Background Foreground Segmentation for SLAM

Padraig Corcoran, Adam Winstanley, Peter Mooney, and Rick Middleton, *Fellow, IEEE*

***Abstract*—To perform simultaneous localization and mapping (SLAM) in dynamic environments, static background objects must first be determined. This condition can be achieved using *a priori* information in the form of a map of background objects. Such an approach exhibits a causality dilemma, because such *a priori* information is the ultimate goal of SLAM. In this paper, we propose a background foreground segmentation method that overcomes this issue. Localization is achieved using a robust iterative closest point implementation and vehicle odometry. Background objects are modeled as objects that are consistently located at a given spatial location. To improve robustness, classification is performed at the object level through the integration of a new segmentation method that is robust to partial object occlusion.**

***Index Terms*—Background–Foreground segmentation, light detection and ranging (LIDAR).**

I. Introduction

Recently, several researchers have started to develop simultaneous localization and mapping (SLAM) algorithms for complex outdoor environments [1], [2]. One of the main challenges presented by such environments is dealing with large numbers of moving objects, such as people and other vehicles. SLAM assumes that the environment is static with moving objects treated as noise. It has been shown that the removal of such foreground objects leads to improved performance and is a necessity in busy outdoor environments [2], [3]. This condition can be achieved using a detection and tracking of moving objects (DTMO) system. On the other hand, DTMO requires the removal of static background regions so that dynamic objects can be identified and tracked. To achieve this goal, a map of such regions may be constructed using SLAM. Based on this discussion, it should be clear that DTMO and SLAM are strongly coupled and should not be implemented in isolation. Wang *et al.* [3] refers to such a system as *SLAM with DTMO* and showed that its implementation requires that sensor measurements can first be decomposed into stationary and moving objects. This step has received significantly less research focus than the development of SLAM techniques. A number of papers propose to use *a priori* information in the form of the actual map created by SLAM to discriminate between static and dynamic objects [4], [5]. This approach represents a causality dilemma. To have such *a priori* information, static and dynamic objects must first be discriminated [6]. In the context of object tracking, discriminating between static and dynamic objects based on whether the objects in question are currently tracked has been proposed [7]. This approach is also subject to the same causality dilemma, because static and dynamic objects must be discriminated to initialize and update tracks. Another disadvantage is assuming that objects always remain static or dynamic, which is not the case. The occupancy grid approach proposed by Weiss *et al.* [8] is subject to the same issue. Kondaxakis *et al.* [9] propose to overcome this issue by periodically refreshing the map of static objects. Zhao *et al.* [2] uses a similar strategy and also integrates object size features to improve classification. Diego *et al.* [10] proposed to determine static and dynamic objects by clustering points from two consecutive scans. This approach was not evaluated in the context of SLAM. Hahnel *et al.* [6] proposed to determine dynamic locations using the expectation–maximization (EM) algorithm. The expectation step computes a probabilistic estimate with regard to which measurements may correspond to static objects. The maximization step uses these estimates to determine the robot localization and map. This process is iterated until no further improvement can be achieved. Because classification is performed at the point level, this technique does not consider the likelihood that neighboring points will be of the same class. Pietzsch *et al.* [11] also proposes a strategy that performs static/dynamic classification at the point level. Zhao *et al.* [12] proposed a light detection and ranging (LIDAR) background–foreground segmentation algorithm. Their method assumes a nonmoving scanner.

We propose a new background–foreground segmentation method that improves upon the aforementioned techniques. This technique overcomes some of the limitations of previous approaches by integrating a new object segmentation technique with a static/dynamic point classification algorithm. Robustness to noise is improved by performing a final static/dynamic classification at the object level. The segmentation method proposed is a region-growing algorithm that builds upon the algorithm proposed by Hwang *et al.* [13]. The occlusion of one object by another can cause incorrect oversegmentation of the occluded object [14], [15]. Oversegmentation can also occur if an object contains a break or hole. The space between two legs of a pedestrian is one common example. The segmentation method of Broggi *et al.* [16] and Gidel *et al.* [17] cluster points using only proximity, without considering possible breaks. Mendes *et al.* [14] propose to overcome this problem by postprocessing the segmentation result. The segmentation method proposed here accounts for such cases, and no postprocessing is required.

Manuscript received July 13, 2009; revised November 11, 2009, May 7, 2010, October 30, 2010, and March 6, 2011; accepted April 10, 2011. Date of publication May 12, 2011; date of current version December 5, 2011. This work was supported in part by the Science Foundation Ireland through the National Development Plan under Strategic Research Cluster Grant 07/SRC/I1168 and Research Professor Award 07/RPR/I1177 and by Mr. M. Bell, who inspired this project, through a private donation. The Associate Editor for this paper was P. Grisleri.

