Region-Growing Planar Segmentation for Robot Action Planning

Reza Farid

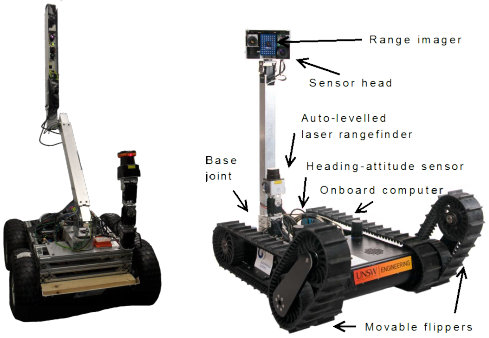
Institute for Integrated and Intelligent Systems, Griffith University,
170 Kessels Rd, Nathan, QLD 4111, Australia
r.farid@griffith.edu.au

Abstract Object detection, classification and manipulation are some of the capabilities required by autonomous robots. The main steps in object classification are: segmentation, feature extraction, object representation and learning. To address the problem of learning object classification using multi-view range data, we used a relational approach. The first step of our object classification method is to decompose a scene into shape primitives such as planes, followed by extracting a set of higher-level, relational features from the segmented regions. In this paper, we compare our plane segmentation algorithm with state-of-the-art plane segmentation algorithms which are publicly available. We show that our segmentation outperforms visually and also produces better results for the robot action planning.

Keywords: object classification, robot action planning, planar segmentation, point cloud, range data.

1 Introduction

A considerable amount of research has been devoted to generic object recognition (Opelt, 2006; Vasudevan et al., 2007; Shin, 2008; Endres, 2009), which is required by robots in many tasks. For instance, in service robotics applications, such as a catering or a domestic robot (Shin, 2008), the robot must recognise specific kinds of tableware, while the robot's ability to distinguish a set of products is necessary in industrial applications (Endres, 2009). We are mostly interested in urban search and rescue; where a team of robots are sent to a post-disaster environment. The robot's mission is to traverse the arena, to search for victims while making a map of the area. Rescue robots may be tele-operated or autonomous. When running autonomously, classification of objects is useful for reporting to human rescuers what is in the environment as well as determining the robot's behaviour. For example, recognising a *staircase* can provide necessary information to a wheeled robot (Figure 1a) to avoid that object, whereas a tracked robot (Figure 1b) is capable of climbing stairs but it must reconfigure its flippers (Figure 2) to be able to climb (Kalantari et al., 2009) successfully as shown in Figure 3. Another example is to use the relation between surfaces to grasp objects (Prankl et al., 2013).



(a) Emu **(b)** Negotiator

Figure 1: Some Autonomous Robots (Team CASuality in RoboCup 2011) (McGill et al., 2012)

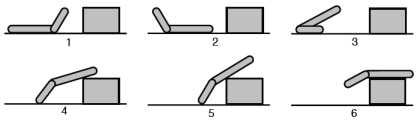


Figure 2: Robot Planning

Figure 3: Robot climbing stairs

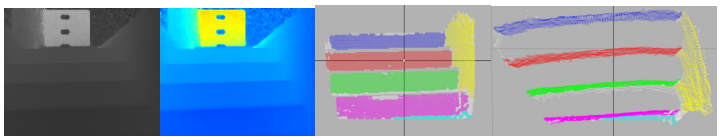


Figure 4: Range image and corresponding point cloud from front and top view (Farid and Sammut, 2014a)

Range cameras have become popular in robotics because they are small, light, consume relatively little power and have the ability to produce range measurements of up to several metres, making them suitable for indoor use. Range images are acquired by these 3D range/depth cameras, such as the Microsoft Kinect and ASUS Xtion PRO LIVE. These images are like grey scale images in which the value of each pixel represents the distance of the sensor to the surface of an object in a scene from a specific viewpoint (Gächter et al., 2006), and can be used to infer the shape of the object (Hegazy and Denzler, 2009). The Kinect and Xtion sensors also provide a colour video image. However, in this research, only the depth image is used for object recognition as it is capable of operating in the dark, which is often required in search and rescue operations. Furthermore, colour calibration under different lighting conditions is troublesome (Opelt, 2006).

