Feature Extraction of Line-Scan LIDAR for Calibrating Camera and LIDAR

Pakapoj Tulsuk
Faculty of Electrical Engineer
Kasetsart University
Bangkok, Thailand
Email: tang_900@hotmail.com

Panu Srestasathiern
Geo-Informatics and Space
Technology Development Agency
Bangkok, Thailand
Email: panu.sre@gmail.com

Miti Ruchanurucks
Faculty of Electrical Engineer
Kasetsart University
Bangkok, Thailand
Email: fengmtr@ku.ac.th

***Abstract*—This paper presents a feature extraction procedure of 2D-scan Lidar's data. The procedure is a detection of a line and its correspondent points. Detected features are used to calibrate the Lidar and a camera for their orientation and position. We applied Hough transform and RANSAC to estimate the line and its correspondent points. And we also applied Zhang and Pless's algorithm [1] to calibrate their orientation and position.**

I. Introduction

3D mapping systems are popular in recent years. There are many sensors that provide an 3D environment such as Sonar, Laser or Microsoft Kinect. Those sensors mostly present environment with point clouds. The point clouds can represent 3D scenes but they are lack of scene's texture. Therefore we provide color into the scene to increase point clouds texture. Integrating texture into point clouds gives a sufficient understanding of 3D map

Lidar is depth measurement instrument using laser's time of flight. There are 2 major types of Lidar. The first one scans the environment's depth, reproducing a set of point cloud that represents a scene. THough its cost is very high, it gives rich data; however, its operation range is quite limited compared to the second. The other scans the surrounding perimeter depth on a single plane or more. Its cost is lower than the first one. Also as its point cloud data is sparse; however the operational range is farther.

Calibration between the camera and Lidar relatively explain by position and orientation of both devices. Point clouds can be colorized by overlaying Lidar's points to image. The colorizing process need the position and orientation which can be obtained by calibrate camera and Lidar Calibration require a set of calibration pattern that have uniquely geometry shape. The unique calibration pattern yields a simple co-feature detection such as points on the plane. Calibration object is commonly a chessboard because they can be easily defined by their mathematical model. There are 2 calibration steps to find the orientation and position of Lidar and camera, calibrate camera to world coordinate (chessboard) first and then calibrate Lidar to camera coordinate.

Camera model can be explained by intrinsic parameters and extrinsic parameters. Intrinsic parameters can be represented in 3×3 matrix containing camera's characteristic such as camera's focal length and center of image. Extrinsic parameters can also be represented in 3×3 matrix containing rotation matrix and translation matrix which show the orientation and position of the camera in world coordinate respectively. For calibration of Lidar, it requires only estimation of extrinsic parameters that present Lidar's orientation and position in the camera coordinate. In summary, we can represent both devices in any coordinate system using those mentioned transformation. Therefore we can overlay data from Lidar to that of camera.

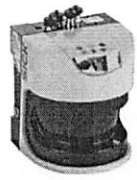


Fig. 1: Lidar SICK LMS511

In this paper, we will give a simple procedure to accommodate the calibration process by extracting the co-feature points on Lidar's data that correspond to the features on the camera automatically. In other words, we will detect a checkerboard observed by Lidar to used the detected information for calibrate camera and Lidar. We selected SICK LMS511 (Lidar), please see figure 1, to be the depth scanner.

II. Literature

Our literatures contain methods to calibrate 2D-scan Lidar and a camera, 3D-scan Lidar and a camera, and 2D-scan Lidar characteristic. To compare and to illustrate the feature points used by various methods.

Vasconcelos present an estimation method in classical solution, pNp manner [2]. Furthermore he show the minimum case of his method. Mainly he point out that his method is effective and yield a beeter result than Zhang and Pless in minimum case situation. He also applied his algorithm in practical experiment and simulation.

Zhang and Pless calibrated Lidar and a camera using checkerboard as a calibration pattern. They created a constraint of plane's normal for every point that is observed by Lidar [1]. Moreover, They evaluated their result of rotation and translation matrices using a simulation.

Kwak also proposed a method for calibrating 2D-scan Lidar and a camera by a v-shape board as a calibration pattern.

He set up a configuration that the board is visible to both devices. He uses a peak of v-shape and 2 ending edges of the board as features [3] to estimate orientation and position of camera and Lidar.

