

Explicit 3D Change Detection using Ray-Tracing in Spherical Coordinates

J. P. Underwood¹, D. Gillsjö², T. Bailey¹ and V. Vlaskine¹

Abstract—Change detection is important for autonomous perception systems that operate in dynamic environments. Mapping and tracking components commonly handle two ends of the dynamic spectrum: stationarity and rapid motion. This paper presents a fast algorithm for 3D change detection from LIDAR or equivalent optical range sensors, that can operate from arbitrary viewpoints and can detect fast and slow dynamics. Distinct from prior work, the method explicitly detects changes in the world, and suppresses apparent changes in the data due to exploration at frontiers or behind occlusions. Comprehensive experimentation is performed to assess the performance in several application domains. Sample data and source code are provided.

I. INTRODUCTION

Autonomous perception systems designed for dynamic environments require an explicit mechanism to reason about change over time. Consider the following motivating examples from four distinct application domains: (1) an indoor domestic or office environment in which people are tracked, furniture and walls are mapped, but furniture is moved from day to day; (2) an outdoor urban environment in which roads and buildings are mapped, people and cars are tracked, but parked cars change configuration daily and buildings yearly; (3) an indoor or outdoor security scenario in which a threat may be detected as a subtle but persistent change in the environment due to the placement of a briefcase or roadside bomb; and (4) a mining environment with the objective to actively change the landscape over timescales spanning days to decades.

Perception systems typically use a *mapping* component to represent static spatial structure and a *tracking* component to represent dynamic elements. This is a practical division due to differing requirements. Mapping requires large volumes of data to represent entire scenes, but does not require frequent updates. Tracking needs only to represent moving objects, but must do so continuously, on a short timescale relative to their motion. The combination is effective for environments with short timescale dynamics, however, changes that occur over longer timescales are usually not modelled well by

This work is supported in part by the Australian Centre for Field Robotics (ACFR), by the Rio Tinto Centre for Mine Automation and by the Agency for Defense Development of Korea (Grant#: U111071F)

¹J. P. Underwood, T. Bailey and V. Vlaskine are with the ACFR, School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney NSW 2006, Australia j.underwood, t.bailey and v.vlaskine at acfr.usyd.edu.au

²D. Gillsjö is with the Institutionen för Systemteknik, Department of Electrical Engineering, Linköpings Universitet

Data and code at <http://www.acfr.usyd.edu.au/papers/icra13-underwood-changedetection.shtml>

either component. The examples above are realistic applications in which mapping and tracking do not span the requisite time scales, motivating the need for explicit change detection strategies.

This paper focusses on 3D range data, with experimentation relating to the scenarios above. An algorithm is presented that can rapidly and explicitly detect geometric scene change from measurements taken from arbitrary overlapping vantage points, with arbitrary temporal separation.

II. BACKGROUND

Several methods for change detection using 3D sensor data exist in the literature. Motivated by security applications, [1] uses a Gaussian Mixture Model (GMM) to align 3D scans, compress them into a compact representation and detect change. The representation is bounded in size by the environment, not the increasing number of scans. Given a prior GMM, change is detected by measuring the Mahalanobis distance of new data points to the nearest Gaussian. Their work uses colour information from a visual camera to assist in detecting change. GMMs are also used in [3] and [9], replacing Mahalanobis distance by a maximum clique search between two GMMs in the former and by the Earth Mover's Distance [12] in the latter, though it is unclear whether either method improves over the original.

In [15] it is argued that a GMM may not appropriately model 3D structure, because the implicit Gaussian primitives “scantly fit the data”. Though theoretically reasonable, empirically there are many successful applications of GMM clustering to represent point cloud data (e.g. for incremental modelling [10], natural scene classification [7] and surface intercepts [11]). As an alternative to clustering, [15] adopts a Gaussian kernel to form a continuous 3D density function. Changes in density are expressed as a boolean operation over the representations after thresholding. Similar logical differencing is performed in [13] for excavation applications, using a discrete voxel occupancy grid instead of the continuous functions in [15].

Motivated by disaster response scenarios, [6] provides a Bayesian method for 3D change detection from aerial point cloud data. After non-rigidly aligning two point clouds data sets, an observation model for the second is derived by predicting ranges to the first. Predictions are modelled with Gaussian error in the distance from the sensor locations in the second set to the nearest point neighbours in the first data set. Bayes rule is applied to estimate the probability of environmental change.

Several aspects of the methods above are combined in [5] to provide a sophisticated and holistic approach to data alignment, representation and change detection, using a dense colour/depth camera (RGB-D sensor) in an indoor environment. Following the intuition that changes indoors are often due to the movement of whole objects, and that change is simpler to segment than static object geometry [14], change detection in [5] is motivated by its use as part of an unsupervised approach to identifying semantically significant objects. The method uses batch Simultaneous Localisation and Mapping (SLAM) for alignment and surfels for spatial modelling. The beam model from [6] is extended to model the likelihood of range returns along beams, similar to occupancy grids for sonar mapping [8].

