



Enabling Situational Awareness via Augmented Reality of Autonomous Robot-Based Environmental Change Detection

Christopher Reardon^{1(✉)}, Jason Gregory¹, Carlos Nieto-Granda²,
and John G. Rogers¹

¹ U.S. Army Research Laboratory, Adelphi, MD 20783, USA

{christopher.m.reardon3.civ,jason.m.gregory1.civ,
john.g.rogers59.civ}@mail.mil

² University of California, San Diego, San Diego, CA 92093, USA
cnietogr@ucsd.edu

Abstract. Accurately detecting changes in one’s environment is an important ability for many application domains, but can be challenging for humans. Autonomous robots can easily be made to autonomously detect metric changes in the environment, but unlike humans, understanding context can be challenging for robots. We present a novel system that uses an autonomous robot performing point cloud-based change detection to facilitate information-gathering tasks and provides enhanced situational awareness. The robotic system communicates detected changes via augmented reality to a human teammate for evaluation. We present results from a fielded system using two differently-equipped robots to examine implementation questions of point cloud density and its effect on visualization of changes. Our results show that there are trade-offs between implementations that we believe will be constructive towards similar systems in the future.

Keywords: Human-robot teaming · Augmented reality · Simultaneous Localization and Mapping (SLAM) · Change detection · Field robotics

1 Introduction

The real world is an endlessly dynamic and varying place. The ability to detect changes in an environment is important both for intelligent robots to operate in the real world and for robots to operate alongside humans as teammates.

Robots with minimal capabilities may be able to operate using simple reactive or closed-loop control to perform a basic task. This is particularly true where efforts can be made to engineer and instrument the environment to simplify the task space. However, the increased task and environmental complexity present in real-world scenarios will require more sophisticated capabilities, including

This is a U.S. government work and not under copyright protection in the U.S.; foreign copyright protection may apply 2020

J. Y. C. Chen and G. Fragomeni (Eds.): HCII 2020, LNCS 12190, pp. 611–628, 2020.

https://doi.org/10.1007/978-3-030-49695-1_41

scene understanding so that a robot can reason about essential information such as the state of a task or the feasibility of goal completion [14]. Importantly, this mission-critical information can be deduced from detecting changes in the robot’s environment and contributes greatly to whether an autonomous robot can complete a given mission. Take, for example, a search and rescue application where a change may make the goal state unachievable, such as a planned route becoming blocked by rubble. In this case, not only will the autonomous robot not be able to navigate to its intended destination, but the safe execution of the entire mission may be in jeopardy if the change’s context indicates a high likelihood of additional adverse events, e.g., falling debris or adversarial activity.

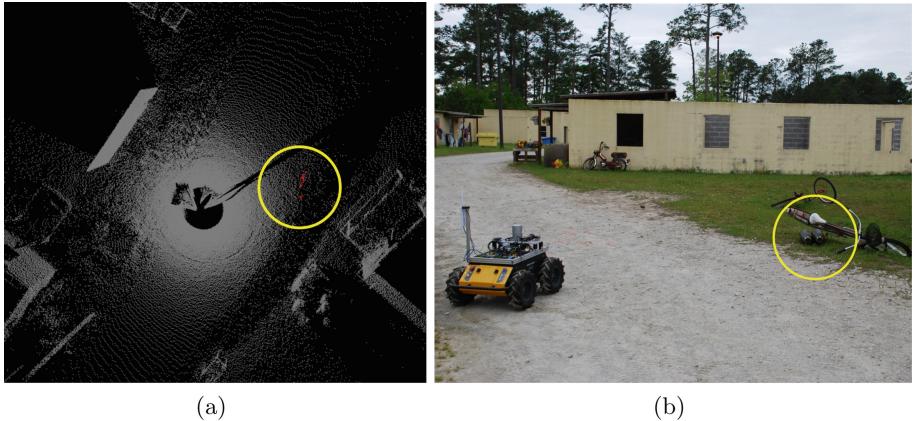


Fig. 1. Changes detected in the point cloud model (a) indicate potential locations of devices placed by an adversary (b).

In order for robots to operate alongside humans as teammates, the ability to communicate these detected changes and corresponding reasoning is also important for a variety of cooperative tasks. For example, in autonomous inspection of infrastructure, the ability to detect, identify, and communicate changes that represent structural deterioration to a human is essential for affecting timely repairs. An example in security robotics is the ability for a surveillance robot to detect changes along its regular patrol route that might indicate actions by adversarial agents, e.g., a break-in or the placement of a dangerous device, such as shown in Fig. 1, which a human teammate would want to be informed of immediately. In the context of cooperative tasks involving humans and autonomous robots, the relevant information must be exchanged between the teammates in a timely fashion for optimal and responsive decision making.

Detecting even large changes in complex environments can be challenging for humans [3, 13]. Mobile robots can autonomously perform metric-based comparisons of sensor readings to detect potential changes, but lack contextual understanding to determine their significance. In addition to the potential for innocuous change in any real world environment, algorithmic methods for robots to

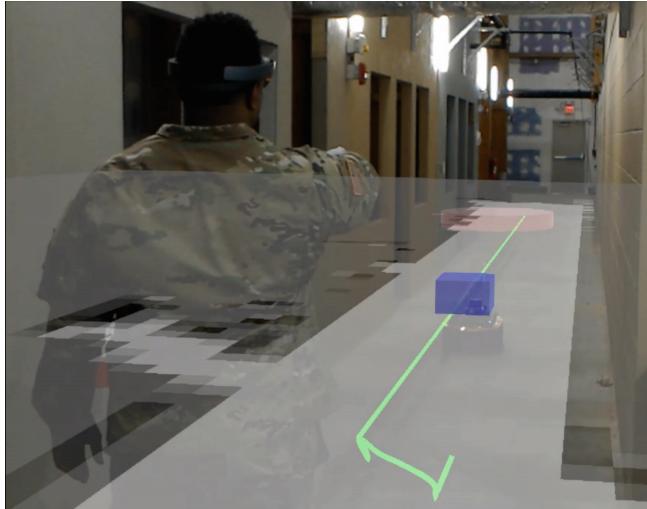


Fig. 2. Augmented reality situational awareness in human-robot teaming.

detect changes can be noisy and yield numerous false positives. Combined, these issues provide significant obstacles for autonomous robots to interpret and act upon detected changes. We believe that a human-robot team working cooperatively to detect, interpret, and act upon changes offers a powerful means to overcome their mutual shortcomings. Together, the human and robot possess the capabilities to detect and identify changes of importance in any specific scenario context; however, the critical challenge becomes creating an efficient method of communicating, interpreting, and prioritizing changes.