A range image can be transformed into a point cloud by converting each pixel of the image into 3D coordinates. Figure 4 shows a range image of a *staircase* with four steps. The image was taken by a robot positioned in front of the staircase. In the leftmost image (the grey scale), darker colours represents closer surfaces. The next image (a colour-mapped version) is presented for clarity. The next two images are front and top views for the same point cloud, in which the point cloud is segmented into planes that are identified by unique colours. Since a range image is taken from one viewpoint, it only provides a partial view of a scene.

An object class describes a set of instances that share common properties, such as shape or function. A common starting point for finding the properties to be used in similarity matching is to segment the image into different regions and to characterise the relationships between those regions. Segmenting a point cloud can be viewed as the process of assigning each point to a region, with an accompanying label.

In previous work (Farid and Sammut, 2012b; 2014a), we extracted planes from a 3D point cloud based on a region growing plane segmentation algorithm (Farid and Sammut, 2012a; 2014c) and used them as primitives for object categorisation. In this paper, we show that our plane segmentation algorithm outperforms state-of-the-art plane segmentation methods which are publicly available. For this purpose, in the following sections, we compare the methods based on their visual results and the suitability of the plane features for robot action planning.

2 Background Work

Planes are useful features in built environments, including urban search and rescue for identifying floors, walls, stairs, ramps and other terrain that the robot is likely to encounter. Modelling a scene from planar patches is used in computer vision, robotics and augmented reality (Prankl et al., 2013). For example, it has been used for scene understanding (Bartoli, 2007; Xu and Petrou, 2011), localisation (Mohr et al., 1992) and 3D virtual reconstruction of the environment (McGill et al., 2012).

Our earlier approach (Farid and Sammut, 2012b; 2014a) was most closely related to Shanahan (2002) and Shanahan and Randell (2004) who used a logic program as a relational representation for 3D objects in 2D line drawings, while abduction is used in object recognition. We extended this representation, replacing the 2D lines with 3D planes. Furthermore, we used ALEPH (Srinivasan, 2002) to learn the logic programs from instances obtained by a robot equipped with a depth camera.

The fact that all points belonging to the same plane are supposed to have approximately the same normal vector, formed the core of the our segmentation algorithm. We introduced a region-growing plane segmentation algorithm based on neighbourhood normal vector similarity to segment an object into a set of planar surfaces. The method starts with a point and traverses the other points through the neighbourhood structure. To decide if the point can be added to the planar surface, it must satisfy the planar surface criteria, which determine when to add a point to a region. Our algorithm is based on using neighbouring points to grow the region. This is where the distance threshold, δ, can be used to decide whether a point is too far away to be accepted as a neighbour for a point. If a point is not too far, it can be included in the *not visited neighbours* list, *candidates*, as shown in the algorithm 1. We have used the below values for input variables:

`min_neighbour_num = 4` `base_update_step = 8`
`num_initial_points = 16` $\theta = 15^\circ - 20^\circ$

Features of these planar regions and their relationships were generated to form a planar description for an object class. The segmentation result and features were used for learning. We also showed that the learning system was able to use other primitives such as cylinders and spheres for the same purpose (Farid and Sammut, 2014b). A relational representation is useful in this application because it is our interest to recognise objects that are characterised by relationships between its parts, as in the steps constituting a staircase, and the number of parts may not be fixed, as the number of steps in a staircase can vary.

Ideally, an off-the-shelf segmentation method would have been used to decompose a scene into shape primitives. Several methods claim to provide good plane segmentation. However, they are not publicly available or they are not usable as claimed (Farid, 2014). Two algorithms are provided by PCL (Rusu and Cousins, 2011) using the RANdom SAmple Consensus (RANSAC) (Fischler and Bolles, 1981) algorithm. We use these state-of-the-art plane segmentation methods especially because they are publicly available:

- One of the PCL algorithms, setting the model type as SACMODEL_PLANE, uses 3D points belonging to the point cloud[1] without considering normals or any additional constraints. We call it as SP.
- The other algorithm, using SACMODEL_NORMAL_PLANE for model type, has an additional constraint similar to the method used in our research. We call it as SNP. It assumes the normal of each point must be parallel to the output plane normal within a maximum angular difference[2]. The use of SNP has been shown as a part of PCL's tutorial for cylinder model segmentation[3].

These two methods require a few parameters such as "Distance Threshold" and "Angle Threshold" to decide whether a point must be added to a plane. We will discuss these parameters later.