Wasielewski used black and white v-shape board to detect a corner of the board and use it as a co-feature to fine the pose of camera and Lidar. He was the first person that proposes of using 1 peak of v-shape board and its 2 ending edges [4] for calibration.

Li calibrate for Lidar and camera's orientation and position using line features with a specific checkerboard such as an arrow shape. Besides he applied stereo vision to refine their geometry [5].

Lipu Zhou calibrated 3D-scan Lidar (Velodyne HDL-64E S2) and a camera in 2 different ways, using a checkerboard as a calibration pattern. First is minimizing calibration pattern normal and the points lied on the checkerboard. Second is minimizing edges of checkerboard, left and right end of the checkerboard boarder, with the scanned points [6] [7]. And both method yield for for rotation and translation matrix with fine result.

Ye described the characteristic of 2D-scan Lidar. There are mixed pixels that are points came from averaging 2 neighbouring point, causing data false measurement [8]. The mixed pixels should be removed after our procedure.

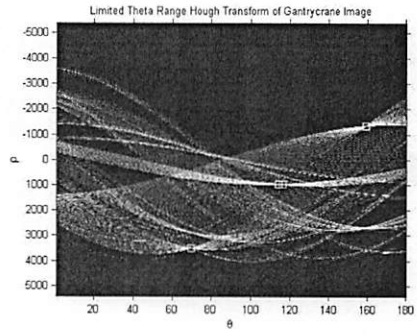
We review various of calibration method that using differential co-features, and observe that their calibration pattern is a plane or two combined planes. 2D-scan Lidar observes those calibration pattern like a line segment or two connected line segments respectively. We also noticed that calibration methods usually involve with line or its ending point. As a result, we come up with line and its correspondent points feature extraction to accommodate the process.

III. Feature Detection

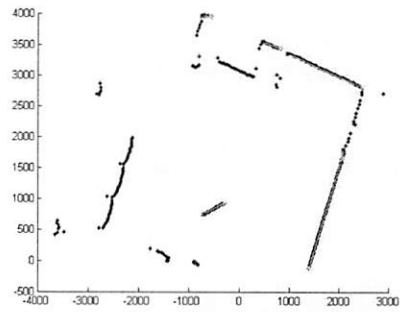
Lidar, SICK LMS511, scans an environment and represents it in 2 dimensional space. Which means the environment is observed by a scan-plane of Lidar. This section concern a detection of line segment in Lidar data which is an intersection between Lidar's scan-plane and checkerboard's finite plane.

We set up an experiment by letting Lidar and camera observed the checkerboard. Lidar observes the checkerboard as a line segment. However there might be other parts of background detected by Lidar. Since we set the calibration plane to be the nearest plane that Lidar can observe. Therefore it's easy to detect the nearest line segment using Hough transform and RANSAC.

We applied Hough Transform to locate all points that can be formed into a line segment. And we choose to detect the nearest line segment using a distance of the center of the line segment to the origin in Lidar coordinate. If there are some ambiguity point among the estimated points, RANSAC will remove those points using a iterative probabilistic method. After the data went through all the processes, the data had a high probability to be the points that actually lie on the plane.



(a) White squares are top 5 highest score of line parameters that indicate 5 line in scanned data.



(b) Detected lines using Hough transform. Green lines are the detected lines, red dot and yellow dot indicate left and right ending of line segment.

Fig. 2: A sample of Hough transform with five detected lines using Lidar data set

A. Hough Transform

Hough Transform is an algorithm to detect simple shape such as line, circle or eclipse. In this case we will detect a line. The method is vary θ for each point(x_i, y_i), solve for ρ and keep counting the repeated parameter. Any point that fits to the parameter model, that have the highest count, is most likely to be a member of a line segment. It's known as transformation of x-y space to ρ-θ parameter space. Additionally, parameter space is a space that indicate line or desire shape parameter. Line(l) can expressed in a form of equation as $y_i = kx_i + d$. Moreover, it can also represented as $\rho(\theta) = x_i \cos(\theta) + x_i \sin(\theta)$.