P. Corcoran, A. Winstanley, and P. Mooney are with the Department of Computer Science, National University of Ireland, Maynooth, Co. Kildare, Ireland (e-mail: padraigc@cs.nuim.ie).

R. Middleton is with the Hamilton Institute, National University of Ireland, Maynooth, Co. Kildare, Ireland.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2011.2143706

We model background objects as objects that are consistently located at a given spatial location. Initially, a background–foreground pointwise classification of the LIDAR data is performed. Next, the proposed region-growing segmentation algorithm is applied, which returns a set of objects. Finally, based on the percentage of background points contained within each object, we classify each object as either background or foreground. This method does not require an *a priori* map and is robust to objects that change state. Because classification is performed at the object level, it is more robust to noise. Localization is achieved using a robust iterative closest point (ICP) implementation, which filters association due to moving objects. This approach was originally proposed by Rodriguez-Losada *et al.* [18]. Kondaxakis *et al.* [9] also used ICP to achieve localization for background–foreground segmentation but did not filter associations due to moving objects.

This paper is organized as follows. In the following section, we briefly introduce the LIDAR data set used within this paper and present our background–foreground point classification method. Section III describes the proposed segmentation and object classification methods. Results are presented in Section IV. Finally, in Section V, we draw the conclusions.

II. Background–Foreground Point Classification

Conceptually, our algorithm operates as follows. For each LIDAR point i in the current scan at time t, we attempt to find a correspondence between this point and point(s) in consecutive previous scans. If correspondences are found, we infer that an object has always been situated at this location and, consequently, has a high probability of being static. If such correspondences cannot be found and the corresponding location was visible in previous scans, we infer that an object has just appeared at this location and has a high probability of being dynamic. To relate spatial locations between scans, an estimate of vehicle pose is required. This condition is obtained using a robust ICP implementation, which ignores associations that do not explain the main motion of the scan and will therefore filter association due to moving objects. The initial alignment of the data is achieved using vehicle odometry. Greater details concerning this technique can be found in [18].

LIDAR measurements can be described in the following two ways: 1) the local coordinate system of the vehicle in question and 2) the global coordinate system in which the vehicle is situated. All measurements are initially made in terms of polar coordinates relative to the local coordinate system of the vehicle. To relate measurements between different scans, we must do so through the global coordinate system. At each time t, when a new scan is encountered, we update the vehicle position in this coordinate system. For each LIDAR point i at time t, we determine whether a correspondence between this point and a point at time $t-1$ can be made as follows. We transform point i at t from its local polar coordinate system to its local Cartesian coordinate system and then, in turn, to the global Cartesian coordinate system. We next determine the location of this point in terms of the local Cartesian coordinate system defined by the vehicle pose at time $t-1$. We then convert this local Cartesian coordinate to a local polar coordinate, where the variables $e\beta^i_{t-1}$ and er^i_{t-1} represent the corresponding estimated angle and radius, respectively. For example, consider the illustration in Fig. 1, where the vehicle at time t measures the location of the point i relative to its current pose. The polar coordinates of this point relative to the vehicle pose at time $t-1$ are estimated to be $e\beta^i_{t-1}$ and er^i_{t-1}.

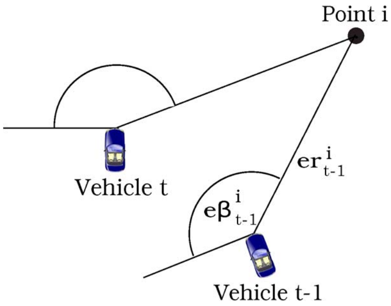

Fig. 1. Polar coordinates of the point i relative to the vehicle pose at time $t-1$ are estimated to be $e\beta^i_{t-1}$ and er^i_{t-1}.

Let the variable C^i_{t-1} represent if a *correspondence* between the point i in the scan at time t and any point in the scan at time $t-1$ can be obtained. Let the variable V^i_{t-1} represent if the point i in the scan at time t was *visible* to the scanner at time $t-1$ and no correspondence could be obtained. If this variable evaluates to true, this case represents that no object occluded the view of the location of the point in question.

The evaluation of V^i_{t-1} and C^i_{t-1} are determined using Algorithm 1, which we now describe. Both variables take the value 1 when evaluated to true and 0 when evaluated to false. If $e\beta^i_{t-1}$ is not within the scanners view at time $t-1$ because it is behind the vehicle, V^i_{t-1} and C^i_{t-1} are evaluated to be false (see line 2 in Algorithm 1). If the point i is not within the scanner's maximum range at time $t-1$, again, V^i_{t-1} and C^i_{t-1} evaluate to false (line 4 in Algorithm 1). If point i is within the scanner's view and not beyond its range at time $t-1$, we read the measured radius value at the angle $e\beta^i_{t-1}$ in the scan at time $t-1$ and call this value mr^i_{t-1}. Subsequently, C^i_{t-1} and V^i_{t-1} are evaluated using the equations on lines 6 and 7 in Algorithm 1. CT is a *correspondence threshold*, which specifies how similar er^i_{t-1} and mr^i_{t-1} must be for a correspondence to be obtained. VT is a *visibility threshold*, specifying how much smaller than er^i_{t-1} the point at angle $e\beta^i_{t-1}$ in scan time $t-1$ must be for it to be determined an object that blocks the view of the required location.