PCL has an algorithm as region growing segmentation. However, this algorithm merges the points to form a segment considering a smoothness constraint. The output clusters can be considered as smooth surfaces, not primitives such as planes, spheres and cylinders. This algorithm can be used to cluster the point cloud before passing each cluster to other segmentation algorithms such as SP and SNP. That is why we will not consider this algorithm for comparison in this paper. Our algorithm will be compared with SP and SNP visually and based on the quality of the segmented planes.

3 Experimental Evaluation

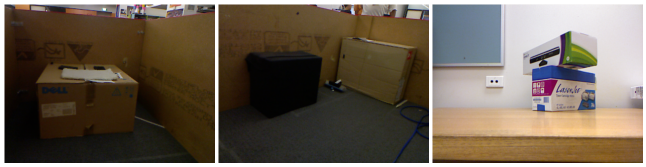
3.1 Dataset

Figure 1 shows several robots used for urban search and rescue. These ground robots were designed to participate in the RoboCup Rescue Robot competition, held annually. The competition arena uses elements developed by the US

[1] http://pointclouds.org/documentation/tutorials/planar_segmentation.php
[2] http://docs.pointclouds.org/1.7.0/group__sample__consensus.html
[3] pointclouds.org/documentation/tutorials/cylinder_segmentation.php

Algorithm 1 Region growing plane segmentation algorithm using normal vectors

Input: $PointCloud$, $normal\ vector\ for\ all\ points\ in\ PointCloud$
Input: $min\_neighbour\_num > 0$, $base\_update\_step > 0$
Input: $num\_initial\_points > 0$, min_region_size
Input: θ // angle threshold
Input: δ // distance threshold
Input: $angle\_mf < 1$ // angle modifying factor
$R \leftarrow \{\}$ // output: Regions
for all p in the PointCloud **do**
  if $p\ is\ visited \vee p\ is\ rejected$ **then**
    continue
  else if $number\_of\_usable\_neighbour(p) < min\_neighbour\_num$ **then**
    continue
  end if
  $C_R \leftarrow p$
  $Base\_normal \leftarrow get\_normal\_vector(p)$
  $candidates \leftarrow get\_not\_visited\_neighbours(p, \delta)$
  for all q in candidates **do**
    if $Size(C_R) < num\_initial\_points \vee mod(Size(C_R), base\_update\_step) = 0$ **then**
      $Base\_normal \leftarrow get\_average\_normal\_vectors(C_R)$
    end if
    $current\_angle \leftarrow get\_angle(Base\_normal, get\_normal\_vector(q))$
    $accepted \leftarrow$ **false**
    if $Size(C_R) < num\_initial\_points$ **then**
      if $current\_angle < \theta$ **then**
        $accepted \leftarrow$ **true**
      end if
    else if $current\_angle < \theta * angle\_mf$ **then**
      $accepted \leftarrow$ **true**
    end if
    if $accepted$ **then**
      $C_R \leftarrow C_R \cup q$
      $set\_visited(q)$
      $candidates \leftarrow candidates \cup get\_not\_visited\_neighbours(q, \delta)$
    end if
  end for
  if $Size(C_R) > min\_region\_size$ **then**
    $set\_final\_normal\_vector(C_R)$
    $R \leftarrow R \cup C_R$
  end if
end for
return R



(a) *box* class

(b) *stairs* class

(c) *pitch/roll ramp* class

Figure 5: Examples of instances used in this research

National Institute of Standards and Technology (NIST, 2010) to certify robots for emergency operations. These elements simulate typical hazards that might be encountered in buildings damaged by a disaster such as an earthquake. We captured data during RoboCup Rescue competitions, as well as from rescue laboratories and other indoor locations. In this paper, we use a subset (45 images) of such data which we used for learning classes such as *box* (12 images), *stairs* (15 images) and *pitch/roll ramp* in a maze(18 images). Since it is difficult to comprehend the range image, the corresponding colour (RGB) image of the scene will be shown in the rest of the paper. For each class, different multi-view data are chosen. For example, Figure 5 shows one view of some of the examples in this research which previously were used for training *box*, *stair*s and *pitch/roll ramp* classes respectively.

3.2 Parameters

Distance Threshold SP and SNP are using the distance threshold parameter to limit the maximum acceptable distance of a point to the plane model. If the point is further, it will not be considered as an inlier for the plane. In the PCL tutorial, this value is set to 0.01 for SP. However, for SNP, this value is set to 0.03. Due to this difference, we use more than one value as the distance threshold in our experiments. For example, for experiments regarding SP, we configured four experiments by using 0.005, 0.01, 0.03 and 0.05 as distance thresholds.

