

A Simple Sample Consensus Algorithm to Find Multiple Models

Carlos Lara-Alvarez, Leonardo Romero,
Juan F. Flores, and Cuauhtemoc Gomez

Division de Estudios de Posgrado
Facultad de Ingenieria Electrica
Universidad Michoacana
Morelia, Mexico

{carlosl,lromero,juanf}@umich.mx, temo@michoacan.gob.mx

Abstract. In many applications it is necessary to describe some experimental data with one or more geometric models. A naive approach to find multiple models consists on the sequential application of a robust regression estimator, such as RANSAC [2], and removing inliers each time that a model instance was detected. The quality of the final result in the sequential approach depends strongly on the order on which the models were. The MuSAC method proposed in this paper discovers several models at the same time, based on the consensus of each model. To reduce bad correspondences between data points and geometric models, this paper also introduces a novel distance for laser range sensors. We use the MuSAC algorithm to find models from 2D range images on cluttered environments with promising results.

1 Introduction

In many applications it is necessary to describe some experimental data with multiple models. A common application in robotics and vision consists on finding models from images. For example, in man-made environments it is useful to discover a set of planes from a set of 3D laser scans. Several approaches to find multiple models have been reported in literature, Franck Dufrenois and Denis Hamad [1] divide those approaches into:

- *Methods that assume the existence of a dominant structure or model in the data set.* These methods successively apply a robust regression estimator. Each time that a model instance is detected, its inliers (data points represented by the model) are removed from the original data.
- *Methods that consider the presence of several structures.* These methods simultaneously extract multiple models from a single data set. Learning strategies range from region growing (or region merging) [3,4,7,6] to probabilistic methods such as Expectation–Maximization [8], or Markov Chain Monte Carlo Methods [5].

This paper introduces a simple algorithm to find multiple models from laser scans based on the well known Random Sample Consensus paradigm (RANSAC) [2]. Two similar approaches to the one presented in this paper are MultiRANSAC algorithm proposed by Zuliani and others and Sequential RANSAC (see [11] for more information on both approaches).

MultiRANSAC uses a binary cost function; consequently, it defines the set of data inliers L for a given model θ ; that is, $L(\theta) = \{z_i \mid d(z_i, \theta) < t\}$, where $d(\cdot, \cdot)$ is a distance function, and t is a given threshold. To find W models (W is a pre-defined number), the MultiRANSAC algorithm fuses the W random hypotheses generated at the i -th iteration with the best W models available at the moment to get a new set of W models. Every set of hypotheses $\{\theta_1, \dots, \theta_W\}$ generated in the MultiRANSAC approach has the property that their corresponding sets of data inliers are pairwise disjoint, $\forall i \neq j \ L(\theta_i) \cap L(\theta_j) = \{\}$.

The MuSAC algorithm proposed in this paper has some differences with the MultiRANSAC Algorithm: it does not need to know a priori the number of models, and it can allow those models to have a small amount of common data between them. As a side-effect contribution, a new distance measure is defined. The new distance takes into account how a typical laser range sensor takes measurements from the environment. The rest of the paper is organized as follows. Section 2 reviews the RANSAC method. Section 3 analyzes the Sequential RANSAC approach. Section 4 introduces a new metric used in our approach, called Directional Distance. Section 5 introduces the MuSAC algorithm. Section 6 shows some results of applying the proposed algorithm to solve the problem of obtaining multiple planar models from 2D range images. Finally, the conclusions can be found in Section 7.

2 The Random Sample Consensus Approach

The Random Sample Consensus Approach is very popular for fitting a single model to experimental data. In their seminal paper, Fischler and Bolles[2] describe that the RANSAC procedure is opposite to that of conventional smoothing techniques: rather than using as much of the data as possible to obtain an initial solution and then attempting to eliminate the invalid data points, RANSAC uses as the minimum initial data set as feasible and enlarges this set with consistent data, when possible.