We present a novel approach to address this challenge that uses Augmented Reality (AR) to create a human-robot team where the robot identifies changes, communicates them via AR to the human teammate, who can then interpret their context for further action. Our system is intended to address the general case of detecting changes in an arbitrary environment without external instrumentation and presenting them to a human teammate using AR, building upon previous work in [10]. The AR system we employ is a head-mounted device (HMD) worn by the human teammate, who is co-located in the environment with the robot (Fig. 2). This allows the robot to present augmented visualizations via the HMD to provide situational awareness to the human teammate, which enables improved decision making and collaboration. We believe this is the first example of the use of AR for communicating and interpreting environmental changes detected by an autonomous robot to a human teammate.

Detection of environmental changes takes place on-board the mobile robot in real time (Fig. 1). A prior model is collected of the environment in a “clean”, initial state. This model consists of a point cloud together with an anchoring position and orientation, referred to as a pose, which is registered into a global reference frame. To compare the current state of the environment with this

model, a fine alignment is computed using generalized-ICP [12] with the robot’s pose from a Simultaneous Localization and Mapping (SLAM) solution as an initial guess. Points in the current scan which are further than the intrinsic sensor noise threshold from the model are clustered into candidate change regions. The candidate change regions of sufficient size are highlighted in the user’s AR interface for the human to evaluate for further action.

One important question when implementing a point cloud-based change detection system is: What is the appropriate sampling density with regards to system performance and user experience? To validate our approach and examine this question, we implement and test our system using two otherwise identical robots equipped with different commercial-off-the-shelf LiDAR devices that generate relatively sparse and dense point clouds. Our robots autonomously perform change detection and present changes online to the human teammate via the HMD interface.

We hypothesize (*H*₁) that the higher-density LiDAR would provide more accurate detection in all environments than the lower-density LiDAR. Given that expectation, we further hypothesize that (*H*₂) when the user is teamed with the higher-density LiDAR robot, the visual presentation to the user would be more discriminative, i.e., correct change detections would be more obvious.

To evaluate these hypotheses, we compare performance between the two robotic systems in two different field environments: in an alleyway street scene and an outdoor driveway with a parking space. Our results show that while the higher-resolution LiDAR does produce a denser point cloud and therefore more true positive detections, when evaluated in field environments there are a number of distinct trade-offs that mean the higher density is not always more accurate, nor does it always provide a better user experience. Full results are discussed in Sect. 5.

2 Background and Related Work

Novelty detection is a broad and robust area of research that generally means the recognition of elements in *test* data that differ from *training* data or a *model* learned from that data [7]. Environmental change detection can be seen as the application of novelty detection to tasks where physical changes in a specific environment, e.g., object addition or removal, are identified on an ongoing basis by comparing continuously reacquired *test* data against a known *model*.

Robots operating in real-world environments have a strong need for accurate change detection, particularly for tasks where one robot or teams of robots repeatedly encounter the same environment. This area of research has broad applications including inspection [5], surveillance [6, 18], safety and security [15], and general robust outdoor navigation [14].

Augmented and mixed reality technologies are currently experiencing a period of growth for use in human-robot interaction (HRI) as they present a mechanism for overcoming issues of communication in HRI [16]. Similarly, this work uses AR to overcome issues of communication and contextual understanding in HRI by creating more robust human-robot teams for operation in field

environments. Previous work by the authors presented an overview of examined applications in this domain [10].

3 Approach

3.1 SLAM

In our system, both the AR-HMD and the mobile robot construct independent 3D representations of the environment utilizing a Simultaneous Localization and Mapping (SLAM) algorithm. For the AR-HMD, this approach is based upon visual feature tracking and is provided as a black-box solution delivered with the interface (Microsoft HoloLens). The SLAM implementation utilized onboard the robot is based upon *OmniMapper* as described in [17] with further refinements described in [4]. Briefly, the approach is to build a *pose graph* over measurements between adjacent point clouds and loop closures when locations are revisited. These measurements are used to compute a solution to the robot's trajectory in a least squares sense via the nonlinear optimization framework *GTSAM* [1] based upon square root smoothing and mapping [2]. Each point cloud taken along this optimized trajectory solution is then projected into a common frame of reference and accumulated into a point cloud representing the environment.

Of course, using shared environmental information like change detections for teaming between the human and the robot is impossible without a common frame of reference. Alignment of the human and robot teammates' coordinate frames is therefore critical for understanding teammate position. We use the approach presented in previous work [9] to enable this capability. Since both the robot and the AR-HMD can generate a geometric representation of the environment in point cloud format, we can then compute the homogeneous transformation matrix between the robot and human point clouds using the Iterative Closest Point (ICP) algorithm [12]. The initial computation is performed on a coarse estimate provided by the human, and is then recomputed online as the human and robot maneuver through the environment.

3.2 Change Detection

To perform change detection, first a *model* cloud representing the “clean” state of the world is built via the SLAM process described in Sect. 3.1. At any time in the future, the robot can then collect a *test* cloud using the same procedure. This cloud can either be collected completely and then processed, or processed incrementally during collection.