Angle Threshold SNP also considers the surface normal of each point and employs a weight value to determine the surface normal influence. We used the suggested value by the PCL tutorial as 0.1. We also have used similar threshold (15) in our region-growing algorithm to accept or reject adding a point to the current plane. In other words, if the angle of the current plane and the candidate point is more than 15, the point can not be added to the region.

Minimum Region Size All methods use a value as the minimum size for the plane. If the number of points belonging to the region is less than this value, the region will be rejected. We employed 90 for this purpose.

3.3 Data Preparation

Before applying PCL plane segmentation algorithms, we must prepare our data. Since our range images have 640×480 pixels, we sub-sample them to 160×120, while we converting them to point clouds. All data and the result of experiments are available via http://rfarid.altervista.org/plane_seg_compare/index.html.

3.4 Evaluating SP

The first experiment set is based on using the first PCL plane segmentation algorithm, called as SP. We applied SP on our data four times by using 0.005, 0.01, 0.03 and 0.05 as distance thresholds.

Considering the number of planes Table 1 shows the total and average number of planes produced per each class. This table indicates that the number of planes is closer and more reliable using distance thresholds 0.03 and 0.05.

Visual quality of the output We defined four levels of segmentation quality as H, MH, ML and L indicating high, mid to high, mid to low and low respectively. We went through all visual results and scored them based on a human-manual segmentation expectation. Table 2 shows the percentage of the images per each distance threshold and segmentation quality level. It illustrates that although we get high segmentation quality around 2% and 11% of times for using distance threshold as 0.005 and 0.01, these thresholds cause less quality

Table 1: Total and average number of planes using SP

	Sum				**Average**			
	Distance Threshold				**Distance Threshold**			
Class	**0.005**	**0.01**	**0.03**	**0.05**	**0.005**	**0.01**	**0.03**	**0.05**
box	182	108	67	60	15.17	9	5.58	5
pitch/roll ramp	493	281	144	116	27.39	15.61	8	6.44
stairs	329	226	124	84	21.93	15.07	8.27	5.6
Total	1004	615	335	260				

Table 2: Distribution of segmentation quality using SP

Distance Threshold	Segmentation quality level			
	H	**MH**	**ML**	**L**
0.005	2.22%	11.11%	8.89%	77.78%
0.01	11.11%	6.67%	53.33%	28.89%
0.03	0.00%	33.33%	62.22%	4.44%
0.05	0.00%	24.44%	71.11%	4.44%

Table 3: Distribution of segmentation quality per object class using SP for distance threshold as 0.03

	Seg. quality level			
Class	**H**	**MH**	**ML**	**L**
box	0%	83%	17%	0%
pitch/roll ramp	0%	28%	61%	11%
stairs	0%	0%	100%	0%

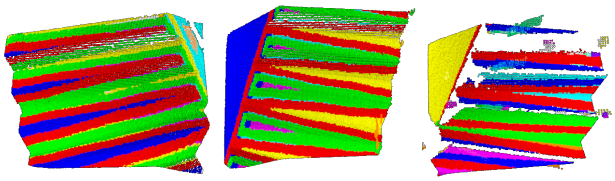


Figure 6: Example of SP segmentation result for *stairs* using dis. thr. as 0.03

level of segmentation. In contrast, using threshold as 0.03 and 0.05 produces results with the mid to low and mid to high level of segmentation quality.

Table 3 shows the same distribution per class while we used 0.03 as distance threshold. It indicates that the 83% of images containing *box* class are segmented with the mid to high level of segmentation quality, while 61% of *pitch/roll ramp* images have mid to low level of segmentation quality and all the *stairs* class images are segmented with a mid to low level of segmentation quality. Figure 6 shows three examples of *stairs* using SP segmentation (with distance threshold as 0.03) corresponding to the scenes shown in Figure 5b. Each plane is coloured differently. All the segmentation results are available in the experiment website.

3.5 Evaluating SNP

The second experiment set is based on using the second PCL plane segmentation algorithm, called as SNP, which employs normal vectors in its process. Since using 0.005 as distance threshold caused a major low quality segmentation for SP, we avoided using 0.005 and applied SNP on our data three times by using 0.01, 0.03 and 0.05 as distance threshold values.