Algorithm 1: Determine correspondence and visibility.

```
1: if $e\beta^i_{t-1}$ is outside the view then
2:     $C^i_{t-1} = 0, V^i_{t-1} = 0$
3: else if $er^i_{t-1}$ is outside the range then
4:     $C^i_{t-1} = 0, V^i_{t-1} = 0$
5: else
6:     $C^i_{t-1} = (|mr^i_{t-1} - er^i_{t-1}| < CT)$
7:     $V^i_{t-1} = (mr^i_{t-1} - er^i_{t-1} > VT) \wedge \neg C^i_{t-1}$
8: end if
```

In reality, $e\beta^i_{t-1}$ will rarely equal an angle for which we have a measurement but will fall between two angles for which we have measurements. To address this issue and allow for uncertainty in position estimation, we repeat the calculation of C^i_{t-1}

and V^i_{t-1} for the two closest angles at either side of this angle. We then perform logical OR for the two C^i_{t-1} results and logical AND for the two V^i_{t-1} results. The motivation for the OR operation is that, if it is possible to make a correspondence, we do so, whereas the motivation for the AND operation is that we do not declare a point to be visible if there is a possibility that it is not.

The aforementioned method describes how we can determine if point i at time t was visible at time $t-1$ and if a correspondence exists. To conclude with confidence that a point is static or dynamic, it is important to examine more than a single previous scan. We therefore calculate the following two statistics from the previous n scans:

$$CP^i = \frac{1}{n}\sum_{j=1}^{n} C^i_{t-j}$$
$$VP^i = \frac{1}{n}\sum_{j=1}^{n} V^i_{t-j}. \tag{1}$$

The term CP^i represents the *correspondence percentage*, which is equal to the percentage of times that a correspondence can be found for the point i with previous scans. The term VP^i represents the *visible percentage* and is equal to the percentage of times that the point i is visible in previous scans but no correspondence could be found. Using the aforementioned statistics, we derive a background–foreground classification for each point i, which are denoted as BF^i, as follows. If point i is beyond the maximum scanner range, we classify it as so. If VP^i exceeds a threshold known as VPT, we classify it as dynamic. If CP^i is greater than a threshold known as CPT, we classify it as static. Otherwise, we classify it as possibly dynamic. The pseudocode for this method is given in Algorithm 2.

Algorithm 2: Static–Dynamic point classification.

```
if $ar^t_i \geq max\ scanner\ range$ then
    $BF^i = beyond\ range$
else if $VP^i \geq VPT$ then
    $BF^i = dynamic$
else if $CP^i \geq CPT$ then
    $BF^i = static$
else
    $BF^i = possibly\ dynamic$
end if
```

To produce accurate results, this background-subtraction algorithm requires an estimate of vehicle position, which is obtained from odometry. Due to inherent inaccuracies in equipment and effects such as slippage, position estimates that were obtained in this manner will never be entirely accurate and can suffer from cumulative error. This effect is minimized in our algorithm, because only relative measurements between a small number of consecutive scans are made. The global error in position will not affect the result.

III. Background–Foreground Object Segmentation and Classification

The proposed region-growing segmentation algorithm is presented in Algorithm 3 and is outlined as follows. Each point in a given scan has a corresponding angle, radius, and background–foreground class (see Algorithm 2), which are denoted as β^i, r^i, and BF^i, respectively. Each point is assigned an index i for $i = 0\ldots 360$ such that points are ordered by the increasing value of β^i. The list ga represents the *group assignment* of each point, and all elements in this list are initially assigned the value 0. The variable $groupSize$ represents the number of groups currently found and is initialized to the value 0.

For each point i not previously assigned a group and with a BF^i class not equal to $Beyond\ Range$, the following steps are performed. First, the point is assigned to a new unique group (see lines 3 and 4). A region-growing segmentation method is then started at the point in question (see line 5). The segmentation method overcomes the problem of oversegmentation caused by the occlusion of one object by another as follows. The region-growing method searches OT (*occlusion threshold*) points beyond a boundary for a continuation of the current object that it is attempting to grow. This method first attempts to grow the region on one side (see lines 10–14) and then the other side (see lines 15–19).

