{n} x_i y_i \tag{5}$$

To distinguish between (m, q) or (s, t) parameters, N_1 and N_2 are defined as follows:

$$N_1 = R_{xx}n - R_x^2,\ N_2 = R_{yy}n - R_y^2. \tag{6}$$

N_1 and N_2 represent the width of the cluster of regression points along the X- and Y-axis. If N_1 is larger than N_2, the cluster of points are more horizontal than vertical, which makes the regression y to x ($y = mx + q$) favourable. If N_1 is smaller than N_2 the regression of x to y ($x = sy + t$) is selected.

With $T = R_{xy}n - R_xR_y$, the parameters (m, q) or (s, t) are given by following equations:

$$m = \frac{T}{N_1};\ q = \frac{(R_y - mR_x)}{n};$$
$$s = \frac{T}{N_2};\ t = \frac{(R_x - sR_y)}{n} \tag{7}$$

Thus we can get the slope angle of line segment with $\theta = \tan^{-1}(m)$ or $\theta = \tan^{-1}(1/s)$ $(s \neq 0)$. The length of the line segment can be given by the following equation:

$$L = \sqrt{(x_n - x_1)^2 + (y_n - y_1)^2} \tag{8}$$

where (x_1, y_1) and (x_n, y_n) represent two end points of line segment respectively.

The coordinates of the central point (x_m, y_m) are calculated by these expressions:

$$x_m = \frac{R_x}{n},\ y_m = \frac{R_y}{n}. \tag{9}$$

The coordinates of left bottom corner and right top corner of the enclosing rectangle are represented by (x_{lb}, y_{lb}) and (x_{rt}, y_{rt}), where:

$$\begin{aligned} x_{\text{lb}} &= \min(x_i),\ x_i \in \{x_1, x_2 \cdots x_n\}; \\ y_{\text{lb}} &= \min(y_i),\ y_i \in \{y_1, y_2 \cdots y_n\}; \\ x_{\text{rt}} &= \max(x_i),\ x_i \in \{x_1, x_2 \cdots x_n\}; \\ y_{\text{rt}} &= \max(y_i),\ y_i \in \{y_1, y_2 \cdots y_n\}. \end{aligned} \tag{10}$$

If only one measured point exists in the cluster, the slope angle and the length of line segment are both zero, the coordinates of central point are equal to the coordinates of the measured point.

3. Obstacle merging

After obstacle extracting, lots of clusters representing either real obstacles or brushwood on roadsides are shown together in the obstacle map. We must merge the points produced by the brushwood in order to reform the roadsides. The process is as the following:

Two neighboring clusters are checked for merging by using the following criteria:

a. The difference of slope angle between obstacles must smaller than a threshold;

b. The range from the central point of one obstacle to the line segment of another obstacle must be smaller than a given threshold.

If at least one obstacle's length is zero, criteria (a) is skipped.

Once two obstacles are merged, the feature extraction module is carried out again to obtain the feature of new obstacle in order to implement the merging process successively.

After this step, the number of clusters will be greatly reduced.

4. Distinguishing obstacles from brushwood

We must distinguish the real obstacles on the road from brushwood on the roadsides in order to provide enough information for the ALV's navigation. The following laws are used to distinguish them:

a. Clusters longer than 4 meters and slope angle larger than 75 degree are considered as road edges;

b. Shorter than 2 meters clusters whose range from each side of rectangle to roadsides is more than 0.5 meters are considered as obstacles;

c. If the rest of the clusters look like roadsides, do obstacle merging again.

Only two roadsides besides real obstacles are given after this step.

FUSION OF OBSTACLE INFORMATION

Strong noises due to the bouncing of the bodywork as the ALV runs and the different reflectivity of the environment will affect the obstacles' positioning in the obstacle map and even cause omission of obstacles sometimes. To minimize effect of noise, Kalman filter technique is used to trace the obstacles in realtime. There are two steps in the process: correspondence and estimation. After the precise position is obtained, data provided by the top LRF are collated and used to compute the height of the obstacles.