The approaches above all implicitly assume that unmeasured space is unoccupied. As a result, none of these methods explicitly reason about the nature of occlusions and incomplete sensor coverage that is commonplace with practical sensing equipment. In this paper, an efficient approach to change detection is presented, using ray tracing to incorporate the information contained along sensor ‘beams’¹ and not only the beam end-points summarised by point clouds or representations derived from them, such as voxels, GMMs or surfels. Therefore the interplay between the vantage point of the sensor and the geometry of the scene is captured and regions of environmental change are categorically detected. This occurs despite the complex patterns of shadow caused by occlusions and incomplete sensor coverage. Emphasis is placed on detecting differences in the environment and not differences in the data.

The main concepts behind the change detection approach are presented in Section III, followed by the details of the algorithm in Section IV. Section V describes the experiments that were performed to verify the algorithm and quantify performance in simulation and with real data from two different sensors, in a cluttered indoor environment, an outdoor urban setting and on an operational open pit mine. Results are given in Section VI and Section VII concludes.

III. PRINCIPLES OF CHANGE DETECTION IN 3D

Changes are defined as differences in the 3D geometry of an environment at two different times. Examples of change cover all of the scenarios presented in Section I at various timescales; a person walking through a scene, repositioned table-top objects or furniture, demolition or construction of a building or the excavation of a mine or building site.

A. Dependence on Representation

The nature of geometric change and its measurement depends on the representation of the environment. A piecewise surface model may represent change by the differences in the surfaces [5], piecewise shape models such as GMMs express change as differences in shape components represented by

¹A beam refers to the free space light path inferred between a 3D data point and the sensor origin. It relates directly to the physics of LIDAR, but is also a reasonable approximation for other light based range sensor such as RGBD cameras and stereo vision.

eigenvalues [3], [9], and spatial density functions operate on changes in density [15]. Along these lines, much of the prior work bases change detection within the chosen representation, achieving significant data compression at the cost of assuming unobserved regions of the environment are the same as observed free space. Representations such as these could in theory be adapted to explicitly model free space and occlusion separately, however, this would significantly affect their ability to compress the raw data, which is one of their main benefits.

B. Change from Raw Observations

By contrast, in this work change is identified directly from raw optical 3D sensor data, without the need to refer to representations or models of the world. Increasing computation and storage capacity enables realistic methods such as pose SLAM to maintain large records of raw observation data, which then facilitates separate reasoning about free and unmeasured space, leading to explicit detection of geometric change. This work focusses on scanning LIDAR sensors, but is applicable to any form of data that indicates free space between the sensor and a surface intercept. In theory, this includes active sensors such as LIDAR, RADAR and structured light sensors (e.g. RGB-D) and passive range sensors such as stereo vision, however, experimentation with sensors other than LIDAR remains future work. The remainder of this section describes how information pertaining to geometric change can be extracted from laser measurements.

C. Sensor and Observation Model

Figure 1 shows a simplified laser beam model. This single datum informs about two things: 1) the location of a boundary between free and occupied space and 2) that free space is likely within the cone defined by the sensor origin and the beam divergence up to the boundary. Point cloud models or representations derived from them ignore the latter. When multiple beams have been measured at different times, changes can be detected wherever the information overlaps. Figure 2 illustrates the concept for a situation where a simple environment has changed, and a noise free sensor has been repositioned between two scans. It is clear that for even the simplest environments, complex patterns of information are formed due to restricted sensor apertures and shadows from occlusion. Noise from real sensors further complicates the differentiation of occlusion and free space. Figure 2(c) shows the region of overlap in which change can be explicitly detected. A small portion of the change is missed, but larger differences in the raw scan data (due to occlusion) are not falsely detected as change. If the environment is sampled more completely, all of the genuine change can be found without being disturbed by misleading differences in the data.

Environmental changes may be further categorised as *additions*, meaning a region that was free and is now occupied as seen on the left side of Figure 2(c), *subtractions* meaning a region that was occupied and is now free (right side of Figure 2(c)) or *discrepancies*, meaning an apparent change caused by sensor noise or other sources of error.

Change is detected by evaluating all overlapping regions of space between two sets of beams measured at different times, and potentially from different vantage points. Additions, subtractions and discrepancies are detected when a beam's end-point in one set is found to overlap a beam's free space cone in the other. The specific categorisation is determined by the temporal order of the scans: additions when the end-point occurred after the free cone, subtractions by symmetry and discrepancies if both beams are concurrent or cover a period where no change should occur. The next section presents an algorithm to detect end-points in free space cones.