For an object to successfully be grown from a point i to include a point $i+j$, the following two conditions must be satisfied: 1) The *absolute difference*, which is represented by the variable AD between r^i and r^{i+j} must be less than a *boundary threshold* represented by the variable BT (see lines 10 and 11), and 2), the point in question must not be previously assigned to a group (line 11). The same two conditions must be satisfied if the object will successfully be grown from a point i to include a point $i-j$. If the region is successfully grown to include a particular point, this point is assigned to the corresponding group (see lines 12 and 17), and the region-growing function is recursively called (see lines 13 and 18). Following the successful completion of this algorithm, all points that correspond to a particular object will have a unique $ga[i]$ value > 0.

Algorithm 3: Object segmentation.

```
1: for $i = 0$ to 360 do
2:     if $(ga[i] == 0) \wedge (BF^i \neq beyond\ range)$ then
3:         groupSize++
4:         $ga[i] =$ groupSize
5:         regionGrow(i)
6:     end if
7: end for
8: function regionGrow(i)
9: for $j = 1$ to $OT$ do
10:     $AD = |r^i - r^{i+j}|$
11:     if $(ga[i+j] == 0) \wedge (AD \leq BT)$ then
12:         $ga[i+j] = ga[i]$
13:         regionGrow$(\mathrm{i}+\mathrm{j})$
14:     end if
15:     $AD = |r^i - r^{i-j}|$
16:     if $(ga[i-j] == 0) \wedge (AD \leq BT)$ then
17:         $ga[i-j] = ga[i]$
18:         regionGrow$(\mathrm{i}-\mathrm{j})$
19:     end if
20: end for
21: end function
```

We individually classify each object returned by Algorithm 3 as a background or foreground object. For each object, we calculate the percentage of points within this object classified as $static$ by Algorithm 2. If this percentage is above a specified threshold, we classify the object in question as background; otherwise, we classify it as foreground. Such a percentage-based approach is important, because, due to vehicle movement, part of a background object may come into view, although the remainder of the object may have been visible. LIDAR points that correspond to such previously nonvisible areas will be classified as $possibly\ dynamic$, although the remainder of the object points are classified as $static$. On the other hand, due to different parts of a moving object that is located at the same position over time, a foreground object may contain points classified as $static$.

IV. Results

All data that were used in this paper are publicly available and were originally captured by Wang *et al.* [19]. Using a SICK LMS 221 LIDAR scanner and an omnidirectional camera positioned at the front of a vehicle, the data were captured in the city of Pittsburgh, PA. Suitable parametrization of the proposed algorithm is important. In the background–foreground point classification in Algorithm 2, the most important parameter is n, which is equal to the number of previous scans with which the correspondence and visibility of a given point are determined using Algorithm 1. Very few previous scans will mean that results will be less accurate due to classifications that were made over a short time window. If too many previous scans are used, the probability that static objects are correctly classified decreases. The reasons for this case are given as follows. For a given static object, the further back in time that we look, the greater the probability that part of or the whole object was not visible due to the following three cases: 1) It is beyond the visible range; 2) it is not within the viewing angle; or 3) it is occluded. Consequently, there is less chance that the necessary correspondence is obtained for classification as static. The worst-case situation is when the vehicle both moves forward and turns, causing the view to change by two degrees of freedom. By evaluating multiple values, it was found that $n = 4$ gave the best results. In the same algorithm, we found that the parameter values $CT = 30$ cm and $VT = 50$ cm gave the best results. Setting CT too low resulted in static objects to be determined dynamic. Although setting CT too high resulted in dynamic objects to be determined static, slow dynamic objects cannot accurately be classified when the distance moved between scans is smaller than the uncertainty in vehicle localization, object localization, or sensor accuracy. The percentage parameters VPT and CPT in Algorithm 2 must be set reasonably high. This approach ensures that all points that were classified as static or dynamic have a high probability of being so. We found that the values $VPT = 0.75$ and $CPT = 0.75$ gave the best results. For the object segmentation in Algorithm 3, it was found that the parameter values of $OT = 7$ and $BT = 150$ cm were suitable for object segmentation. The percentage of static points that an object must contain for it to be classified a static object was set to 75%. This value was found to give good results and minimize misclassification.

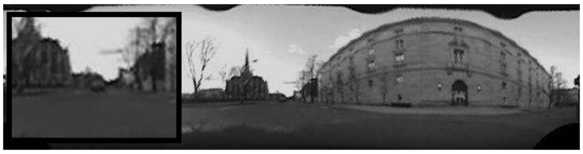

Fig. 2. Image center is enhanced and displayed within the black box on the left.