Considering the number of planes Table 4 shows the total number and average number of planes produced using SNP per each class. This table indicates that the number of planes are closer and more reliable using distance thresholds as 0.03 and 0.05.

Visual quality of the output For analysing the visual quality of the output, we used the same approach employed for SP. Table 5 shows the percentage

Table 4: Total and average number of planes using SNP

	Sum			**Average**		
	Distance Threshold					
Class	**0.01**	**0.03**	**0.05**	**0.01**	**0.03**	**0.05**
box	235	98	89	19.58	8.17	7.42
pitch/roll ramp	456	269	177	25.33	14.94	9.83
stairs	174	205	191	12.43	13.67	12.73
Total	865	572	457	19.22	12.71	10.16

Table 5: Distribution of segmentation quality using SNP

Distance	Segmentation quality level			
Threshold	**H**	**MH**	**ML**	**L**
0.01	0.0%	0.0%	2.2%	97.8%
0.03	0.0%	51.1%	48.9%	0.00%
0.05	4.4%	80.0%	15.6%	0.00%

Table 6: Distribution of segmentation quality per class using SNP

Dist. Thr.=0.05	Seg. quality level			
Class	**H**	**MH**	**ML**	**L**
box	8.3%	75.0%	16.7%	0.00%
pitch/roll ramp	0.0%	72.2%	27.8%	0.00%
stairs	6.7%	93.3%	0.0%	0.00%

of the images per each distance threshold and segmentation quality level. It illustrates that the distance threshold set at 0.05 produces more mid to high quality segmentation.

Table 6 shows the same distribution per each class while we used 0.05 as distance threshold. It indicates that for each class images, the majority of segmentation quality are mid to high. The visual comparison between SP and SNP results shows that SNP outperforms SP. Figure 7 shows the corresponding version of Figure 6 using SNP segmentation (with distance threshold as 0.05), where each plane is coloured differently. All the segmentation results are available in the experiment website.

3.6 Comparing SNP and Our Method

We applied our region-growing plane segmentation algorithm on the same data. As shown before, SNP outperformed SP, so we just compared the result of segmentations between ours and SNP (using distance threshold 0.05) as follows:

Considering the number of planes The average number of planes is 9.44 for ours while this average is 10.16 for SNP.

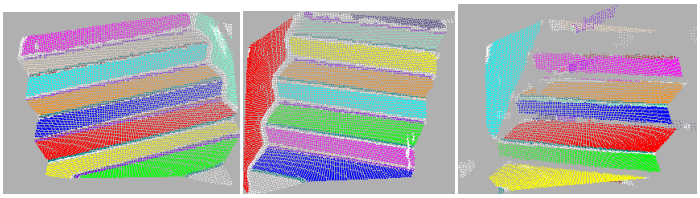


Figure 7: Example of SNP segmentation result for *stairs* using dis. thr. as 0.05

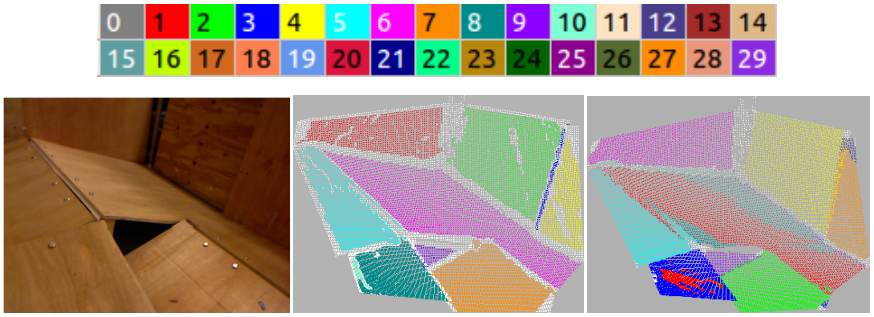


Figure 8: Colour legend for segmentation, RGB version of the scene and results of segmentations using our algorithm (left) and SNP (right)

Visual quality of the output To compare the results visually, we split a score 100 between the result of our segmentation method and SNP. We scored SNP as 2138 totally, which means 47.51 on average. Additionally, we also asked some people to do the same. We provided a web-page[4] showing the RGB version of the scene and the results of segmentation for method 1 and 2. The participants did not know which method was which. They were asked to split the score 100 between the two methods based on their expectations of the correct manual segmentation. SNP was scored 46.86 on average, while our algorithm was scored 53.14 on average. This comparison shows that the our segmentation algorithm outperforms the SNP visually.