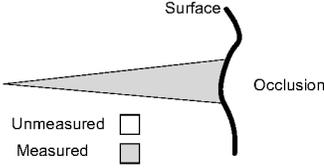


Fig. 1. A simplified laser beam model. A single beam informs about the conical volume of space defined by the light path up to the surface intercept.

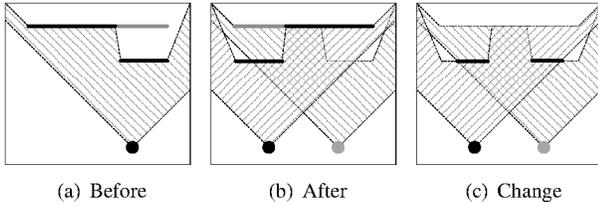


Fig. 2. A simple scene is scanned in (a) by a sensor placed at the black dot. The shaded region represents the scan coverage, which is the union of individual conical beam volumes shown in grey in Fig. 1. The black/grey line shows the observed/unseen structure. The small foreground ledge moves from right to left and the scene is scanned again from a different perspective (b). Change can be explicitly detected only in the crosshatched intersection of the two scans. The detectable change is drawn in black in (c).

IV. CHANGE DETECTION IN SPHERICAL COORDINATES

Given two data sets \mathbf{D}^A and \mathbf{D}^B containing LIDAR beams at times t^A and t^B respectively, an algorithm is required to find all end points in \mathbf{D}^B that lie within the conical free space volumes defined by \mathbf{D}^A . With arbitrarily distributed start and end points, the search for overlapping beams would require a complete iteration of \mathbf{D}^A for every end point in \mathbf{D}^B , resulting in $\mathcal{O}(|\mathbf{D}^A| \times |\mathbf{D}^B|)$ computational complexity.

If restricted to volumetric sensors, where each set of beams (or scan) can be considered to originate from a single location, then the complexity is significantly reduced to $\mathcal{O}(|\mathbf{D}^A| + |\mathbf{D}^B|)$ by projection into spherical coordinates to allow for $\mathcal{O}(1)$ grid based neighbourhood search in 'A' (for continuous lookup, not discretisation). The order of computation is constant with respect to the number of points in a scan. For situations where change is detected between many scans, computation scales linearly in the number of comparisons.

A. Representation

A scan 'A' is represented by one global 6D pose vector $\mathbf{p}^A = (x, y, z, roll, pitch, yaw)$ and a set of 3D spherical end points $\mathbf{S}^A = \{\mathbf{s}_1^A, \dots, \mathbf{s}_N^A\}$ with $\mathbf{s}_i^A = (\theta_i, \phi_i, r_i)$, corresponding to azimuth, elevation and range (or depth) respectively. Summarised by $\mathbf{D}^A = (\mathbf{p}^A, \mathbf{S}^A)$, this format is a generalisation of a 2D range image, for irregular or unstructured 3D scan patterns. The original free space cones in cartesian space are approximated by circular regions in (θ, ϕ) , though a warping exists between the equator and the poles of this space.

B. Detection Algorithm

The proposed change detection method is summarised by Algorithm 1. Each end point in \mathbf{D}^B is first transformed to the cartesian reference frame of \mathbf{D}^A and converted to spherical coordinates in line 2. All spherical neighbours from \mathbf{D}^A within an angular radius of T_a are retrieved in line 3. The azimuth wraps around in the grid for continuity at $\theta = \pm\pi$. Changes are flagged when the two conditions on line 4 are both met.

Algorithm 1: ChangeDetection

```

input :  $\mathbf{D}^A, \mathbf{D}^B, T_a, T_r$ 
output:  $\mathbf{C}^B = \{i_1, \dots, i_M\}, i \in \{1, \dots, |\mathbf{D}^B|\}$ 
          indices of changed end points in  $\mathbf{D}^B$ 

for every point in  $\mathbf{D}^B$ 
1 for  $i \leftarrow 1$  to  $|\mathbf{D}^B|$  do
    transform to spherical frame of  $\mathbf{D}^A$ , get neighbours
     $\mathbf{s}_i^{AB} \leftarrow \text{transform}(\mathbf{s}_i^B, \mathbf{p}^B, \mathbf{p}^A)$ 
     $\mathbf{S}_{near}^A \leftarrow \text{neighbours}(\mathbf{S}^A, \mathbf{s}_i^{AB}, T_r)$ 
    if  $r_i^{AB} < \min(r \in \mathbf{S}_{near}^A) - T_r$  and
        $\text{isbound}(\mathbf{s}_i^{AB}, \mathbf{S}_{near}^A)$  then
         $\mathbf{C}^B \leftarrow \mathbf{C}^B \cup i$ 
    end
2 end