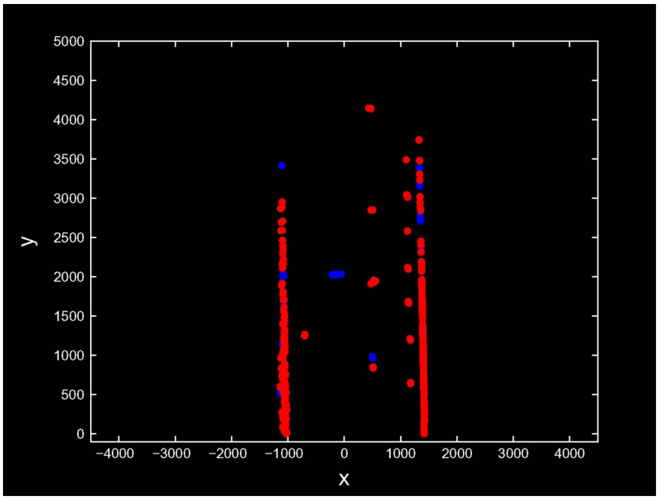

Fig. 3. Each point is classified as static (red), dynamic (green), or possibly dynamic (blue). This scan corresponds to the image in Fig. 2.

To determine the performance of each algorithm, in a qualitative manner, we visually inspected the plotted output and cross validated this output against the corresponding camera imagery for a number of test cases (see Fig. 2). This strategy is used in several similar works [15], [20]. Fig. 3 displays the result of background–foreground point classification for the scene in Fig. 2. LIDAR points that correspond to the sole vehicle directly in front of the scanner are classified as possibly dynamic. This case is because, at each time step, the vehicle in question moves in a northerly direction in the figure and therefore moves into an area that was previously hidden by itself. This case highlights the ability of the algorithm to correctly deal with uncertainty due to the lack of visibility. In this image, we also see that the algorithm correctly classifies the vast majority of points that correspond to background buildings as static. One section of the building on the right is incorrectly classified as possibly dynamic. This case is because this section has only just come into view, and therefore, the algorithm cannot determine if it is background or foreground. Fig. 5 displays the result of background–foreground point classification for the scene in Fig. 4. In this example, the vehicle takes a right turn and is not traveling in a straight line as in the previous example, which represents an important test case. There is a vehicle that moves in a southeasterly direction just below and to the right of center, as shown in Fig. 5. The front of this vehicle moves to an area that was previously visible and unoccupied; therefore, points that correspond to its front are classified as dynamic. In the same figure, the majority of points that correspond to buildings are classified as static. Just to the left and above the center, there

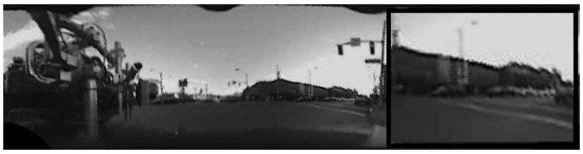

Fig. 4. Image center is enhanced and displayed within the black box on the right.

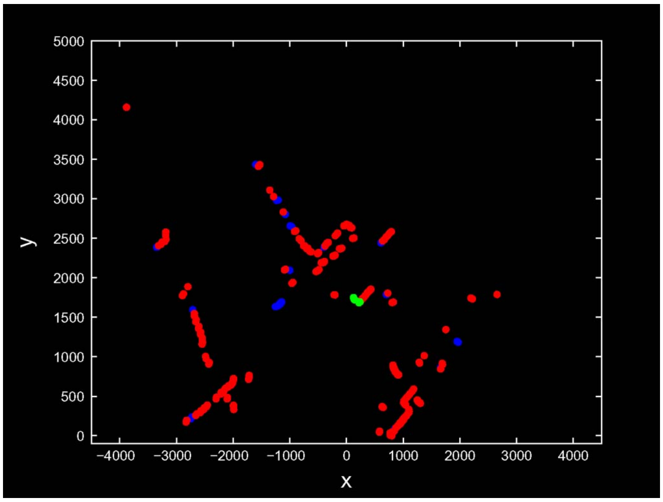

Fig. 5. Each point is classified as static, dynamic, or possibly dynamic. This scan corresponds to the image in Fig. 4.

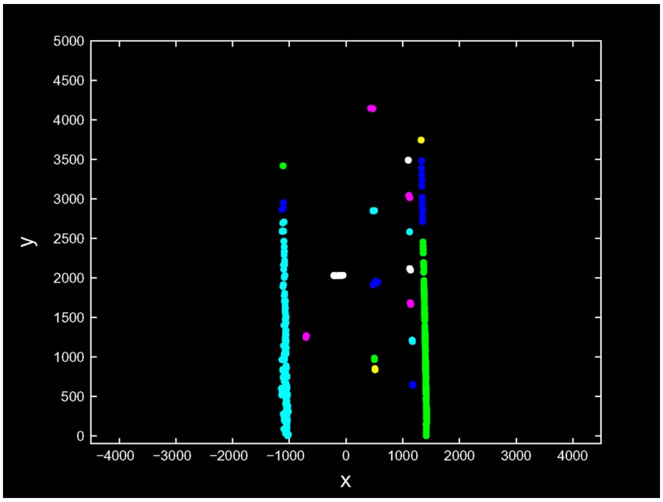

Fig. 6. Object segmentation of the scan that corresponds to the image in Fig. 2. Each object is represented by a different color.

are a few points that correspond to a building and are classified as possibly dynamic. This case is because these locations were occluded in previous scans, and therefore, the algorithm cannot determine if they are static or dynamic.