1. Correspondence of obstacles

Before Kalman filtering is carried out, the correspondence between obstacles from the current map and fused map should be determined. A circular region can be formed around the position prediction produced by the Kalman filter last time. The region represents the area in which obstacles will possibly appear at this time. Only the points inside the region are checked for correspondence.

The "least difference" rule is used to determine which one matches the predicted obstacle best when multiple obstacles have to be checked for correspondence. The difference between two obstacles is defined by the following function:

$$\begin{aligned} &\text{Difference}(\text{obs}(A), \text{obs}(B)) = \\ &\quad W_1 \Delta D + W_2 \Delta\alpha + W_3 \Delta L \end{aligned} \tag{11}$$

where:

$\text{obs}(A)$ and $\text{obs}(B)$ are the two obstacle to be checked;

ΔD is the difference of coordinates between the central position of obstacles;

$\Delta\alpha$ is the difference between the slope angles of the obstacles;

ΔL is the difference between the lengths of the obstacles;

W_1, W_2, W_3 are the weights of each item.

Among the three weights, because $\Delta\alpha$ often changes acutely, W_2 is set the smallest one and W_1 is the largest. In our application, $W_1 = 0.8$,

$W_2 = 0.05$, $W_3 = 0.15$.

Only the obstacle with the least difference from the predicted one is considered to have correspondence with the predicted obstacle.

2. Position estimation

As above mentioned, the jounce of ALV may add random noise producing error into the detection result. The noise can be considered as Gaussian white noise, against which a Kalman filter can suitably be used because of its excellent performance in signal estimation in additive noise environment, and ability to produce a prediction value for the next time.

Let (x_n, y_n) be the coordinates of the central point of the obstacle to be estimated, $\boldsymbol{Z}_n = [x_n\ y_n]^{\mathrm{T}}$ be the corresponding state vector. Assuming ALV runs in a straight line and at constant speed, we have:

$$\begin{cases} \boldsymbol{Z}_k = \boldsymbol{Z}_{k-1} + h\dot{\boldsymbol{Z}}_{k-1} \\ \dot{\boldsymbol{Z}}_k = \dot{\boldsymbol{Z}}_{k-1} = (\boldsymbol{Z}_{k-1} - \boldsymbol{Z}_{k-2})/h \\ \boldsymbol{Y}_k = \boldsymbol{Z}_k + V_k \end{cases} \quad (12)$$

Here h is sampling time, Y_k the observing vector and V_k the observed Gaussian noise.

Let $\boldsymbol{X}_k = [\boldsymbol{Z}_k \quad \boldsymbol{Z}_{k-1}]^{\mathrm{T}}$, rounding up of the above set of equations yields:

$$\begin{cases} \boldsymbol{X}_k = A\boldsymbol{X}_{K-1} \\ \boldsymbol{Y}_K = H\boldsymbol{X}_K + \boldsymbol{V}_K \end{cases} \quad (13)$$

Where:

$$A = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}, \quad H = [1 \quad 0].$$

Finally, we have the following Kalman filtering formula:

$$\begin{cases} K_K = \dfrac{AP_{K-1}A^{\mathrm{T}}H^{\mathrm{T}}}{HAP_{K-1}H^{\mathrm{T}} + R} \\ \hat{X}_K = A\hat{X}_{K-1} + K_K(Y_K - HA\hat{X}_{K-1}) \\ P_K = (I - K_K H)AP_{K-1}A^{\mathrm{T}} \end{cases} \quad (14)$$

Here R is the covariance of V_k, K_K the Kalman gain, $\hat{X}_K$ the filter's output and P_K the error covariance.