Once a *test* cloud is created, either at the end of a patrol or incrementally online, it is analyzed for changes from the *model* cloud. These clouds are in approximately the same reference frame either through coarse GPS alignment or by originating the maps near the same place, as was done in this paper. Note, however, either of these approximate alignment methods is insufficient to support change detection due to the large errors that inherently result from

small rotational alignment errors. Therefore, the alignment of these clouds is first refined with a generalized ICP [12] procedure.

Once the *model* and *test* clouds are accurately aligned, change detection is implemented in PCL [11] via a set of difference segmentation functions and outlier filters. The difference segmentation routine builds a KD-tree of the model to reduce quadratic search complexity to $n \log n$. Each point in the test cloud is then compared with the model via the KD-tree to find the nearest point and its distance. If this distance is greater than a threshold, which in our experimentation is 10 cm, it is accumulated in a new point cloud denoted *change*.

The *change* cloud is then filtered to remove noisy detections by looking for support of at least 10 detection points within a radius of 30 cm. This will remove small isolated groups of detection points which might be due to range error in the sensor or quantization error in the representation used by the mapper. Other errors are possible due to occluded regions in the *model* cloud which happen to be visible due to slight viewpoint variance in the *test* cloud; these regions will be present in the *change* cloud and will lower the precision of the analyzed results in Sect. 4. The segmented changes can be seen in Fig. 1 for an example scenario where a device has been hidden under a bicycle and is detected by the robot system.

3.3 Augmented Reality Interface

Change detections described in Sect. 3.2 are continuously collected as the robot navigates through the environment. Filtered candidate changes are presented to the user via the AR-HMD as translucent red spheres with a radius of 4 cm. An example of changes detected and visualized in the AR-HMD is shown in Fig. 3, where the user can see the detected change locations superimposed over the physical changes in the environment. For the user's reference, the 2D occupancy grid generated by the robot's SLAM implementation (Sect. 3.1) is also visualized as a 2D projection onto the ground plane, with white representing unoccupied and black representing occupied space in the robot's map. Using this information, the user is able to evaluate and identify actual changes for future investigation. For example, the user could prioritize examining a suspicious package or removing debris blocking a road. The locations where visualizations coincide with movable objects are the most likely candidates for such changes.

Future work will be directed at refining the interface through testing different data aggregation, visualization, and interaction types through the HMD with an aim towards improving the interpretability and accuracy of the information displayed, as well as directing the robot to autonomously address changes that the human deems of interest.

4 Experiments

We tested our hypotheses from Sect. 1 through an evaluation of our approach using the complete system online in experiments in two different environments.

By examining the performance of the change detection and AR visualization by teaming a human wearing an AR-HMD with two otherwise identical robots equipped with different resolution LiDAR systems, we are able to reach several valuable conclusions regarding the development, applicability, and configuration of such systems.



(a)



(b)

Fig. 3. Augmented reality visualizations of changes (red spheres) in the two experimental environments: (a) alley and (b) driveway. Also shown are the robot's 2D occupancy grid map as white (unoccupied) and black (occupied) cells projected onto the ground plane, and the current location of the robot (blue box). (Color figure online)

4.1 Hardware

The hardware employed for these experiments included two Clearpath Robotics Jackal robots. This wheeled platform measures $0.508 \times 0.430 \times 0.250$ m and can

move at a maximum velocity of 2.0 m/s. Each have an Intel Core i5-4570TE CPU and runs Ubuntu 16.04 and the Robot Operating System (ROS) [8] on board. Each was equipped with a MicroStrain 3DM-GX4-25 inertial measurement unit (IMU) for improved mapping and state estimation performance, and a Ubiquiti Bullet M5 HP 5 GHz WiFi radio for communications.

Because both the change detection (Sect. 3.2) and corresponding visualization of those changes in AR (Sect. 3.3) are highly dependent upon the density of the point clouds collected by the system, we equipped each robot with a different Light Detection and Ranging (LiDAR) device. The first robot was equipped with a lower density Velodyne VLP-16 LiDAR sensor, which has 16 laser rangers oriented with a 1.9° elevation angle separation, a range of 100 m, and collects approximately 300,000 points per second in a 360° azimuthal field of view and a 30° elevational field of view. The second robot was equipped with a higher density Ouster OS1 with 64 laser rangers oriented with a 0.7° elevation angle separation and a range of 120 m that collects over 1.3 million points per second with 360° azimuthal and 45° elevational fields of view. Both robots are shown in Fig. 4.



Fig. 4. The two robots used in our experiments, which are equipped identically with the exception of Velodyne VLP-16 LiDAR (left) and Ouster OS1 LiDAR (right).

The human is equipped with the Microsoft HoloLens AR-HMD¹. Custom visualization messages are communicated between the robots and the HoloLens using a combination of ROSBridge² (on the robots) and ROS#³ (on the HoloLens).

¹ <https://www.microsoft.com/en-us/hololens>.

² http://wiki.ros.org/rosbridge_suite.

³ <https://github.com/siemens/ros-sharp>.

4.2 Environments

Two environments were used for these experiments. The first was an alleyway street scene constructed for the purpose of robotics experimentation that features a narrow alley space between two multi-story buildings seen in Fig. 5a. The second was an outdoor driveway with a parking space that is adjacent to trees and a building, as seen in Fig. 5c. Both presented unique features that excited the change detection system in different ways.