The results achieved by the proposed object segmentation method in Algorithm 3 were very positive. Fig. 6 displays the segmentation of the scene in Fig. 2. The algorithm accurately segments both background objects such as buildings and foreground objects such as vehicles. The building on the right illustrates the capability of our segmentation algorithm to overcome the issue of oversegmentation caused by occlusion. Although the building is occluded by small static objects, the segmentation algorithm correctly represents it as a single object. The result of classifying each object in this segmentation as background or foreground, based on the percentage of static points contained within each object, is displayed in Fig. 7. We can see in this figure that most building objects are classified as background, whereas the moving vehicle at the figure center is classified as foreground. Part of the building in the upper right of this figure is classified as foreground for the reason discussed earlier. Fig. 8 displays the segmentation of the scene in Fig. 4, which highlights the ability of the algorithm to segment both background and foreground objects. The result of classifying each object in this segmentation as background or foreground is displayed in Fig. 9. The algorithm correctly classifies the majority of background and foreground objects, including the two moving vehicles to the left and right of the figure center and the buildings.

To perform a quantitative evaluation of our background–foreground segmentation and classification algorithm, we constructed the ground truth of all vehicles and buildings in a subset of LIDAR scans through the visual inspection of the

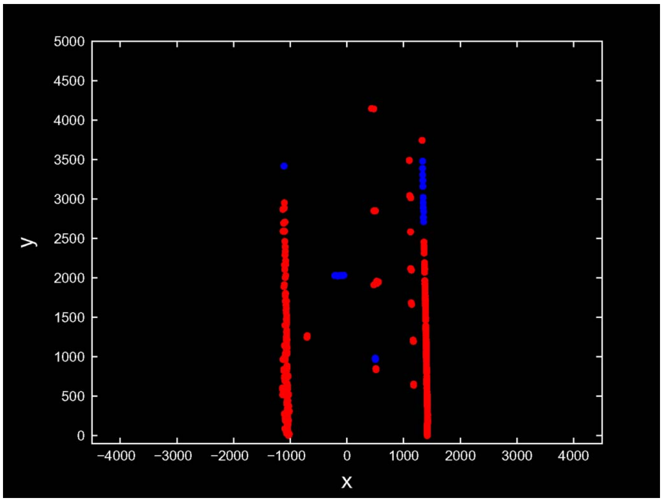

Fig. 7. Background–Foreground segmentation of the objects in Fig. 6. All points that belong to objects that were classified as background and foreground are represented by the colors red and blue, respectively.

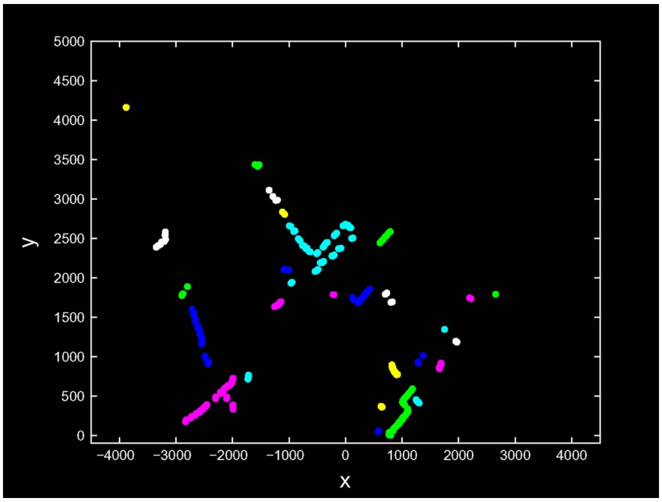

Fig. 8. Object segmentation of the scan that corresponds to the image in Fig. 4.

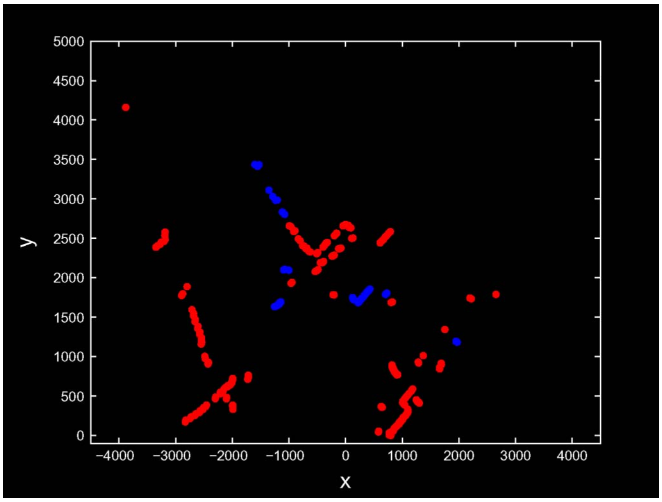

Fig. 9. Background–Foreground segmentation of the objects in Fig. 8.