```

C. Conditions for Detection

Algorithm 1 defines change by the two conditions on line 4:

- The range of the transformed point \mathbf{s}_i^{AB} (originally from \mathbf{D}^B) is less than the *minimum* range to all neighbours \mathbf{S}_{near}^A in \mathbf{D}^A , by more than the threshold T_r .
- The point \mathbf{s}_i^{AB} is inside the convex hull of neighbours \mathbf{S}_{near}^A in (θ, ϕ) .²

1) *Condition 1*: A shorter range indicates that a free space region in \mathbf{D}^A contains a boundary surface in \mathbf{D}^B . Using the *min* operator on neighbouring ranges is conservative and allows some resilience to small errors in scan alignment or calibration. Furthermore, it provides an implicit adaptive threshold to handle surfaces that are observed from different

²In practice it is sufficient to check that there are neighbouring spherical coordinates above, below, left and right of the point.

grazing angles, which result in varying range gradients in spherical coordinates. This was overcome explicitly in [5] by calculating surface normals, which is not required here. The approach is similar to the maximum likelihood smoothing proposed by [4].

Using the minimum range models the reference scan as a piece-wise constant range surface. The model is less sophisticated than alternative approaches [6], [5] but is several orders of magnitude cheaper computationally, and the results in Section VI show good performance.

2) *Condition 2*: The second condition is required to handle shadows caused either by incomplete scan coverage or missing ‘holes’ due to non-returns. In both cases there is no corresponding foreground surface in the data to explain the shadow, so the first condition may incorrectly return true. The second condition prevents false positives in this case.

D. Temporal Context

Algorithm 1 detects end points in one scan that violate the free space regions defined by another. From the timing of the scans the following external reasoning is applied:

$$\begin{aligned} t^A &= t^B \rightarrow \text{change} = \text{discrepancy} \\ t^A &< t^B \rightarrow \text{change} = \text{addition} \\ t^A &> t^B \rightarrow \text{change} = \text{subtraction} \end{aligned}$$

The inequalities may be loosely interpreted depending on the context. For example, in a mine environment, scans taken 1 hour apart may be considered contemporary, where only daily differences are considered to imply meaningful change. Given a pair of scans, Algorithm 1 is called twice, swapping the order of data to find additions and subtractions by symmetry.

V. EXPERIMENTAL DESIGN

This section describes the experiments that were conducted to quantify the performance of the algorithm. Four sets of data were used (Table I), with simulated data and real data from two different LIDAR sensors, in indoor laboratory and outdoor urban and open pit mine environments.

A. Simulated

The simulated data were produced from a model of a room with a box. Virtual LIDAR scans were generated with 1 degree resolution in θ and ϕ with full spherical coverage. Eight scans were synthesised from four sensor and two box positions, as illustrated in Fig 3. Points were automatically labelled as box or background for ground truth. No noise was added, in order to test the validity of the algorithm in perfect conditions.

B. Laboratory

Velodyne LIDAR measurements were taken in the ACFR Field Laboratory in controlled conditions. A box of size $14 \times 17 \times 40$ cm was placed in four different locations, after hours with no other background movement. No modifications were made to the cluttered lab, which contains chairs and tables, a kitchen space, packing crates, robots and assorted

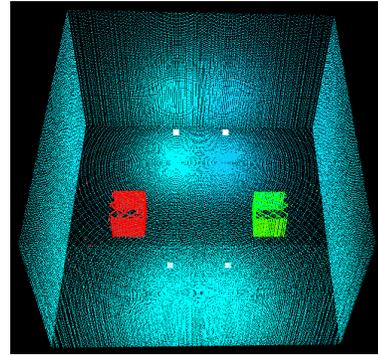


Fig. 3. Simulated change detection. Scans are synthesised from four poses (white), before and after the inner box is moved from left to right. The result of the change detection algorithm is shown for optimal $T_a = 1.4^\circ$, $T_r = 0.15m$ (TableII, line 1). Blue=raw data, Green=added, Red=removed.

electronic and mechanical fabrication equipment. The four box positions included two below the sensor height (similar to the ‘table top’ viewpoints used in [5]), one at approximately equal height and one above the sensor. For each box configuration, two scans were acquired from each of three different sensor poses, yielding 24 individual scans, which were aligned using iterative closest point (ICP). The box points were labelled manually for ground truth. The data are illustrated in Fig. 4.

All measurements from glass windows were labelled separately and specifically discounted, because these are not viewpoint invariant with LIDAR and therefore difficult to experimentally control. Glass reflections accounted for only 0.45% of the data, but are worth removing as the small box accounted for only 0.19%.