After doing the above procedures, we obtains five candidates that most likely to be a line, please see figure 2. Consequently, we set the checkerboard to be the nearest planar object. And we add an infrared camera, please see figure 3, into our experiment in order to confirm that calibration pattern is scanned by Lidar. We choose the nearest one by measuring a distance form the origin to the center of each line. Please see figure 5.

The algorithm below show the Hough transform used for line detection. HoughLines is a basic algorithm describe the mentioned Hough transform [9]. IsPntInLine uses for evaluate

whether a point is in the line or not. DetectClosestLine is the main procedure to detect a closest line

Algorithm III.1: HOUGHLINES(P, N_θ, N_r, K)

$$r_{max} \leftarrow \left\| \begin{matrix} \underset{m}{argmax}|x| \\ \underset{m}{argmax}|y| \end{matrix} \right\|_2$$

$\Delta T \leftarrow \frac{180}{N_\theta}$
$\Delta R \leftarrow \frac{2 \cdot r_{max}}{N_r}$
accumulator array Acc(i_θ,i_r) of size $N_\theta \times N_r$

for each accumulator cells (i_θ,i_r)
  do $Acc(\theta_i, r_i) \leftarrow 0$
for $i \leftarrow 1$ **to** m
  do { **for** $i_\theta \leftarrow 1$ **to** N_θ
    do { $\theta \leftarrow \Delta T \cdot i_\theta$
      $r \leftarrow x_i \cdot \cos\theta + y_i \cdot \sin\theta$
      $i_r \leftarrow \frac{N_r}{2} + round\left(\frac{r}{\Delta R}\right)$
      $Acc(i_\theta, i_r) \leftarrow Acc(i_\theta, i_r) + 1$

FindPeak returns K highest of i_θ,i_r
peaks is $[i_\theta \quad i_r]^T$
$Peaks \leftarrow FindPeak(Acc, K)$

return ($Peaks$)

Algorithm III.2: ISPNTINLINE($P, Peaks, Thres$)

for $i \leftarrow 1$ **to** m
  do { $error \leftarrow x \cdot \cos\theta + x \cdot \sin\theta - r$
    if $|error| < Threshold$
      then $Indicator \leftarrow 1$
      else if $Indicator \leftarrow 0$
return ($Indicator$)

Algorithm III.3: DETECTCLOSESTLINE(P, N_θ, N_r)

$Peaks \leftarrow HoughLines(P, N_\theta, N_r, K)$
for $i \leftarrow 1$ **to** K
  do { $In \leftarrow IsPntInLine(P, Peaks, Thres)$
    $POnLine(i) \leftarrow$ every $P(In = 1)$
    $Middle(i) \leftarrow MiddleOfLine(POnLine)$

$$i_{closest} \leftarrow \underset{K}{argmin} \left\| Middle - \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\|_2$$

return ($POnLine(i_{closest})$)

B. RANSAC

RANSAC is an abbreviation for Random Sample Consensus. It is a random and iterative method uses to estimate all inlier and their co-response model such as Planes fitting, essential matrix estimation or geometry model matching. We apply RANSAC to estimate and verify whether any detected point from Hough transform is a member of the line or not; because there are some situation where Hough transform detects a false data. First is a improper experimental setup, the closest object that we assume to be checkerboard, is not the board. Second is mixed pixels [8], an approximated datum by the device itself, that occurred and can not distinguish by Hough transform. Please see figure 6. RANSAC MATLAB library is from Peter Kovesi [10] which is popularly used. There are pseudo-code below to explain about RANSAC.

RANSAC randomly uniformly picks a set of data, solves for their line model and applies the model with the rest of points. After that evaluate sum of orthogonal distance from line to each point, called distance function. Then repeat the process again until it reaches the minimum distance. Such model that yields the minimum distance means the majority set of points is fitted with this model and the rest is not. Therefore we can distinguish between the points that are member of the line, and those who are not by thresholding. RANSAC is robust to noisy data as mentioned in previous work's discussion. So it's useful since we will apply this algorithm in our practical experiment.