Object	Background	Foreground
Building	79%	21%
Moving Vehicle	0%	100%
Static vehicle	75%	25%

Fig. 10. Classification performance matrix.

LIDAR and corresponding imagery. We then computed the classification performance matrix by comparing the result with the ground truth using this test set. The subset of scans used consisted of every 50th scan, i.e., scans 1, 50, 100, and 150. This approach resulted in a test set of size 74. The resulting classification performance matrix is displayed in Fig. 10. In this matrix, we can see that all moving vehicles in the test set were correctly classified as foreground. More than three fourths of the building was correctly classified as background, and three fourths of static vehicles were correctly classified as background. Due to vehicle movement, new buildings and static vehicles constantly come into view. No correspondence with previous scans can be established for such objects, and they will automatically be classified as foreground. Thus, obtaining 100% classification accuracy is not possible. To evaluate performance in the context of a complete SLAM algorithm, we implemented the ICP-based SLAM approach of Wang *et al.* [4]. Fig. 11(a) and (b) show a subsection of the resulting maps, corresponding to the area in Fig. 2, with and without background foreground segmentation performed, respectively. The map that was created using the proposed preprocessing step is less noisy, resulting in trees and building walls more clearly represented.

V. Conclusion

In this paper, a background–foreground object segmentation and classification method that can form the initial step in a SLAM with DTMO system has been proposed. The results achieved by a qualitative and quantitative evaluation of this method are very positive. The algorithm correctly segments and classifies the majority of background and foreground objects. When used as a preprocessing step to SLAM, the proposed technique has been shown to generate a more accurate map.

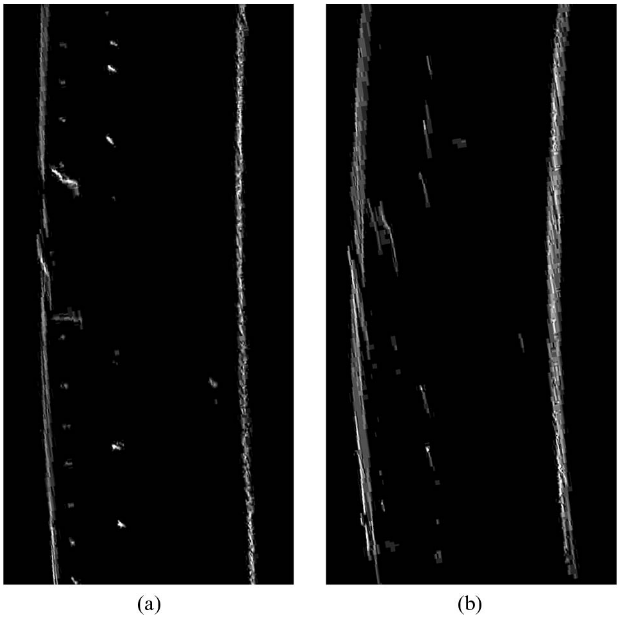
(a) (b)

Fig. 11. Partial map generated by ICP SLAM (a) with and (b) without background–foreground segmentation performed.

Acknowledgment

The authors would like to thank Dr. B. Wang for the provision of data, Dr. D. Rodriguez-Losada for answering several iterative closest point (ICP)-related questions, and Prof. J. J. Leonard for the helpful discussions.

References

[1] M. S. Darms, P. E. Rybski, C. Baker, and C. Urmson, “Obstacle detection and tracking for the urban challenge,” *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 475–485, Sep. 2009.

[2] H. Zhao, M. Chiba, R. Shibasaki, X. Shao, J. Cui, and H. Zha, “A laser-scanner-based approach toward driving safety and traffic data collection,” *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 534–546, Sep. 2009.

[3] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, “Simultaneous localization, mapping and moving object tracking,” *Int. J. Robot. Res.*, vol. 26, no. 9, pp. 889–916, Sep. 2007.

[4] C.-C. Wang and C. Thorpe, “Simultaneous localization and mapping with detection and tracking of moving objects,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, vol. 3, pp. 2918–2924.

[5] D. F. Wolf and G. S. Sukhatme, “Mobile robot simultaneous localization and mapping in dynamic environments,” *Auton. Robots*, vol. 19, no. 1, pp. 53–65, Jul. 2005.

[6] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun, “Map building with mobile robots in dynamic environments,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2003, pp. 1557–1563.

[7] D. Schulz, W. Burgard, D. Fox, and A. Cremers, “Tracking multiple moving targets with a mobile robot using particle filters and statistical data association,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Seoul, Korea, 2001, vol. 2, pp. 1665–1670.