Comparison based on the quality of the features Visual comparison might not be good enough to compare two segmentation methods. Since the result of segmentation can be passed to a robot as features for action planning, it is important to evaluate the correctness of these features, which is not possible to do just by visual comparison. In this case, a plane can be represented by a point belonging to the plane, its normal vector and its boundaries. The boundary can be represented by a convex hull (Farid and Sammut, 2012b). That is where SNP fails. SNP uses RANSAC and produces planes that cover many sparse points. As a result, two set of points, which are very far from each other, are put together in the same plane, while there is no such planar surface in the reality. These virtual planes can interfere with robot action planning, since there is no planar surface where the robot expects one based on the features provided. Figure 8 shows an example of such situation. Using the colour legend provided, the figure shows that our segmentation produces 10 planes, while SNP produces 12 planes in which planes coloured as regions 1, 3, 8, 9 and 10 are sparse and the corresponding features will be problematic. Figure 9 shows another example based on the leftmost *stairs* instance in Figure 5b. SNP produces regions 8, 9, 12 and 13 by putting edges of steps together as planes, which cause trouble when the robot uses these planes for actions such as climbing.

[4] http://rfarid.altervista.org/plane_seg_compare/comp.html

Table 7: Distribution of sparse planes using SNP (distance threshold=0.05)

Number of Sparse Planes	Frequency
0	**5**
1	**13**
2	**9**
3	**9**
4	**7**
5	**2**

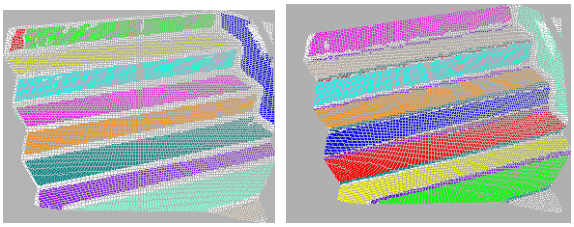


Figure 9: Segmented planes using our algorithm (left) and SNP (right)

Considering this sparse issue, we counted the number of sparse planes for the segmentation results by SNP. The details are available in the experiment website. Table 7 shows the numerical results of this evaluation and how the number of sparse regions per images are distributed. Based on these numbers, we can say 96 planes of total 457 planes for 45 images had this issue due to SNP segmentation. That is, there is an average of 2.13 planes per image affected by this issue.

Comparison on distance threshold As shown before, SP and SNP are sensitive to the value chosen for the distance threshold. Some images might work well with one value and another value might produces better results on another subset of images. SP and SNP do not suggest any systematic way to define the distance threshold. Our algorithm calculates the distance threshold based on each image automatically by using the minimum distance between each point and its adjacent neighbours and finding the average of them as the base. So, this distance threshold is also reliable in existence of noisy data. The detail and the relevant experiments are provided in Farid and Sammut (2014c).

4 Conclusion

Segmentation is an important step in robotics applications such as object classification and robot action planning. In this paper, we compared our region-growing plane segmentation algorithm with two state-of-the-art plane segmentation algorithms which are publicly available by PCL. We showed that the visual quality of our segmentation outperforms the others. We also showed that those RANSAC based segmentation algorithm can create planes with very sparse points which provide wrong information for robot action planning. We are planning to add our plane segmentation algorithm as a new method to PCL and/or ROS. The URL http://rfarid.altervista.org/plane_seg_compare/index.html provides the data and the detailed results.

Acknowledgement

We thank the people who kindly participated on visual comparison between our method and SNP.

References

A. Bartoli. A random sampling strategy for piecewise planar scene segmentation. *Computer Vision Image Understanding*, 105(1):42–59, 2007.

F. L. Endres. *Scene Analysis from Range Data*. Master thesis, Albert-Ludwigs-University Freiburg, Faculty of Applied Sciences, 2009.

R. Farid. *Generic 3D Object Recognition Using Multi-view Range Data*. PhD thesis, 2014. URL http://handle.unsw.edu.au/1959.4/53848.

R. Farid and C. Sammut. A relational approach to plane-based object categorisation. In *RSS 2012 workshop on RGB-D: Advanced Reasoning with Depth Cameras*, 2012a.