The RANSAC strategy has been adopted by many researchers because it is simple and can manage a significant percentage of gross errors (outliers). The smallest number of data points required to uniquely define a given type of model is known as the minimal set (two points define a line, three points define a plane, etc). When using RANSAC, minimal sets are selected randomly; each set produces a hypothesis for the best model. These models are then measured by statistical criteria using all data.

Many efforts have been done to improve the performance of the RANSAC algorithm (replacing the cost function that defines inliers is a usual one). The RANSAC Algorithm optimizes the number of inliers, MSAC (M-Estimator Sample Consensus) [9] incorporates an M-Estimator. M-estimators reduce the effect

of outliers by replacing the binary cost function by a symmetric, positive-definite function with a unique minimum at zero. Several functions such as Huber, Cauchy, Tuckey have been proposed. MLeSaC (Maximum Likelihood Sampling Consensus) [10] evaluates the likelihood of the hypothesis, using a Gaussian distribution for inliers and a uniform distribution for outliers.

3 Sequential RANSAC

A naive approach to find multiple models consists of sequentially applying a robust regression estimator, such as RANSAC, and removing inliers each time that a model instance is detected. Figure 1 illustrates the Sequential RANSAC approach. In the first stage (Figure 1(a)), a dominant line is found. After removing the points represented by the first line, it searches for a second line (Figure 1(b)). This process is iterated until the best model does not fulfill a given requirement (usually when the number of points are less than a given threshold). This strategy has some drawbacks: the number of tries is usually large to guarantee that the dominant model can be found on each stage; and, some data points can be misclassified.

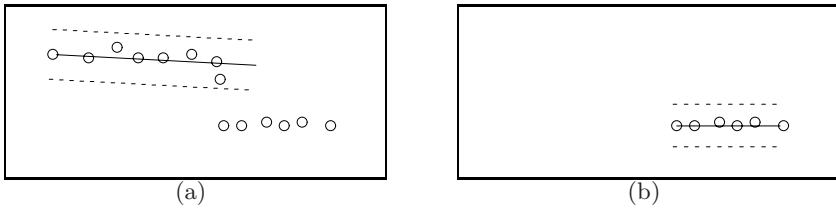


Fig. 1. Using the Sequential RANSAC approach to find multiple lines

4 Directional Distance

The orthogonal distance d_{\perp} is the preferred metric used to extract geometric models. To increment the rate of points correctly classified, a new metric is proposed. The directional distance d_{\nearrow} from the data point z_i to the hypothetical model θ_j is defined as

$$d_{\nearrow}(z_i, \theta_j) = d(z_i, \theta_j \wedge r_i) \tag{1}$$

where $d(\cdot, \cdot)$, is the Euclidean distance between two points; $\theta_j \wedge r_i$, is the intersection point between the model θ_j and the measurement ray r_i associated to z_i (assuming the sensor is at the origin of the reference system).

Suppose we want to evaluate the consensus of the line θ_j as shown in Figure 2. In this case, the observed point z_i (represented by the filled circle) does not correspond to points predicted by θ_j . The intersection point predicted by θ_j (denoted by $\theta_j \wedge r_i$) is represented by the empty circle. d_{\nearrow} is the distance between the point

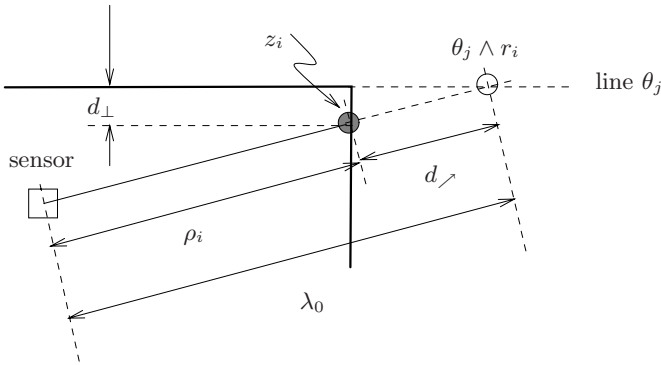


Fig. 2. Comparing the d_{\perp} distance and the d_{\nearrow} distance

Z_i and the point $\theta_j \wedge r_i$. Although d_{\perp} is the best metric for many fitting problems, d_{\nearrow} is better than d_{\perp} because it considers the measurement process of laser range sensors. The directional distance represents the real error given the ray r_i on which z_i was measured.