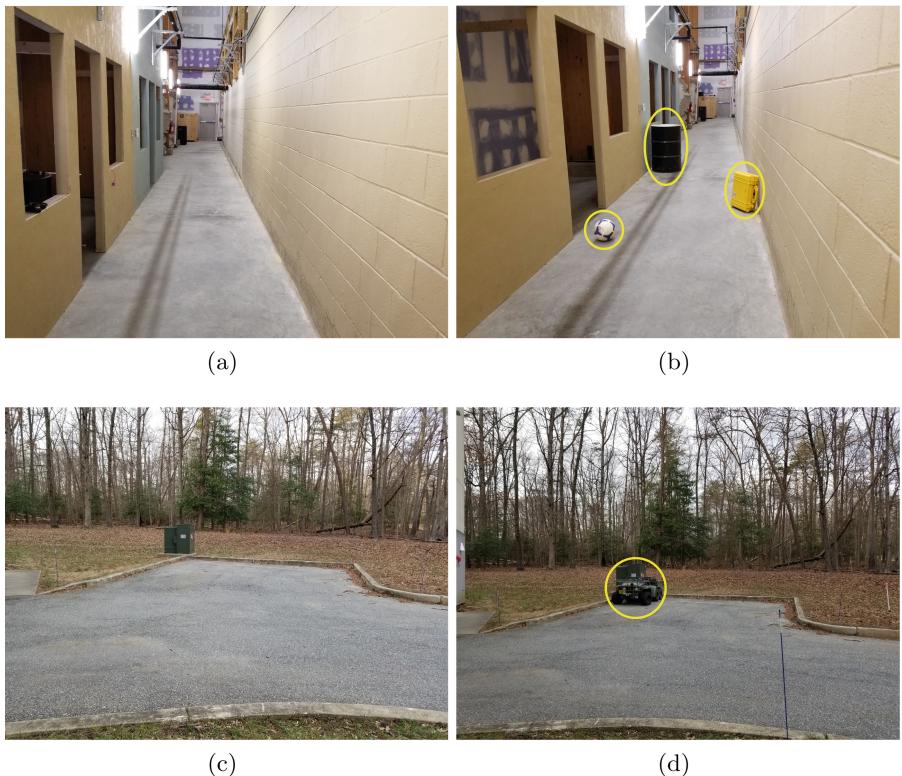


Fig. 5. Environments used in the experiments. (a) and (c) show the alley and driveway environments. (b) and (d) show environments with changes added. In (b), a ball, yellow case, and steel drum were added to the alley scene. In (d), a small All-Terrain Vehicle (ATV) was placed in a parking space. (Color figure online)

The *model* point clouds generated by each robot for each environment are shown in Fig. 6a–6c. The differing density of the clouds due to the different LiDAR resolutions can be clearly seen in the quantity of points and resulting appearance of fidelity.

4.3 Procedure

The procedure for each experiment was as follows. In each environment, following the approach in Sect. 3, a *model* point cloud was collected and stored.

Then, changes in the form of novel objects were placed in the environment that were not present in the model, as depicted in Figs. 5b and 5d. For the next phase, a robot re-explored the environment while collecting *test* clouds and evaluating them online against the *model* cloud. As changes were detected, visualizations were immediately displayed via the AR-HMD interface to a human who was co-present in the environment. *Model*, *test*, and the robots' change detection performance, as well as video from the AR-HMD user's experience were all recorded for analysis.

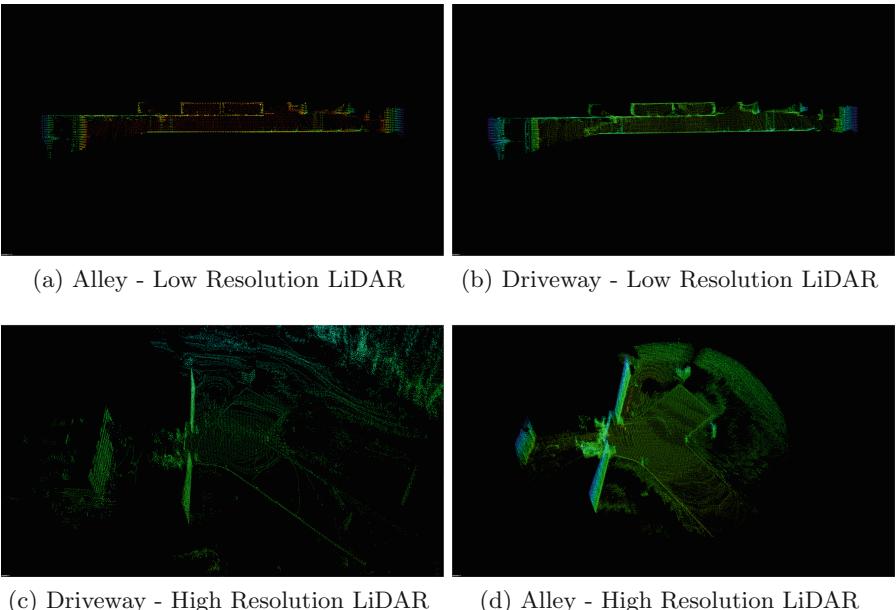


Fig. 6. *Model* point clouds of each environment for each LiDAR sensor type.

5 Results and Discussion

Recall that our initial hypotheses were:

H1: The robot equipped with the higher resolution LiDAR would provide more accurate detection than the robot with the low-density LiDAR.

H2: Visual presentation to the user will be more discriminative (correct change detections would be more obvious) with the higher resolution LiDAR.

Interestingly, the first hypothesis $H1$ did not hold entirely for either environment, given our assumptions, and for different reasons. First, it is worth noting that our change detection algorithm is far from perfect; it can at times produce a large number of false positives. These can be due to small errors and misalignments in the collection, noise in the data, or noise in the environment itself such as random motion effects like wind. In light of this, the density of the point cloud was actually detrimental to the statistical performance of the higher-resolution LiDAR. Where the low-resolution LiDAR would detect a few spurious point changes, the high-resolution would detect a large number. This effect can be seen clearly in the alley environment between Figs. 7a and 7b. Figures 9 and 10 show the actual point cloud detections over time; one can observe the relative number of false positives in particular. Further, even though the alley has many of the visual aspects of a real street scene, because it is part of a larger “mock” staged environment contained in large building it is not subject to environmental changes such as wind. The effect of wind noise on the system was highly