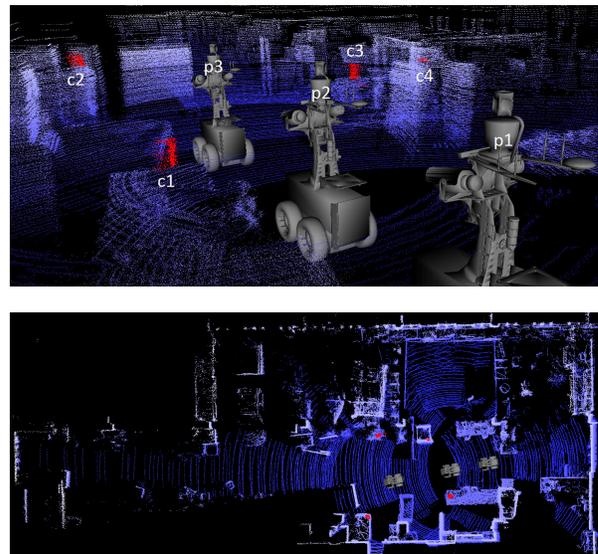


Fig. 4. Laboratory change detection, with scans of all permutations of three poses and four box configurations. Perspective (top) & plan view (bottom). Results of the change detection algorithm are shown for optimal $T_a = 1.2^\circ$, $T_r = 0.10m$, using multiple pairings and clustering (Table II, line 11). Blue=raw data, Red=changed. Note that changes are summarised in one colour because all boxes are added *and* removed from each location in succession. Only the box base is seen at c4 due to sensor aperture.

TABLE I

DATA SETS FOR EVALUATION.

Name	Sensor	Res(θ, ϕ) ^o	Environment	Size (m)	Movement	Configs ¹	Poses ¹	
Sim	Simulated	(1.00,1.00)	Regular	Empty room with box	100×100×100	Moved box	2	4
Lab	Velodyne HDL64ES2	(0.32,0.38)	Irregular ²	Cluttered indoor lab	107×79×7	Moved box	4	3
Carpark	Velodyne HDL64ES2	(0.32,0.38)	Irregular ²	Open outdoor carpark	115×86×14	Moved car & opened garage door	2	3
Mine	Riegl LMSZ420	(0.03,0.03)	Regular	Open pit iron ore mine	2361×1255×225	Detonated ore bench	2	2

¹ Configs indicates the number of different geometric configurations of the scene, poses indicates distinct sensor locations.

² Irregular Velodyne scan pattern contains 64 discrete elevations and is continuous in azimuth. Resolutions are averaged per scan.

C. Carpark

The Carpark data set was produced similarly to the Lab data. The scene is bound by the Lab and another building, both with windows. It contained four cars, a shipping container and two trees. Between the two configurations of the scene, one car was moved and the garage door of the lab was opened (revealing part of the inside wall), see Fig. 5. The data were manually labelled and found to contain 0.17% glass and 1.87% of changed data. Compared to the Lab data, the ten fold increase in the ratio of changed/unchanged points is due to the larger size of objects that were moved.

In keeping with the intention of the algorithm, only the car and garage door were manually labelled as a change, and not the section of the lab that was revealed by opening the door. Unlike the Lab data, it was impossible to acquire all permutations of configuration and pose, because neither the car nor robot could be repositioned precisely. The following ‘Gray Code’ progression of pose and configuration was used to avoid repositioning: $(p1, c1)$, $(p1, c2)$, $(p2, c2)$, $(p3, c2)$.

D. Mine

The mine data were acquired over a scale of several square kilometres, using a tripod mounted Riegl scanning LIDAR at the West Angelas open pit mine in Western Australia. A scan of an active bench was acquired prior to a controlled explosion and a second scan was taken afterwards to detect the change to the terrain. It was logistically impossible to manually label data on this scale, however, areas of cavitation (subtraction) and expansion (addition) are expected given the controlled nature of the explosion. Only visual results are analysed.

E. Test Procedure

Algorithm 1 was run on every non-equal pair of scans in each data set. The output was compared to the ground truth to provide a performance measure under various conditions. All tests were run on a discrete set of parameter values (T_a, T_r) as an exhaustive search for optimality. For the Lab and Carpark data, two additional variations were tested:

1) *Clustering*: Change points are clustered by proximity as in [2] and filtered by size, for inexpensive noise rejection. This is a heuristic approach to the regularisation method in [5]. Randomly distributed sensor noise is initially identified as change (reasonably so), hence the purpose of this test is to confirm that higher accuracy can be obtained for changed *object* detection by post-filtering. The benefit is application dependent. For example, mapping applications may optimally remove all change and noise together without filtering,