4.1 Point to Line Distance in 2D

A laser range finder takes measurements from the environment by emitting a ray in a given direction α . A ray with direction α represents the set of points (x_1, x_2) given by

$$x_1 = \lambda \cos \alpha; \quad x_2 = \lambda \sin \alpha; \quad \lambda \geq 0 \tag{2}$$

where λ , is the distance from the point (x_1, x_2) to the origin. Let $Ax_1 + Bx_2 + C = 0$ be the equation of a line (a simple model θ) with parameters $[A, B, C]$, and $z_i = (\rho_i, \alpha_i)$ a measured point in polar coordinates. Replacing Equation 2 into the line equation, and solving for the particular value λ_0

$$\lambda_0 = \frac{-C}{B \sin \alpha_i + A \cos \alpha_i} \tag{3}$$

If $\lambda_0 \geq 0$ then the intersection point exists and it defines the distance of the intersection point $\theta \wedge r_i$ to the origin. Finally, d_{\nearrow} is simply given by

$$d_{\nearrow}(z_i, \theta_j) = |\lambda_0 - \rho_i|. \tag{4}$$

4.2 Point to Plane Distance in 3D

Analogously to the 2D case, A 3D ray in its parametric form is

$$x_1 = \lambda \sin \alpha \cos \beta, \quad x_2 = \lambda \sin \alpha \sin \beta, \quad x_3 = \lambda \cos \alpha, \quad \lambda \geq 0 \tag{5}$$

where λ is the distance of the point (x_1, x_2, x_3) to the origin. The 3D plane equation is $A'x_1 + B'x_2 + C'x_3 + D' = 0$. Let $z_i = (\rho_i, \alpha_i, \beta_i)$ be a point measured at angles α_i, β_i then

$$\lambda_0 = \frac{-D'}{A' \sin \alpha_i \cos \beta_i + B' \sin \alpha_i \cos \beta_i + C' \cos \alpha_i}$$

and the point to plane distance is

$$d_{\nearrow}(z_i, \theta_j) = |\lambda_0 - \rho_i|. \tag{6}$$

5 The MuSAC Algorithm

The MuSAC Algorithm (Algorithm 1) iterates two phases: hypotheses generation, and selection. The hypotheses generation phase consists of discovering models from the data, and to quantify the relations between the models. The selection stage keeps the best models to the next iteration.

Algorithm 1. MuSAC(Z , the point set; m , a given number of models to be discovered at each iteration; τ , the minimum acceptable consensus)

```

1:  $n \leftarrow 0, \Theta = \{\}$ 
2: repeat
3:   for all  $i \in \{n + 1, \dots, m\}$  do ▷ Hypotheses generation
4:      $\Theta \leftarrow \Theta \cup \{\theta_i\}$ , where  $\theta_i$  is a random model from  $Z$ 
5:   end for
6:    $\forall i, j \in \{1, \dots, m\} c_{ij} \leftarrow \sum_{z \in Z} I(z, \theta_i) I(z, \theta_j)$ 
7:    $\Theta^* \leftarrow \{\}$  ▷ Selection
8:   repeat
9:     Select  $\theta_b \in \Theta$  such that  $\forall i \in \{1, \dots, m\} c_{bb} \geq c_{ii}$ 
10:    if  $c_{bb} \geq \tau$  then
11:       $\Theta^* \leftarrow \Theta^* \cup \{\theta_b\}$ 
12:      for all  $i \in \{1, \dots, m\}, i \neq b$  do
13:        if  $2c_{ib} > c_{ii}$  then
14:           $\forall j \in \{1, \dots, m\} c_{ij} \leftarrow 0$ 
15:        else
16:           $\forall j \in \{1, \dots, m\} c_{ij} \leftarrow c_{ij} - c_{bj}$ 
17:        end if
18:      end for
19:    end if
20:     $\forall i \in \{1, \dots, m\} c_{bi} \leftarrow 0$ 
21:    until  $\forall i \in \{1, \dots, m\} c_{ii} < \tau$ 
22:     $\Theta \leftarrow \Theta^*, n \leftarrow |\Theta^*|$ 
23: until convergence
24: return  $\Theta$ 