[8] T. Weiss, B. Schiele, and K. Dietmayer, “Robust driving path detection in urban and highway scenarios using a laser scanner and online occupancy grids,” in *Proc. IEEE Intell. Veh. Symp.*, 2007, pp. 184–189.

[9] P. Kondaxakis, H. Baltzakis, and P. Trahanias, “Learning moving objects in a multitarget tracking scenario for mobile robots that use laser range measurements,” in *Proc. IEEE/RSJ Int. Conf. IROS*, Piscataway, NJ, 2009, pp. 1667–1672.

[10] G. D. Tipaldi and F. Ramos, “Motion clustering and estimation with conditional random fields,” in *Proc. IEEE/RSJ Int. Conf. IROS*, St. Louis, MO, 2009, pp. 872–877.

[11] S. Pietzsch, T. D. Vu, J. Burlet, O. Aycard, T. Hackbarth, N. Appenrodt, J. Dickmann, and B. Radig, “Results of a precrash application based on laser scanner and short-range radars,” *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 584–593, Dec. 2009.

[12] H. Zhao and R. Shibasaki, "A novel system for tracking pedestrians using multiple single-row laser-range scanners," *IEEE Trans. Syst., Man, Cybern. A: Syst., Humans*, vol. 35, no. 2, pp. 283–291, Mar. 2005.

[13] J. P. Hwang, S. E. Cho, K. J. Ryu, S. Park, and E. Kim, "Multiclassifier based LIDAR and camera fusion," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 30–Oct., 3 2007, pp. 467–472.

[14] A. Mendes, L. Bento, and U. Nunes, "Multitarget detection and tracking with a laser scanner," in *Proc. IEEE Intell. Veh. Symp.*, Parma, Italy, Jun. 2004, pp. 796–801.

[15] F. Nashashibi and A. Bargeton, "Laser-based vehicles tracking and classification using occlusion reasoning and confidence estimation," in *Proc. IEEE Intell. Veh. Symp.*, Eindhoven, The Netherlands, Jun. 2008, pp. 847–852.

[16] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri, and H. G. Jung, "A new approach to urban pedestrian detection for automatic braking," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 594–605, Dec. 2009.

[17] S. Gidel, P. Checchin, C. Blanc, T. Chateau, and L. Trassoudaine, "Pedestrian detection and tracking in an urban environment using a multilayer laser scanner," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 579–588, Sep. 2010.

[18] D. Rodriguez-Losada and J. Minguez, "Improved data association for ICP-based scan matching in noisy and dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 3161–3166.

[19] C.-C. Wang, D. Duggins, J. Gowdy, J. Kozar, R. MacLachlan, C. Mertz, A. Suppe, and C. Thorpe, Navlab Slammot Datasets, Pittsburg, PA, Carnegie Mellon Univ., May 2004. [Online]. Available: www.cs.cmu.edu/~bobwang/datasets.html

[20] F. Fayad and V. Cherfaoui, "Tracking objects using a laser scanner in driving situation based on modeling target shape," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2007, pp. 44–49.

Padraig Corcoran received the B.Sc. degree in computer science and software engineering and the Ph.D. degree in computer science in 2004 and 2008, respectively.

He is currently a lecturer and researcher with the Department of Computer Science, the National University of Ireland, Maynooth, Ireland.

Adam Winstanley received the B.A. degree in archaeology from the University of Cambridge, Cambridge, U.K., in 1978 and the M.Sc. and Ph.D. degrees in computer science from Queen's University, Belfast, U.K., in 1987 and 1991, respectively.

After working as a post-doctoral researcher with Queen's University and with the University of Ulster, U.K., he became a lecturer of Computer Science with the National University of Ireland (NUI), Maynooth, Ireland, in 1995. Currently, he is a Senior Lecturer and Head of Department of Computer Science and Senior Research Associate of the National Centre for Geocomputation at NUI Maynooth.

Peter Mooney received the B.Sc. and Ph.D. degrees in computer science from the National University of Ireland (NUI), Maynooth, Ireland.

He is currently the Environmental Research Data Manager with the Irish Environmental Protection Agency and a Postdoctoral Research Fellow with the Department of Computer Science, NUI Maynooth.

Rick Middleton (F'99) was born December 10, 1961, in Newcastle, Australia. He received the B.Sc., B.Eng., and Ph.D. degrees from the University of Newcastle in 1983, 1984, and 1987, respectively.

His research interests include a range of control systems theory and applications.

Dr. Middleton has served as an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, and *Automatica*, and has been Head of the Electrical and Computer Engineering Department at the University of Newcastle and a VP of the IEEE Control Systems Society. He is currently a Research Professor with the Hamilton Institute, the National University of Ireland (NUI), Maynooth, Ireland, and President of the IEEE Control Systems Society for 2011.