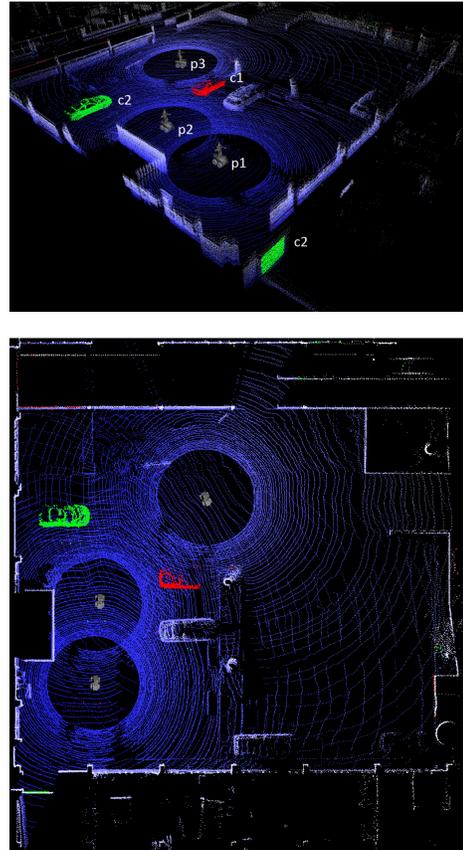


Fig. 5. Carpark change detection. For configuration $(c1)$, the garage door is open and the cars positioned as shown (red). In $(c2)$ the door is closed and the car has moved (green). The result of the change detection algorithm is shown for $T_a = 0.9^\circ$, $T_r = 0.20m$, using single pairings and no clustering (Table II, line 12). Blue=raw data, Green=Added, Red=Removed.

whereas object identification or tracking will benefit from clustering. For all tests, points are clustered by adjacency < 10 cm and filtered by cluster sizes > 40 points.

2) *Grouping*: Combinations of scans larger than pairs are used to identify change. A scan at one pose and configuration is compared to all scans of the same configuration from different poses. Change is identified as the union of the output of each pair-wise call to Algorithm 1. This is equivalent to the comparing a new scan to a set of prior scan data.

VI. RESULTS

Algorithm 1 has a binary output (change / no change) for each point. Subsequent temporal reasoning subdivides change into addition and subtraction, resulting in three classes. Due to the symmetry of Algorithm 1, only the

binary performance needs to be quantified, and f-score is used as an appropriately weighted combination of precision and recall values. All experimental results are presented in Table II. Note the accuracy is always high due to the dominant quantity of unchanged points, whereas the f-score is more discerning.

Subsets of data are evaluated to expose the effect of scene and algorithm configuration on performance.

A. Performance in Different Environments

The performance in simulation is nearly perfect, with an f-score of 0.99. As there is no simulated noise, imperfections are mainly due to false negatives at the object's base, due to ambiguity at the ground intersection. Disambiguation is a function of sensor resolution compared to object size.

The lab environment is a challenging test because a noisy sensor is used in a cluttered environment, to detect an intentionally small object. Given a percentage per scan of change due to noise, performance increases with object size. E.g. if the box were doubled in size, the true positives would approximately double, but the false positives would only increase slightly. This environment has the lowest f-score of 0.71.

The carpark environment has a higher performance than the lab, and is similar to the sim ($f = 0.91$). This is in part due to the larger object size and the reduced clutter and geometric complexity.

B. Stationary or Movable Sensor

Comparing equal, non-equal and all pose subsets reveals significantly better performance with a stationary sensor. Focusing on tests without clustering clearly shows that false positives and negatives are both increased when the sensor is moved. A stationary sensor can better handle biased errors due to imperfect calibration, alignment, and LIDAR artifacts at object boundaries. The optimal parameters (T_a, T_r in Table II) show that a stationary sensor affords a lower angular threshold in particular.

This reveals the effectiveness for security / surveillance applications with a fixed sensor, even in highly cluttered conditions.

C. Vantage Point

Objects in natural scenes are stacked vertically due to gravity. The foreground acts as a filter on the region of detectable change, thus the contrast and scope for detection is greatest when objects are viewed from above. For detection of whole objects [5] showed results from high vantage points. In the lab data, box configurations $c1$ and $c3$ from poses $p1$ and $p3$ have a high viewing angle, and isolated tests show significantly increased performance for this subset, suggesting a sampling strategy for whole object detection applications in particular.

D. Clustering for Noise Rejection

Genuine geometric scene changes yield spatially clustered change points, whereas changes due to noise are scattered

randomly. Therefore, clustering and filtering reassigns false positives as true negatives, without significantly affecting the other statistics. This is evident in Table II for the lab and carpark, and the rejection of noise results in significantly higher performance. Clustering is appropriate wherever object level reasoning is required. This may be improved further by adopting the more sophisticated regularisation from [5], which is outside the scope of this work. The clustering algorithm [2] adds negligible time (see Table II), whereas regularisation increases the computational complexity significantly.