```

Lines 3 through 5 of Algorithm 1 generate a predefined number of hypotheses m . Because the probability that two points belong to the same model is higher

when the points are very close to each other, a local strategy is implemented to draw a minimal set of data points: the first point is drawn from the data points without restrictions, and following points are restricted to be in the hypersphere of radius r_m with center in the first point.

To discover hypotheses that correspond to the same object in the environment, the algorithm creates a fully connected graph, where the nodes represent models and the weight of each edge is a statistical measure of the common inliers between the nodes. The graph is represented by Matrix $C = [c_{ij}]_{m \times m}$, where the element c_{ii} represents the consensus (number of inliers) of the model θ_i and the non-diagonal element c_{ij} is a statistical measure of the common inliers between the models θ_i and θ_j , given Z . A large value of c_{ij} indicates a strong relationship between the models θ_i and θ_j ; that is, a high plausibility that the models represent the same object in the environment. Analogously, a small value indicates a low plausibility that θ_i and θ_j represent the same object. Matrix C is calculated at line 6 of Algorithm 1, where the function $I(z, \theta_i)$ is defined as

$$I(z, \theta_i) = \begin{cases} 1, & \text{if } d^{\wedge}(z, \theta_i) \leq t \\ 0, & \text{otherwise;} \end{cases}$$

here, t is a predefined threshold.

The selection step is performed in lines 7–21. The model with larger consensus θ_b is selected at line 9. If the consensus of θ_b is greater than a predefined threshold τ , then the consensus of other models is reduced. When $2c_{ib} > c_{ii}$ (line 13) the algorithm considers that models θ_b and θ_i represent the same object in the environment. Then model θ_i is marked as invalid at line 14. Each model is stored in the set Θ^* (line 11) and the set of these models are the base for the next iteration (line 22).

6 Experimental Results

We compare MuSAC with sequential RANSAC and MultiRANSAC algorithms by using the simulated environment shown in Figure 3(a). We generate 1000 random robot poses. A 360° range scan was taken from each pose, each measurement was corrupted with gaussian noise to each measurement. We use $\tau = 1.96$ std. dev. $\tau = 10$ points, $r_m = 1m$ for all methods; $m = 50$ lines for MuSAC and $W = 20$ lines for MultiRANSAC.

To figure out the performance of algorithms we use S_g y S_b defined as

$$\overline{S_g} = \frac{1}{n} \sum_{v=1}^n \frac{g_i}{l_i}, \quad \overline{S_b} = \frac{1}{n} \sum_{i=1}^n \frac{b_i}{l_i}$$

where: n , is the number of laser scans; l_i , is the number of lines that generate the i -th scan; g_i , is the number of lines correctly detected, and b_i is the number of lines incorrectly detected. $\overline{S_g}$ is the average ratio of correctly detected models; while $\overline{S_b}$ is the average ratio of spurious models detected.

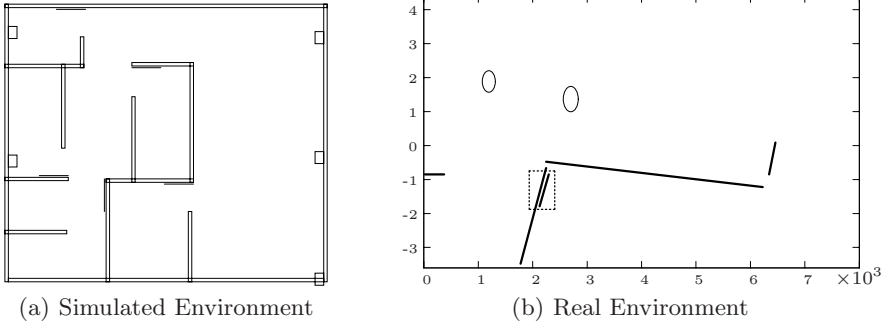


Fig. 3. **a.** Simulated environment. **b.** Lines extracted from a 2D laser scan in real environment. The dotted rectangle shows two parallel lines correctly discovered, the longer line is a wall while the other one is a door.