Algorithm III.4: RANSAC(P, $fittingfn(\cdot)$, $distfn(\cdot), degenfn(\cdot), sample, thres$)

$N \leftarrow$ number of trial
$N_{pts} \leftarrow$ number of points(P)
$trial \leftarrow 0$
$degenerate \leftarrow 1$
$score_{best} \leftarrow 0$
while $N > trial$
  do { **while** $degenerate = 1$
    do { $i \leftarrow randsample(N_{pts}, sample)$
      $degenerate \leftarrow degenfn(P(i))$
      if $degenerate = 0$
        then $model \leftarrow fittingfn(P(i))$
    $(in, model) \leftarrow distfn(model, P, thres)$
    $N_{inliers} \leftarrow$ number of in
    if $N_{in} > score_{best}$
      then { $score_{best} \leftarrow N_{in}$
        $inliers_{best} \leftarrow in$
        $model_{best} \leftarrow model$
return ($model_{best}, in_{best}$)

IV. LIDAR AND CAMERA CALIBRATION FOR EXTRINSIC PARAMETER

Lidar and camera calibration is a method to estimate rotation and translation matrices of those devices by solving a mathematical equation. Firstly, we would like to define a pin hole camera model and experimental geometry and their equations. Next, we going to apply Zhang and Pless's algorithm [1] to calibrate between Lidar and camera. Finally, we will discuss their solution, that can reconstruct the extrinsic parameters. We also redefine variable and add some simply solution to re-scale the estimated rotation matrix to be the real one.

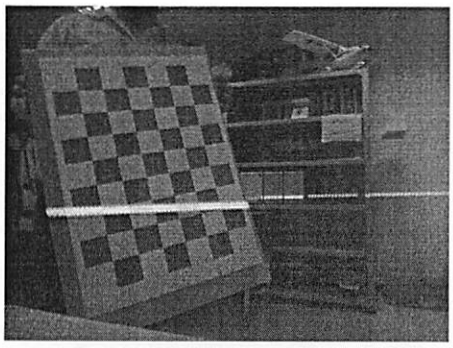


Fig. 3: Image captured by infrared camera.

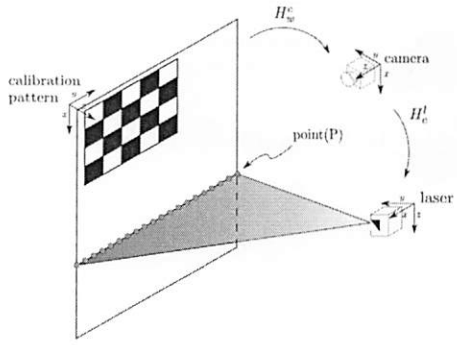


Fig. 4: Experiment set up.

A. Notation

Pin hole camera model including intrinsic parameters(K) and extrinsic parameters(R and T). Intrinsic parameters define camera characteristic such as focal length, center of image and its skewness. Extrinsic parameters comprise of rotation matrix(R) and translation matrix(T) which is orientation and position between camera coordinate and world coordinate. In this scenario, world coordinate system is a coordinate of checker broad. We using MatLab toolbox from Jean-Yves Bouguet [11] to calibrate camera's intrinsic and extrinsic parameters.

We define variables of 3D point(P) and its transformation as follow. P is $\begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$. P^{img} is a point in image plane. P_3^{img} is a z component of P^{img}. P^c is a point in camera coordinate. P^w is a point in world coordinate. R_w^c is a rotation matrix that rotate from world coordinate to camera coordinate. And T_{wc}^c is a translation matrix that translation from world coordinate to camera coordinate.

$$
\begin{aligned}
P^{img} &= \frac{1}{P_3^{img}} K P^c \\
P^c &= \begin{bmatrix} R_w^c & T_{wc}^c \\ 0 & 1 \end{bmatrix} P^w \\
P^c &= H_l^c P^w \\
H_l^c &= \begin{bmatrix} R_w^c & T_{wc}^c \\ 0 & 1 \end{bmatrix}
\end{aligned}
$$

We can also apply rigid transformation to transform camera to Lidar coordinate as well by given similar variables to above. P^l is a point in Lidar coordinate. R_c^l is a rotation matrix that rotate from camera coordinate to Lidar coordinate. And T_{cl}^l is a translation matrix that translates from camera coordinate to Lidar coordinate.

$$
\begin{aligned}
P^l &= \begin{bmatrix} R_c^l & T_{cl}^l \\ 0 & 1 \end{bmatrix} P^c \\
P^l &= H_c^l P^c \\
H_l^c &= \begin{bmatrix} R_c^l & T_{cl}^l \\ 0 & 1 \end{bmatrix}
\end{aligned}
$$