E. Pairs and Larger Combinations

It has been established that the vantage point is an important consideration for change detection. One strategy is to adopt an explicit sampling policy for active change detection, however, this requires higher level knowledge and reasoning about where change is expected. Another approach is to detect change in one scan by comparison with all other available scans from different vantage points. For this case, Table II (lines 4/10 and 14/18) shows only a small increase in performance for the lab (0.64 to 0.69) and carpark (0.90 to 0.92), because the additional vantage points cause more true *and* false positives (due to noise). When we add clustering to filter the noise, the performance is the highest of all tests.

F. Mine Data

The changes detected in the mine are shown in Fig. 6, for manually specified parameters $T_a = 0.05^\circ, T_r = 0.10m$ chosen to match the sensor characteristics (see Table I). Point-wise ground truth labels are not available, but the scans were intentionally taken before and after an intentional change to the mine geometry. The ground has sunk at the location of the explosion and a wave of crushed rock has washed over the face. Dust has stirred and vehicles have arrived. All of these features are correctly highlighted as additions and subtractions by the algorithm, with sparse background noise.

As a proof of concept, the algorithm is appropriate for long range terrain change detection in a mining or excavation context.

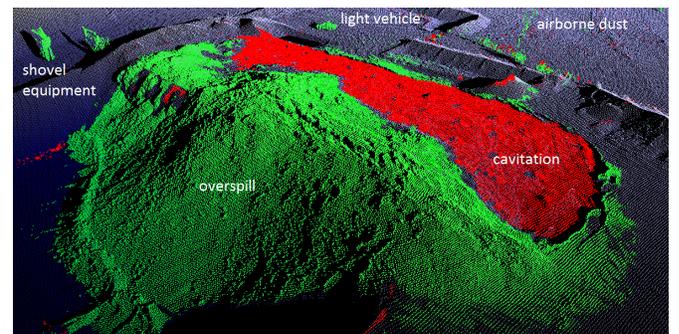


Fig. 6. Mine change detection. Two long range LIDAR scans measure the West Angelas mine before after a controlled explosion. Detected change is shown for $T_a = 0.05^\circ, T_r = 0.10m$: Blue=raw data, Green=added, Red=removed.

TABLE II

STATISTICAL RESULTS, OPTIMAL PARAMETERS AND PROCESSING TIMES

Test	Data set	Variant	Scan subset	true+	true-	false+	false-	precis.	recall	accur.	f-score	T_a^o	T_r (m)	t(s)
1 Fig. 3	Sim	\bar{c} , pairs	all	10579	958702	12	302	0.9989	0.9722	0.9997	0.9854	1.4	0.150	0.40
2	Lab	\bar{c} , pairs	all	5776	4359741	1928	2795	0.7497	0.6739	0.9989	0.7098	1.2	0.250	0.39
3			$p1=p2$	3447	2180935	425	791	0.8902	0.8134	0.9994	0.8501	0.7	0.200	0.43
4			$p1 \neq p2$	2751	2178858	1521	1582	0.6440	0.6349	0.9986	0.6394	1.3	0.250	0.38
5			$p=\{2,3\}, c=\{1,3\}$	466	121345	96	37	0.8292	0.9264	0.9989	0.8751	1.3	0.200	0.41
6			c , pairs	all	5961	4361816	148	2610	0.9758	0.6955	0.9994	0.8121	0.8	0.100
7	Lab	\bar{c} , multi	$p1=p2$	3244	2181350	22	994	0.9933	0.7655	0.9995	0.8646	0.7	0.025	0.46
8			$p1 \neq p2$	2850	2180544	48	1483	0.9834	0.6577	0.9993	0.7883	1.0	0.100	0.44
9			$p=\{2,3\}, c=\{1,3\}$	487	121439	4	16	0.9919	0.9682	0.9998	0.9799	1.2	0.150	0.39
10			$p1 \neq pi$	2906	2170702	1322	1327	0.6873	0.6865	0.9988	0.6869	1.5	0.200	0.77
11 Fig. 4	Lab	c , multi	$p1 \neq pi$	3199	2171978	46	1034	0.9858	0.7557	0.9995	0.8556	1.2	0.100	0.86
12 Fig. 5	Carpark	\bar{c} , pairs	all	4998	364103	110	890	0.9785	0.8488	0.9973	0.9091	0.9	0.200	0.41
13			$p1=p2$	2023	121560	13	177	0.9936	0.9195	0.9985	0.9551	0.5	0.200	0.50
14	Carpark	c , pairs	$p1 \neq p2$	3130	242533	106	558	0.9672	0.8487	0.9973	0.9041	0.9	0.200	0.41
15			all	4993	364220	4	895	0.9992	0.8480	0.9976	0.9174	0.7	0.100	0.44
16			$p1=p2$	2028	121570	4	172	0.9980	0.9218	0.9986	0.9584	0.5	0.025	0.50
17	Carpark	\bar{c} , multi	$p1 \neq p2$	3107	242650	0	581	1.0000	0.8425	0.9976	0.9145	1.0	0.100	0.41
18			$p1 \neq pi$	581	61279	37	71	0.9401	0.8911	0.9983	0.9150	1.0	0.300	0.82
19	Carpark	c , multi	$p1 \neq pi$	621	61316	0	31	1.0000	0.9525	0.9995	0.9756	1.2	0.025	0.93
20	Mine	\bar{c} , single	all	121029	5430092	-	-	-	-	-	-	0.05	0.1	19.09