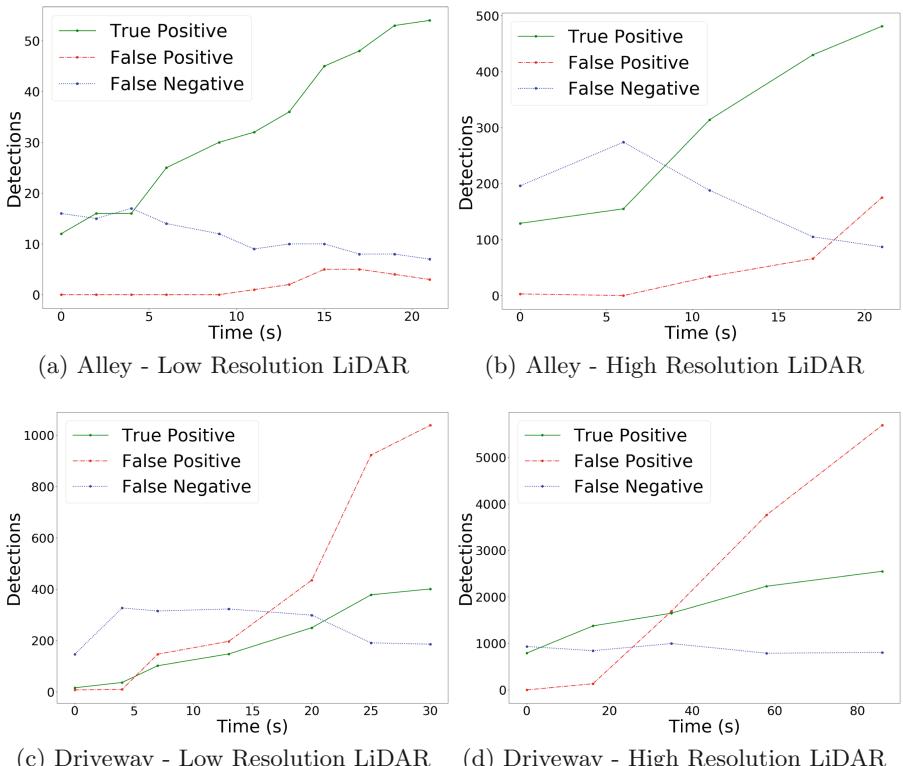


Fig. 7. True positive, false positive, and false negative detection results from the two robot configurations (lower and higher resolution LiDAR) in two environments (alley street scene and outdoor driveway with a parking space). Note the difference in detections scale.

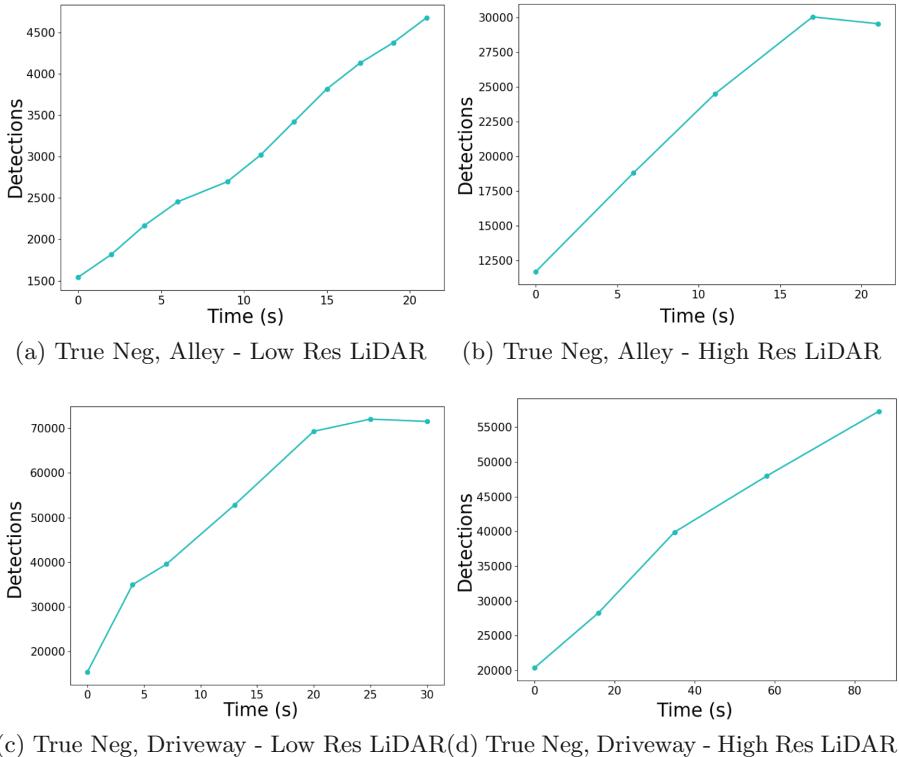


Fig. 8. True negative detection results from the two robot configurations (lower and higher resolution LiDAR) in two environments (alley street scene and outdoor driveway with a parking space). Presented separately from Fig. 7 due to relative magnitude vs. other detection types. Note the difference in detections scale.

pronounced for *both* robots in the driveway scene, as seen in the high number of false positives as time increased in Figs. 7c and 7d. The relative quantity of false positives from wind blowing on surrounding vegetation is shown best in Figs. 11d and 12d.

Despite the difficulty of noisy environments and the system's tendency to be sensitive to these small errors and produce a large number of false positives, we were pleased to find support for invalidating *H2* as well. While this work did not include a user study and therefore cannot be conclusive, anecdotally we found that despite relatively significant numbers of false positives visualized to the user, because of the tight clustering of change detection visualizations on true changes, the noisy data was easily filtered out by a human user. This is illustrated in viewpoints taken from the AR-HMD in Fig. 3, where despite false change detections to the left in Fig. 3a one can clearly see the changes clustered on the objects, and likewise despite false changes in the grass and trees in Fig. 3b, one's attention is immediately drawn to the changes indicated on the ATV.