B. Geometry Constraint

Checker broad is a finite plane and plane has its normal to define itself in 3D coordinate. Zhang and Pless given the vector $N = -R_3(R_3^T T)$, N that parallel to the normal and its magnitude equal to a distance from camera to calibration plane to explain infinite plane. Geometry constraint on rigid transformation between camera and Lidar is show as follow. In camera coordinate, they observed the points on the plane since every point lie on the calibration pattern. Then they gave $N \cdot \widehat{P}^c = ||N||^2$ since $\widehat{P}^c$ is $\begin{bmatrix} x & y & z \end{bmatrix}^T$.

$$
||N||^2 = N \cdot {R_c^l}^{-1}(P^l - T_{cl}^l)
$$

C. Solving Extrinsic Parameters

Lidar SICK LMS511 is a 2D scanner. We defined its coordinate system to represent a point(P^l) to be $\begin{bmatrix} x & y & z & 1 \end{bmatrix}^T$. Since it is 2D scanner, they set y to be zero then they could reduce P^l to $\widehat{P}^l$.

$$
\begin{aligned}
||N||^2 &= N \cdot \widehat{H}_l^{c-1} \widehat{P}^l && (1) \\
\widehat{H}_l^{c-1} &= {R_c^l}^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 0 & -T_{cl}^l \\ 0 & 1 \end{bmatrix} && (2) \\
\widehat{P}^l &= \begin{bmatrix} x & z & 1 \end{bmatrix}^T && (3)
\end{aligned}
$$

After solving equation (1) using Least Square, they obtained $\widehat{H}_l^{c-1}$ 3×3 matrix. They recovered R_c^l and T_{cl}^l by rearranging parameters in $\widehat{H}_l^{c-1}$ using orthonormal property of rotation matrix as follow.

$$
\begin{aligned}
R_c^l &= \begin{bmatrix} \widehat{H}_{l1}^{c-1} & -\widehat{H}_{l1}^{c-1} \times \widehat{H}_{l2}^{c-1} & \widehat{H}_{l2}^{c-1} \end{bmatrix}^T \\
T_{cl}^l &= -\begin{bmatrix} \widehat{H}_{l1}^{c-1} & -\widehat{H}_{l1}^{c-1} \times \widehat{H}_{l2}^{c-1} & \widehat{H}_{l2}^{c-1} \end{bmatrix}^T \widehat{H}_{l3}^{c-1}
\end{aligned}
$$

Estimated R_c^l is not meet $SO(3)$. So we decompose it using svd. And recompose them to the new rotation matrix as follow. $R_c^l = u(diag\begin{bmatrix} 1 & 1 & det(u)det(v^T) \end{bmatrix})v^T$

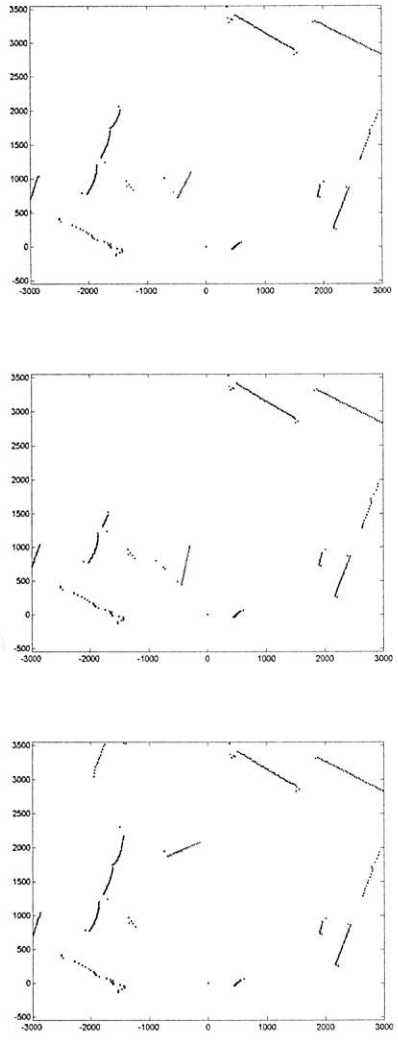


Fig. 5: A sample of detected checkeboards which is the nearest line. The set of green points show a set of point that lie on the calibration pattern.