3. The start and the end of tracing

As above mentioned, the two LRFs have different functions in the obstacle's detection and tracing process. In general, obstacles are always discovered by the top LRF earlier because of its higher resolution. After several times of correspondence determination and updating, the new obstacle is confirmed. Then a new trace record is produced and stored in the memory. After that, the bottom LRF is used to continue the tracing of the obstacle.

For obstacles already stored in the memory of the system, the prediction value of last time is adopted if no correspondence can be found in the current map. But the trace should be deleted if the corresponding obstacle cannot be found in a considerably long time because this obviously means the obstacle has disappeared or gone out of the LRF's effective region.

4. Height and width of obstacle

The obstacle width is defined as the length of the X-axis of its enclosing rectangle and can be directly obtained from the obstacle extracting section. Since the position estimation is obtained by Kalman filtering, we can use Eq. (1) to compute the height of the obstacle.

EXPERIMENTS

1. Experimental environment

Fig.3 shows an about 5 m wide road lined on both sides by about 1.5 m high brushwood, and two spaced about 17 m apart artificial obstacles, with one placed beside the road, the other placed near the middle of the road.

Fig.3 The experimental environment

2. Processing result of current sample

Fig.4 (a) shows one sample of bottom LRF scan where black points in the figure represent the obstacles or brushwood, and the 8 m width and 30 m depth rectangular region represents the preset effective area of the LRF. (b) shows the result of obstacle extraction. From the result we can see that the map contains too many clusters

which will lead to great difficulty in the following path planning stage. Thus we employ the obstacle merging process. (c) shows the result produced by two successive rounds of obstacle merging. From the map we can see the distribution of brushwood and the obstacle clearly. In practice, we directly provide the feature of the road edges to the path planner, while a good estimation of obstacle's position has to be made by the following stage - Kalman filtering.

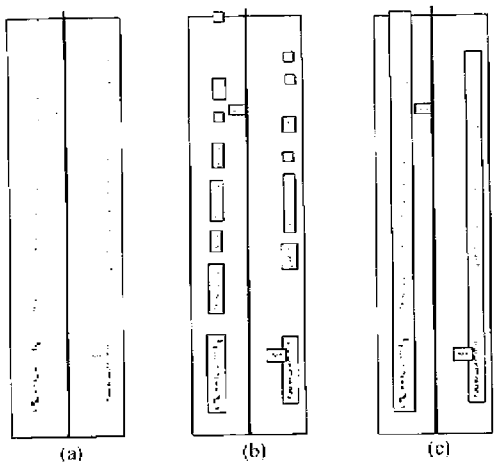

Fig.4 The processing result of current sample
(a) one sample
(b) result of obstacle extracting
(c) result of obstacle merging

3. Experiment result of Kalman filtering

To verify the performance of Kalman filtering, experiment was designed. The ALV was driven at 40 km/h straight towards the obstacles and at the same time, the range data was obtained by the bottom LRF at sampling time of 40 ms. There were a total of about 65 samples over the test distance and the difference of obstacle's central position in Y direction between successive samples is about 44 cm. Let $(\Delta x, \Delta y)$ be the difference of the central position of the obstacle in the successive samples, and (e_x, e_y) be the position errors given by the following expression:

$$e_y = \Delta y - 44, \quad e_x = \Delta x \qquad (15)$$

Smaller (e_x, e_y) leads to preciser pinpointing of the obstacle. The position errors before and after Kalman filtering are shown in Fig. 5 where (a) and (b) represent the Y and X position errors respectively.