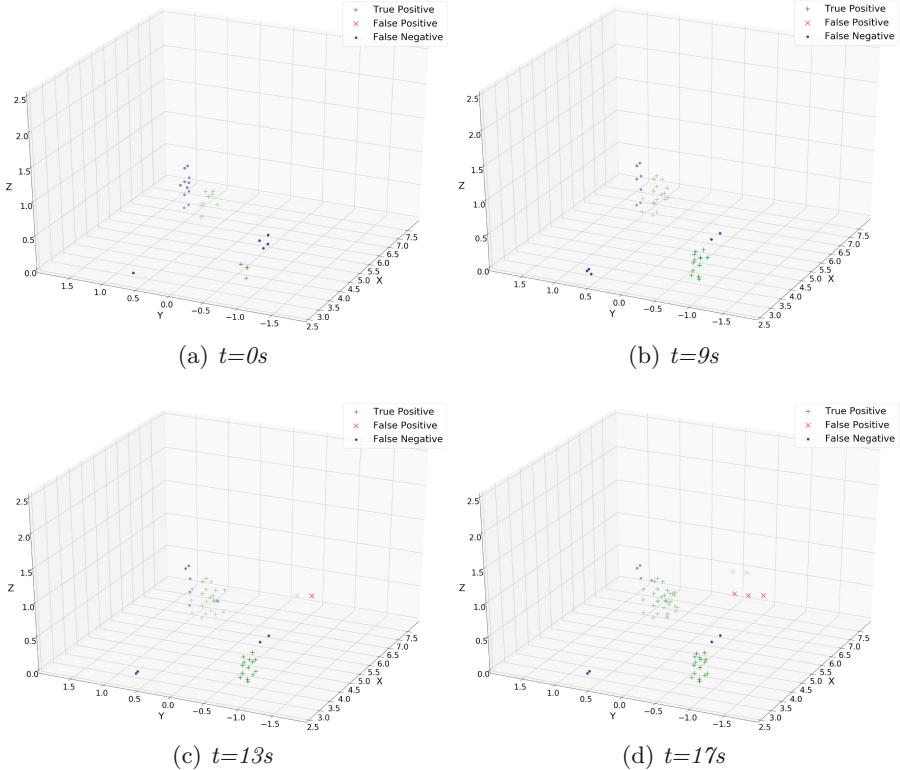


Fig. 9. Example results from the lower resolution LiDAR-equipped robot in the alley scene (Fig. 5b). Plots (a)–(d) spatial representations of true positive, false positive, and false negative detections at progressing time lapses. The robot begins at position $(2, 0, 0)$ and proceeds along the X axis. True positives on the right, centered around $(3.5, -0.5, 0.25)$, are change detections intersecting the yellow case in Fig. 5b. True positives on the left near $(6.5, 1, 0.5)$ are change detections intersecting the black drum. The ball in Fig. 5b was not detected.

With these results in mind, we can define a set of trade-offs and design decisions that we believe may be constructive towards further refinements of similar systems in the future. The LiDAR resolution had significant and somewhat unexpected trade-offs. As noted above, there was a significant magnifying effect on the false positive detections for the high-resolution LiDAR. Extensive outdoor environments where no barriers exist to LiDAR scans present a computational challenge to both robots. In the driveway environment, the processing time for the lower-resolution LiDAR robot was demonstrably longer than in the alley, resulting in about half as many *test* clouds being processed per unit time. This effect was much worse for the other robot, as the size of the point cloud on a high-resolution LiDAR like the Ouster OS1 makes the change detection algorithm, while not completely intractable, too slow to run on our robot's hardware in

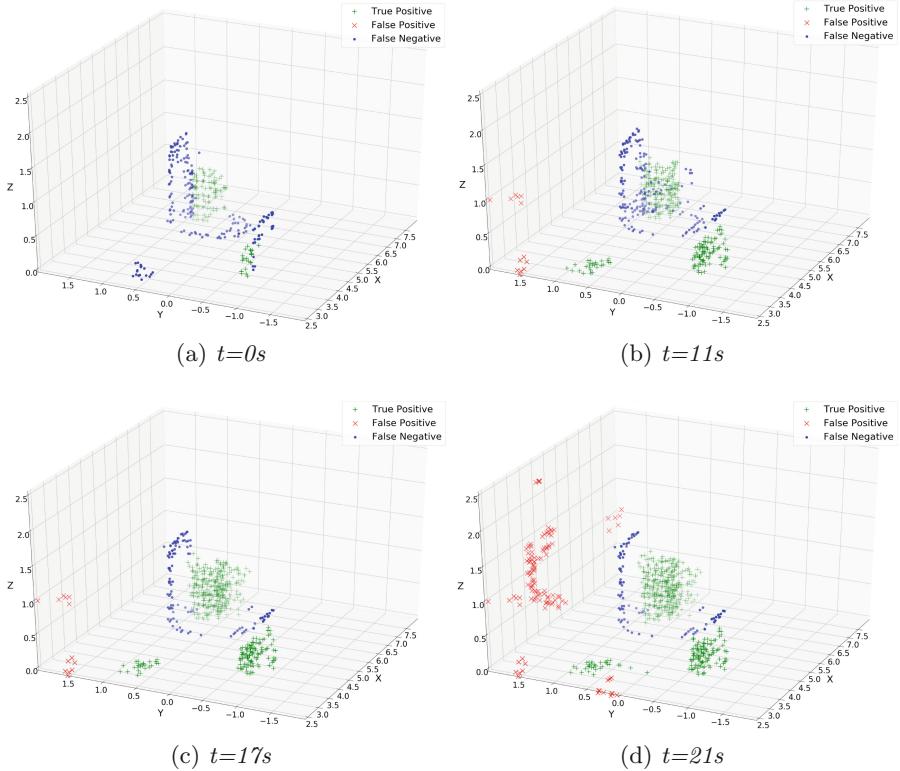


Fig. 10. Example results from the higher resolution LiDAR-equipped robot in the alley scene (Fig. 5b). Plots (a)–(d) show spatial representations of true positive, false positive, and false negative detections at progressing time lapses. The robot begins at position $(2, 0, 0)$ and proceeds along the X axis. True positives on the right, centered around $(3.5, -0.5, 0.25)$, are change detections intersecting the yellow case in Fig. 5b. True positives on the left near $(6.5, 1, 0.5)$ are change detections intersecting the black drum. True positives on the left near $(3, 0.75, 0.1)$ are detections intersecting the ball.

any reasonable timeframe. For this reason we applied a threshold to the LiDAR, limiting any laser return to 50 m in an effort to reduce the point cloud size. For a more fair comparison we made a corresponding change in the post-processing of the results from the low-resolution LiDAR robot, discarding detections over 50 m.