¹ Variants 1&2 from Section V-E: (1) c/\bar{c} denotes clustering is used / not used. (2) Single pairs are used, or multiple pairs are combined.

² Scan subsets: $p1=p2$ or $p1 \neq p2$ denotes pose constancy, $p=\{2,3\}, c=\{1,3\}$ selects subsets where the sensor looks down on objects from above.

³ Statistical measures: (true+, true-, false+, false-, precis., recall, accur., f-score): True/false positives and negatives, precision, recall, accuracy, f-score.

⁴ Optimal parameters: Angle and range thresholds (T_a, T_r).

⁵ Processing time per scan. Laptop with an Intel Core i7-2640M CPU at 2.79GHz, Windows 7 Enterprise, Matlab R2011b / Visual Studio 2010. This is the sum of two calls to Algorithm 1 for pairs and 1 call per pose for multi, plus clustering time if used.

VII. CONCLUSION

This work presents an efficient and effective algorithm for change detection that is appropriate to a class of range sensing commonly used in robotics and more generally in mapping, surveillance and perception systems. The algorithm is versatile, with a wide range of applications and environments. A set of experiments was conducted to support these claims and expose the algorithm's performance in different conditions.

ACKNOWLEDGEMENT

Thanks to Alastair Quadros for aligning the data, to Peter Morton for the simulation and to Mark Calleija, Vsevolod Vlaskine and Cedric Wohlleber for continuous technical development and support.

REFERENCES

- [1] H. Andreasson, M. Magnusson, and A. Lilienthal. Has something changed here? autonomous difference detection for security patrol robots. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3429–3435, 29 2007–nov. 2 2007.
- [2] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. On the segmentation of 3D lidar point clouds. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2798–2805, may 2011.
- [3] P. Drews, P. Nu and andez, R. Rocha, M. Campos, and J. Dias. Novelty detection and 3D shape retrieval using superquadrics and multi-scale sampling for autonomous mobile robots. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3635–3640, may 2010.
- [4] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 755–762, june 2010.
- [5] E. Herbst, P. Henry, Xiaofeng Ren, and D. Fox. Toward object discovery and modeling via 3-d scene comparison. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2623–2629, may 2011.
- [6] Ralf Kaestner, Sebastian Thrun, Michael Montemerlo, and Matt Whalley. A non-rigid approach to scan alignment and change detection using range sensor data. In Peter Corke and Salah Sukkariah, editors, *Field and Service Robotics*, volume 25 of *Springer Tracts in Advanced Robotics*, pages 179–194. Springer Berlin / Heidelberg, 2006.
- [7] Jean-Francois Lalonde, Nicolas Vandapel, Daniel F. Huber, and Martial Hebert. Natural terrain classification using three-dimensional lidar data for ground robot mobility. *Journal of Field Robotics*, 23(10):839–861, 2006.
- [8] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 116–121, mar 1985.
- [9] P. Nu and andez, P. Drews, A. Bandera, R. Rocha, M. Campos, and J. Dias. Change detection in 3D environments based on gaussian mixture model and robust structural matching for autonomous robotic applications. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2633–2638, oct. 2010.
- [10] F. Pauling, M. Bosse, and R. Zlot. Automatic segmentation of 3D laser point clouds by ellipsoidal region growing. In *Australasian Conference on Robotics and Automation*, 2009.
- [11] A. Quadros, J.P. Underwood, and B. Douillard. An occlusion-aware feature for range images. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4428–4435, may 2012.
- [12] Y. Rubner, C. Tomasi, and L.J. Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59–66, jan 1998.
- [13] J. Ryde and N. Hillier. Alignment and 3D scene change detection for segmentation in autonomous earth moving. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1484–1490, may 2011.
- [14] C. Schutz and H. Hugli. Change detection in range imaging for 3D scene segmentation. *Europto, Besancon, SPIE*, 2786:98–106, 1996.
- [15] A.W. Vieira, P.L.J. Drews, and M.F.M. Campos. Efficient change detection in 3D environment for autonomous surveillance robots based on implicit volume. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2999–3004, may 2012.