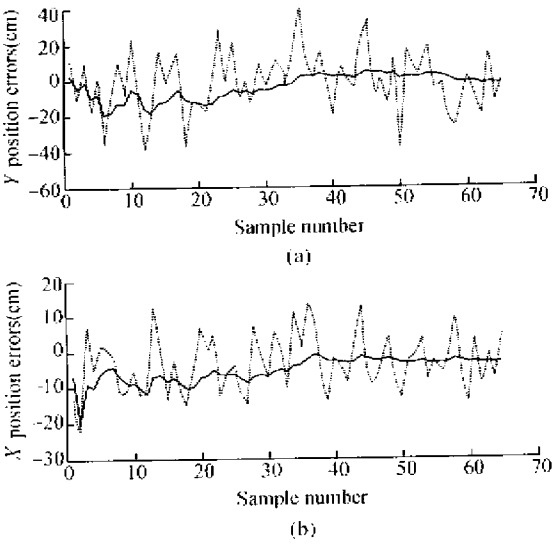

Fig.5 The position error before and after Kalman filtering
(a) Y position errors before and after Kalman filtering
(b) X position errors before and after Kalman filtering

In Fig.5, the dashed line and the real line represent the position error before and after Kalman filtering respectively. The result clearly shows that, as expected, the position error is very small after filtering and becomes smaller when the number of samples increases.

It should be pointed out that, though the experiment was carried out under the condition of constant velocity, the method is also feasible when the speed of the ALV does not change rapidly because the sampling time of the LRF is very short comparing to the variation of ALV's speed. Based on this, the running of ALV can be divided into several segments each of which can be seen as straight motion with constant speed. Though this approximation will increase the estimation error slightly, it will still be much better than the raw value.

CONCLUSIONS

This paper proposes a set of effective algorithms for obstacle detection by ALV using two 2D LRFs. Due to the complex non-structural outdoor environment, we use two 2D LRFs to obtain comprehensive information. After the processing of the current sample, rough information

of obstacles and roadsides can be obtained separately. Then Kalman filtering technique is used to gain better estimation of the obstacle's position. The width and the height of the obstacle can be obtained by collating data provided by the two LRFs. The experimental results showed our success.

References

Bonlanger, P., 1992. Range Image Segmentation, Free Space Determination, and Position Estimation for a Mobile Vehicle. Proceedings of SPIE Vol. 1831: Mobile Robots VII, 444 – 455.

Dunlay, R. Terry, 1986. Obstacle Avoidance on Aoadways using Range Data. SPIE vol.727: Mobil Robots, p.110 – 116.

Gonzales, J., Stentz, A., Ollero, A., 1992. An Iconic Position Estimator for a 2D Laser Range Finder. Proceedings of the IEEE International Conference on Robotics and Automation, p.2646 – 2651.

Gonzalez, J., Ollero, A., Reina, A., 1994. Map Building for a Mobile Robot equipped with a 2D Laser Rangefinder. Proceedings of *the IEEE International Conference on Robotics and Automation*, p.1904 – 1909.

Herbert, M., Kanade T., 1986. Outdoor Scene Analysis using Range data. Proceedings of the IEEE International Conference on Robotics and Automation, p. 1426 – 1432.

Kwon, D. Young, Lee, S. Jin, 1997. A Stochastic Environment Modelling Method for Mobile Robot by using 2-D Laser Scanner. Proceedings of the IEEE International Conference on Robotics and Automation, p. 1688 – 1693.

Mashavan, R., Dissanayake, G., Durrant-Whyte, H., 1998. Map-Building and Map-Based Localization in an Underground-mine by Statistical Pattern Matching. Proceedings of the IEEE International Conference on Robotics and Automation, p.1744 – 1746.

Vandorpe, J., Van Brussel, H., Xu, H., 1996. Exact Dynamic Map Building for a Mobile Robot Using Geometrical Primitives Produced by a 2D Range Finder. Proceedings of the IEEE International Conference on Robotics and Automation, p.901 – 908.

Ying Bishan, Liu Jilin, Gu Weikang, 1998. Algorithm and Parallel Implementation of 3D Vision Understanding in ALV, *Journal of Zhejiang University (Natural Science)*, **33**(3): 305 – 311.

Zhang Qi, Liu Jilin, Guo Xiaojun et al., 1997. Application of 3D Laser Range Finder: Problems and Solution, *Journal of Zhejiang University (Natural Science)*, **32**(6): 732 – 738.