In terms of object detection, the higher-resolution LiDAR robot was the only robot to successfully detect the ball in the alley scene (Fig. 5b). For a visualization of that detection compare Fig. 9 with Fig. 10. Note the ball location at $(3, 0.75, 0.1)$. It is possible that with better tuning of the change detection the ball could be detected by the lower-resolution LiDAR; however, it is entirely possible that the low number of points returned from such a small object will always be filtered.

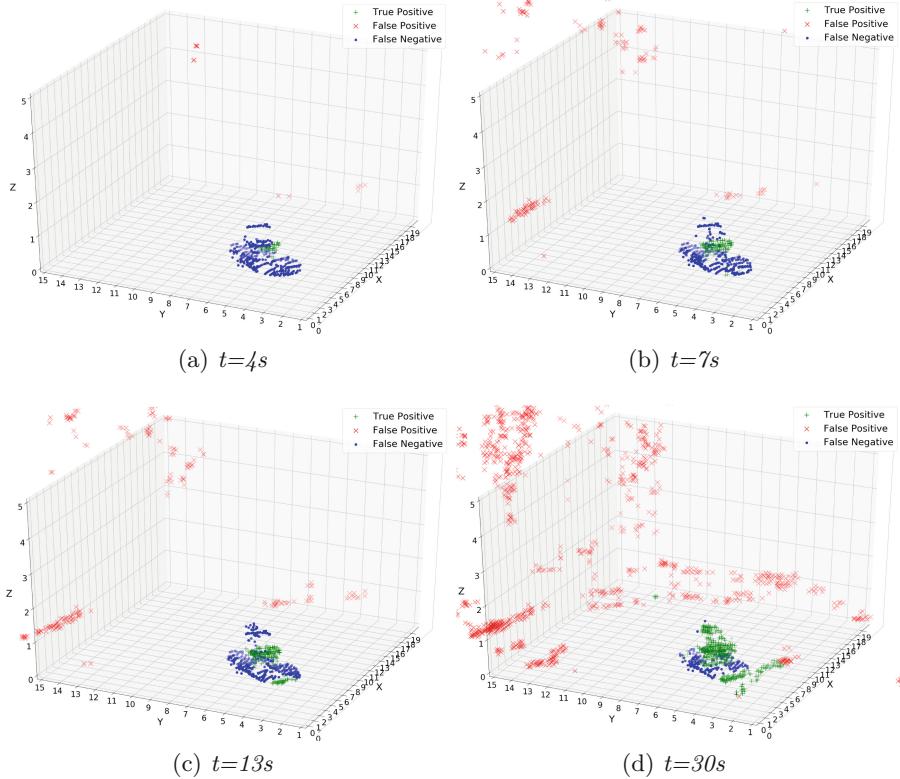


Fig. 11. Example results from the lower resolution LiDAR-equipped robot in the driveway scene (Fig. 5(d)). Plots (a)–(d) show spatial representations of true positive, false positive, and false negative detections at progressing time lapses. The robot begins at position (0, 0, 0) and proceeds along the X axis. True positives on the left near (7, 5, 0.5) are change detections intersecting the small ATV in Fig. 5(d). The numerous false positives in the space over 10 m from the origin coincide with vegetation in the scene blowing in the wind.

Finally, the purpose of this system is ultimately to present changes to the human for evaluation of further action. Given the concerns of tractability of computation, accuracy of detection, and sufficiency of information, and given our change detection approach, unless there is a need to detect small changes in the environment, we believe that a robot equipped lower resolution LiDAR may be capable of detecting and presenting changes to the human user via the AR-HMD just as well, if not better, than a robot equipped with a higher-resolution LiDAR. We caveat this with the expectation that given sufficient optimization for computation, detection accuracy, and information downsampling (e.g., through filtering and clustering) in the user interface, the system with a higher-resolution LiDAR should be able to be made to outperform the lower-resolution.