Table 1. Results using the simulated environment of Figure 3(a)

Algorithm	d_{\perp}			d_{\nearrow}		
	time (msec)	S_g	S_b	time (msec)	S_g	S_b
Sequential RANSAC	7989	0.510	3.440	967	0.956	1.653
MultiRANSAC	6434	0.704	0.186	988	0.773	0.123
MuSAC	3941	0.894	0.601	551	0.907	0.208

Experimental results are shown in Table 1. Sequential RANSAC is the worst because S_b is too high. MuSAC gets a higher value for S_g compared with MultiRANSAC. On the other hand, MultiRANSAC gets a better value for S_b . The average total time used by MuSAC is good enough for real time applications. It is important to note that, the directional distance d_{\nearrow} gets better results than d_{\perp} in all cases.

For the real test, 2D Laser scans were taken from our laboratory with a laser SICK LMS-200. One challenge for every algorithm is to correctly detect geometric models when they are very close to each other. Figure 3(b) shows a typical result. The dotted rectangle shows that two parallel but different lines were correctly found. Some lines from small objects, such as those marked with ellipses in figure 3(b), were not discovered due to the minimum consensus restriction (τ).

7 Conclusions

The MuSAC algorithm introduced in this paper is faster than MultiRANSAC and it is a good option to extract geometric models from laser scans in real time. The MuSAC algorithm generates a predefined number of random hypotheses from the laser scan and then decides which hypotheses represent the same object in the environment based on their consensus. This paper also introduces

the directional distance, this simple metric is helpful for rotational laser sensors because it considers the relationship between the measurement ray and the detected surface. In the near future we want to test our method for other geometric models (such as circles, spheres, etc).

References

1. Dufrenois, F., Hamad, D.: Fuzzy weighted support vector regression for multiple linear model estimation: application to object tracking in image sequences. In: IJCNN, pp. 1289–1294 (2007)
2. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* (1981)
3. Fitzgibbon, A.W., Eggert, D.W., Fisher, R.B.: High-level cad model acquisition from range images. *Computer-Aided Design* 29, 321–330 (1997)
4. Gorte, B.: Planar feature extraction in terrestrial laser scans using gradient based range image segmentation. In: *International Archives of Photogrammetry and Remote Sensing, IAPRS* (2007)
5. Han, F., Tu, Z.W., Zhu, S.C.: Range image segmentation by an effective jump-diffusion method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 1153 (2004)
6. Jiang, X.Y., Bunke, H.: Fast segmentation of range images into planar regions by scan line grouping. *Machine Vision and Applications*, 115–122 (1994)
7. Pulli, K., Pietikäinen, M.: Range image segmentation based on decomposition of surface normals. In: *Proc. of Scandinavian Conference on Image Analysis* (1993)
8. Thrun, S., Martin, C., Liu, Y., fhnel, D., Emery-Muntemerlo, R., Chakrabarti, D., Burgard, W.: A real-time expectation maximization algorithm for acquiring multi-planar maps of indoor environments with mobile robots (2003)
9. Torr, P.H.S., Murray, D.W.: The development and comparison of robust methods for estimating the fundamental matrix. *International journal of computer vision* (1997)
10. Torr, P.H.S., Zisserman, A.: Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding* 78, 2000 (2000)
11. Zuliani, M., Kenney, C.S., Manjunath, B.S.: The multiransac algorithm and its application to detect planar homographies. In: *IEEE International Conference on Image Processing* (September 2005)