R. Farid and C. Sammut. Plane-based object categorisation using relational learning. In *Online Proc. of ILP2012*, 2012b. URL http://ida.felk.cvut.cz/ilp2012/wp-content/uploads/ilp2012_submission_6.pdf.

R. Farid and C. Sammut. Plane-based object categorisation using relational learning. *Machine Learning*, 94(1):1–21, 2014a. doi:10.1007/s10994-013-5352-9.

R. Farid and C. Sammut. Region-based object categorisation using relational learning. In D.-N. Pham and S.-B. Park, editors, *PRICAI 2014: Trends in Artificial Intelligence*, volume 8862 of *Lecture Notes in Computer Science*, pages 357–369. Springer International Publishing, 2014b. ISBN 978-3-319-13559-5. doi:10.1007/978-3-319-13560-1˙29.

R. Farid and C. Sammut. Plane-based object categorisation using relational learning: Implementation details and extension of experiments. Technical Report UNSW-CSE-TR-201416, School of Computer Science and Engineering, The University of New South Wales, 2014c. URL ftp://ftp.cse.unsw.edu.au/pub/doc/papers/UNSW/201416.pdf.

M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981. doi:10.1145/358669.358692.

S. Gächter, V. Nguyen, and R. Siegwart. Results on range image segmentation for service robots. In *Proc. of IEEE Int. Conf. on Computer Vision Systems*, pages 53–53, 2006. doi:10.1109/ICVS.2006.54.

D. Hegazy and J. Denzler. Generic 3D object recognition from time-of-flight images using boosted combined shape features. In A. Ranchordas and H. Araújo, editors, *Proc. of the 4th Int. Conf. on Computer Vision, Theory and Applications*, volume 2, pages 321–326. INSTICC Press, 2009.

A. Kalantari, E. Mihankhah, and S. A. A. Moosavian. Safe autonomous stair climbing for a tracked mobile robot using a kinematics based controller. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM2009)*, pages 1891–1896, 2009. doi:10.1109/AIM.2009.5229765.

M. McGill, R. Salleh, T. Wiley, A. Ratter, R. Farid, C. Sammut, and A. Milstein. Virtual reconstruction using an autonomous robot. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, pages 1–8, 2012. doi:10.1109/IPIN.2012.6418851.

R. Mohr, L. Morin, and E. Grosso. Relative positioning with uncalibrated cameras. In J. L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, pages 440–460. MIT Press, Cambridge, MA, USA, 1992. ISBN 0-262-13285-0. URL http://dl.acm.org/citation.cfm?id=153634.153656.

NIST. The national institute of standards and technology; test methods. Retrieved 2014-02-14, 2010. URL http://www.nist.gov/el/isd/test-methods.cfm.

A. Opelt. *Generic Object Recognition*. PhD thesis, Graz University of Technology, 2006.

J. Prankl, M. Zillich, and M. Vincze. Interactive object modelling based on piecewise planar surface patches. *Computer Vision and Image Understanding*, 117(6):718–731, 2013. ISSN 1077-3142. doi:10.1016/j.cviu.2013.01.010.

R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *Proc. of ICRA2011*, pages 1–4, 2011. doi:10.1109/ICRA.2011.5980567.

M. Shanahan. A logical account of perception incorporating feedback and expectation. In *Proc. of 8th Int. Conf. on Principles of Knowledge Representation and Reasoning*, pages 3–13, Toulouse, France, 2002. Morgan Kaufmann.

M. Shanahan and D. Randell. A logic-based formulation of active visual perception. In D. Dubois, C. A. Welty, and M.-A. Williams, editors, *Proc. of KR2004*, pages 64–72. AAAI Press, 2004.

J. Shin. *Parts-Based Object Classification for Range Images*. PhD thesis, Swiss Federal Institute of Technology Zurich, 2008.

A. Srinivasan. The Aleph Manual (Version 4 and above). Technical report, University of Oxford, 2002.

S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart. Cognitive maps for mobile robots–an object based approach. *Robotics and Autonomous Systems (From Sensors to Human Spatial Concepts)*, 55(5):359–371, 2007. doi:10.1016/j.robot.2006.12.008.

M. Xu and M. Petrou. 3D scene interpretation by combining probability theory and logic: The tower of knowledge. *Computer Vision and Image Understanding*, 115(11):1581–1596, 2011. doi:10.1016/j.cviu.2011.08.001.