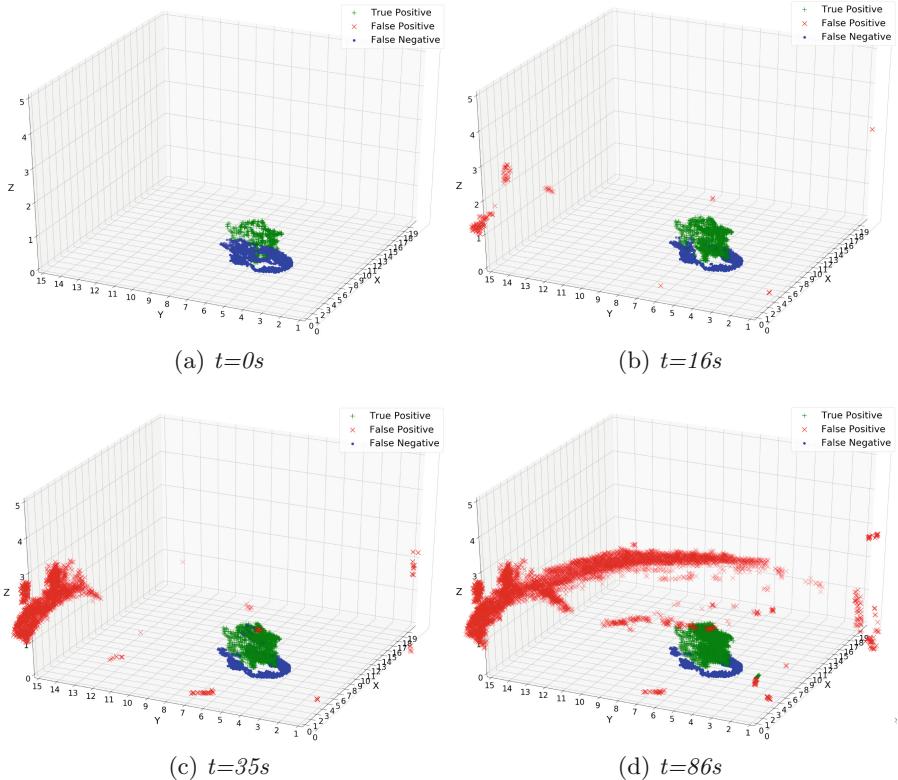


Fig. 12. Example results from the lower resolution LiDAR-equipped robot in the driveway scene (Fig. 5d). Plots (a)–(d) show spatial representations of true positive, false positive, and false negative detections at progressing time lapses. The robot begins at position (0, 0, 0) and proceeds along the X axis. True positives on the left near (7, 5, 0.5) are change detections intersecting the small ATV in Fig. 5d. The numerous false positives in the space over 10 m from the origin coincide with vegetation in the scene blowing in the wind.

6 Conclusion

In this paper we present an approach to provide situational awareness of change detections found by an autonomous robot to its human teammate. This approach is motivated by the complimentary observations that 1) change detection can be challenging for humans yet entirely tractable for properly equipped and programmed autonomous robots, and 2) understanding change context is easy for humans and challenging to implement on an autonomous robot. The approach presented compares observed *test* point clouds against an *a priori* collected *model* cloud to generate change detections. Our system enables the robot to communicate detected changes via AR visualizations to the human teammate for evaluation. To field such a system, an important consideration is the sampling density

of the point cloud. We implement our system on two otherwise identical robots with different resolution LiDAR sensors and examine two hypotheses about the system's change detection performance and visualization interpretability. Our results lead us to conclude that higher resolution is not simply better, and we identify several trade-offs between implementations. We observe that regardless of the implementation and although a full human study is out of scope of this paper, there is evidence that our approach is sufficient to provide situational awareness of changes in the environment to human teammates.

References

1. Dellaert, F.: Factor graphs and GTSAM: a hands-on introduction. Technical report, Georgia Institute of Technology (2012)
2. Dellaert, F., Kaess, M.: Square Root SAM: simultaneous localization and mapping via square root information smoothing. *Int. J. Robot. Res.* **25**(12), 1181–1203 (2006)
3. Durlach, P.J.: Change blindness and its implications for complex monitoring and control systems design and operator training. *Hum.-Comput. Interact.* **19**(4), 423–451 (2004)
4. Gregory, J., et al.: Application of multi-robot systems to disaster-relief scenarios with limited communication. In: Wettergreen, D.S., Barfoot, T.D. (eds.) *Field and Service Robotics. STAR*, vol. 113, pp. 639–653. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-27702-8_42
5. Marsland, S., Nehmzow, U., Shapiro, J.: On-line novelty detection for autonomous mobile robots. *Robot. Auton. Syst.* **51**(2–3), 191–206 (2005)
6. Núñez, P., Drews, P., Rocha, R., Campos, M., Dias, J.: Novelty detection and 3D shape retrieval based on Gaussian mixture models for autonomous surveillance robotics. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4724–4730. IEEE (2009)
7. Pimentel, M.A., Clifton, D.A., Clifton, L., Tarassenko, L.: A review of novelty detection. *Sig. Process.* **99**, 215–249 (2014)
8. Quigley, M., Faust, J., Foote, T., Leibs, J.: ROS: an open-source Robot Operating System. In: *ICRA Workshop on Open Source Software* (2009)
9. Reardon, C., Lee, K., Fink, J.: Come see this! Augmented reality to enable human-robot cooperative search. In: *Proceedings of the 2018 IEEE Symposium on Safety, Security, and Rescue Robotics* (2018)
10. Reardon, C., Lee, K., Rogers, J.G., Fink, J.: Augmented reality for human-robot teaming in field environments. In: Chen, J.Y.C., Fragomeni, G. (eds.) *HCII 2019. LNCS*, vol. 11575, pp. 79–92. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21565-1_6
11. Rusu, R.B., Cousins, S.: 3D is here: point cloud library PCL. In: *2011 IEEE International Conference on Robotics and Automation*, pp. 1–4. IEEE (2011)
12. Segal, A., Haehnel, D., Thrun, S.: Generalized-ICP. In: *Robotics: Science and Systems*, vol. 2, p. 435 (2009)
13. Simons, D.J., Chabris, C.F.: Gorillas in our midst: sustained inattentional blindness for dynamic events. *Perception* **28**(9), 1059–1074 (1999)
14. Sofman, B., Neuman, B., Stentz, A., Bagnell, J.A.: Anytime online novelty and change detection for mobile robots. *J. Field Robot.* **28**(4), 589–618 (2011)

15. Sturari, M., Paolanti, M., Frontoni, E., Mancini, A., Zingaretti, P.: Robotic platform for deep change detection for rail safety and security. In: 2017 European Conference on Mobile Robots (ECMR), pp. 1–6. IEEE (2017)
16. Szafir, D.: Mediating human-robot interactions with virtual, augmented, and mixed reality. In: Chen, J.Y.C., Fragomeni, G. (eds.) HCII 2019. LNCS, vol. 11575, pp. 124–149. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21565-1_9
17. Trevor, A.J.B., Rogers, J.G., Christensen, H.I.: OmniMapper: a modular multi-modal mapping framework. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 1983–1990, May 2014. <https://doi.org/10.1109/ICRA.2014.6907122>
18. Vieira, A.W., Drews, P.L., Campos, M.F.: Spatial density patterns for efficient change detection in 3D environment for autonomous surveillance robots. IEEE Trans. Autom. Sci. Eng. **11**(3), 766–774 (2014)