V. Result

Line model and their co-response points are extracted from Lidar data using Hough transform and RANSAC. Line model(l) is represented in matrix as $l = [\ a \quad b \quad c\ ]^T$ and also satisfies $0 = ax + by + c$. We use the co-response points to calibrate for extrinsic parameter between Lidar and camera using Zhang and Pless's constraint [1]. We re-project Lidar scanned points to image plane to present our result.

Using Hough transform, we have the co-response points to the line which is the nearest one to the origin in Lidar coordinate. Please see figure 5. We can get rid of the mixed pixels [8] and improper data,cause from bad experimental setup by applying RANSAC. We also get line model and a set of point that is a member of the line (inlier). Please see figure 6. After obtaining the inlier points that lie on the calibration pattern, we calibrate for extrinsic parameter using [1] algorithm. And re-project every point into the image to overlay the point cloud with its correspondent color. Please see figure 7.

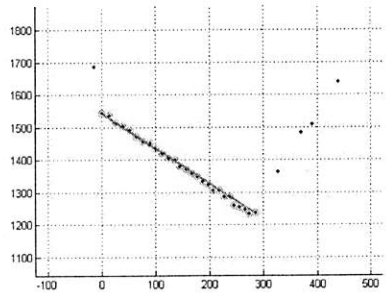


Fig. 6: Points(black dot with green circle) is the point that fit with line model(pink line) and those points(only black dot) are mixed pixel that should not include for calibration.

Fig. 7: Overlay point cloud(blue cross sign) on the image using Zhang and Pless's [1] algorithm.

VI. Conclusion

In this paper, we suggest a simple way to automatically detect features from Lidar's data that will be applied for calibration process later on. We also illustrate the result of the modified calibration process that used our detected feature for calculation. Also we find that the line segment is the intersection between two planes, checkerboard (calibration pattern)'s finite plane and Lidar (SICK LMS511)'s scan plane.

We would like to propose a new calibration method that yield more accurate value of orientation and position between Lidar and camera using the mentioned knowledge as a future work.

Acknowledgment

The authors would like to thank Geo-Informatics and Space Technology Development Agency (Public Organization) who provided Lidars SICK LMS511 and accommodate the laboratory. This research is financially supported by Thailand Advanced Institute of Science and Technology (TAIST), National Science and Technology Development Agency (NSTDA),

Tokyo Institute of Technology, Kasetsart University (KU) under the TAIST Tokyo Tech Program.

REFERENCES

[1] Q. Zhang and R. Pless, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, 2004, pp. 2301–2306 vol.3.

[2] F. Vasconcelos, J. Barreto, and U. Nunes, "A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2097–2107, 2012.

[3] K. Kwak, D. Huber, H. Badino, and T. Kanade, "Extrinsic calibration of a single line scanning lidar and a camera," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011, pp. 3283–3289.

[4] S. Wasielewski and O. Strauss, "Calibration of a multi-sensor system laser rangefinder/camera," in *Intelligent Vehicles '95 Symposium., Proceedings of the*, 1995, pp. 472–477.

[5] G. Li, Y. Liu, L. Dong, X. Cai, and D. Zhou, "An algorithm for extrinsic parameters calibration of a camera and a laser range finder using line features," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 2007, pp. 3854–3859.

[6] L. Zhou and Z. Deng, "A new algorithm for computing the projection matrix between a lidar and a camera based on line correspondences," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2012 4th International Congress on*, 2012, pp. 436–441.

[7] ——, "Extrinsic calibration of a camera and a lidar based on decoupling the rotation from the translation," in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, 2012, pp. 642–648.

[8] C. Ye and J. Borenstein, "Characterization of a 2d laser scanner for mobile robot obstacle negotiation," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 3, 2002, pp. 2512–2518.

[9] W. Burger and M. J. Burge, *Principles of Digital Image Processing, Core Algorithms*. Springer-Verlag London Limited, 2009.

[10] P. Kovesi. (2013) Ransac matlab library. [Online]. Available: http://www.peterkovesi.com/

[11] J. sYves Bouguet. (2013) Camera calibration matlab toolbox. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/