

ARMY RESEARCH LABORATORY



Have I Been Here Before? A Method for Detecting Loop Closure With LiDAR

by John G Rogers III and Jason M Gregory

ARL-TR-7165

January 2015

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Adelphi, MD 20783-1138

ARL-TR-7165

January 2015

**Have I Been Here Before? A Method for Detecting Loop
Closure With LiDAR**

**John G Rogers III and Jason M Gregory
Computational and Information Sciences Directorate, ARL**

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) January 2015		2. REPORT TYPE Final		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Have I Been Here Before? A Method for Detecting Loop Closure With LiDAR				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) John G Rogers III and Jason M Gregory				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: RDRL-CII-A 2800 Powder Mill Road Adelphi, MD 20783-1138				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-7165	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Loop closure detection has progressed to operate in real time in areas as massive as an entire city. This impressive result is enabled by the richness and uniqueness of visual stimuli in a city. If a robot were required instead to operate in a deep underground complex of tunnels that is absent of unique features, then a new set of tools is needed to enable a similar level of performance. We present a unified representation for recognizing global loop closure, which incorporates <i>appearance</i> - and <i>spatial</i> -based techniques on a mobile robot system, which has the unfortunate task of exploring a system of austere underground tunnels with only a laser scanner as a guide.					
15. SUBJECT TERMS Mapping, SLAM, simultaneous localization and mapping, localization, global loop closure, robot system					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 26	19a. NAME OF RESPONSIBLE PERSON John G Rogers III
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 301-394-1811

Contents

List of Figures	iv
1. Introduction	1
2. Approach	2
2.1 Mapping System.....	2
2.2 Appearance Model	2
2.3 Spatial Model	3
2.4 Unified Model	4
2.5 Training	5
3. Experimental Design	5
4. Results	9
5. Conclusion and Future Work	13
6. References	15
List of Symbols, Abbreviations, and Acronyms	16
Distribution List	17

List of Figures

Fig. 1	The map generated from a run where a robot drives approximately 1 km along a road (on the left), then enters an underground facility (thin tunnels on the right), reemerges from the exit at the top, and reenters the first entrance near the middle. The robot has accumulated sufficient pose error during this 3-km trip that loop closure will be needed to correct it. The error can be seen in the zoomed in segment in the lower right.	2
Fig. 2	The iRobot PackBot and Velodyne HDL-32E LiDAR used in this work.	6
Fig. 3	A satellite photograph of the urban testing facility with each of the five routes labeled. Loop 1 is highlighted in cyan, loop 2 in orange, loop 3 in red, loop 4 in light green, and loop 5 in purple. Each loop was driven twice so that the loop closure method could be evaluated for the respective run.	7
Fig. 4	A photograph of the near-featureless underground tunnel environment.	7
Fig. 5	The point cloud merge utility puts together adjacent point clouds into higher resolution places suitable for feature extraction. Four point clouds (top row) are combined into one place (bottom row). Four point clouds are shown for clarity; 10 are used in practice.	8
Fig. 6	A visualization of the ground-truthing utility. The leftmost point cloud is taken from the current pose of the robot. The point clouds on the right are three sequential point clouds that can be chosen if any match the location of the point cloud on the left. The user can iterate through all of the previous point clouds and choose the best match.	9
Fig. 7	Loop closure performance on all runs in the urban test complex, with three options for which components are used in proposing loop closures. a) Precision/recall graph: Loop closure is very sensitive to false positives and must be run at 100% precision. <i>Appearance</i> + <i>Spatial</i> can operate together at around 10% recall without sacrificing precision. Either technique alone has a significantly lower recall. b) ROC graph relating false and true positive rates.	10
Fig. 8	Loop closure found in run 5 from Fig. 3 using the <i>Appearance</i> and <i>Spatial</i> models. a) Loop closure detected by <i>Appearance</i> and <i>Spatial</i> model: the left image corresponds to immediately before loop closure is inserted and the right image corresponds to after. b) Two candidate places shown in upper half of image have their relative poses computed through sampling initial conditions and running GICP; the resulting fused point cloud is shown in lower half of image. c) Detail view of loop closure.	11
Fig. 9	Loop closure performance on a run in the underground tunnel complex, with three options used in the proposing loop closures. Data are from a single run with a short overlapping segment at the end, with only 17 possible loop closures, which occur at the end of the run as the robot reenters the tunnel. In this run, the <i>Spatial</i> model is clearly primarily responsible for high recall rates, but the addition of the <i>Appearance</i> model does improve recall while also reducing the number places for which the <i>Spatial</i> model must be evaluated. a) Precision/Recall graph: Loop closure is very sensitive to false positives and must be run at 100% precision. b) ROC graph relating false and true positive rates.	12

Fig. 10 Underground tunnel run with loop closure detected by *Appearance* and *Spatial* models. a) and b) Loop closures detected by *Appearance* and *Spatial* model, where the left image corresponds to immediately before loop closure is inserted and the right image corresponds to immediately after. c) Full map including approach trajectory.13

INTENTIONALLY LEFT BLANK.

1. Introduction

Techniques for using mobile robots to generate detailed maps of different environments, referred to as simultaneous localization and mapping (SLAM), have proven to possess tremendous potential as a result of their demonstrated performance. By building an accurate map of the environment, autonomous behaviors can be implemented to perform different tasks depending on a given situation. There are two key factors that directly affect the performance of a SLAM technique: 1) the sensor(s) used and 2) the environment in which the robot is operating. Sensors inherently have a non-negligible error that accumulates over time and adversely affects missions spanning long distances and durations. This is especially true for applications involving small mobile robots where sensor drift and inaccuracies can cause significant mistakes in the generated maps.

To address the issue of map quality degradation as a result of aggregated sensor error, there has been a great deal of effort to solve the loop closure problem, that is, identify when a robot has returned to a previously visited location and then use this information to remove error from the generated map. One such place recognition solution is the fast appearance-based mapping (FAB-Map) method.¹ This algorithm uses a bag-of-words representation to determine the probability that the robot has visited the current place earlier on its trajectory. This bag-of-words model is paired with a Chow Liu tree to create a probabilistic framework that addresses perceptual aliasing, i.e., situations that appear very similar to the available sensors.² Another loop closure solution, the Joint Compatibility Branch and Bound (JCBB) method, uses spatial information rather than appearance data.³ In this approach, the algorithm traverses an interpretation tree in search for the loop closure hypothesis associated with the largest number of non-null, jointly compatible pairings. The traversal is executed by applying the Mahalanobis distance to the nearest neighbor rule so as to achieve a heuristic for branching that explores hypotheses with higher degrees of joint compatibility first.

While all of these loop closure solutions have successfully addressed the problem of recovering from sensor error in real time and one has demonstrated loop closure in an environment spanning an entire city, there are situations in which the environment is too complex to use any one of the aforementioned approaches. In this work, we seek to develop a method capable of solving the loop closure problem during long-duration missions in near-featureless environments, namely, in a system of underground tunnels (Fig. 1), in addition to general urban or natural settings. This specific environment is particularly difficult because nearly all locations visually appear identical and there are next to no distinguishable features at any given time. As a result, we have developed and tested a unified representation for recognizing global loop closure, which incorporates *appearance*-based techniques as well as *spatial*-based techniques using a laser scanner on a mobile robot system.

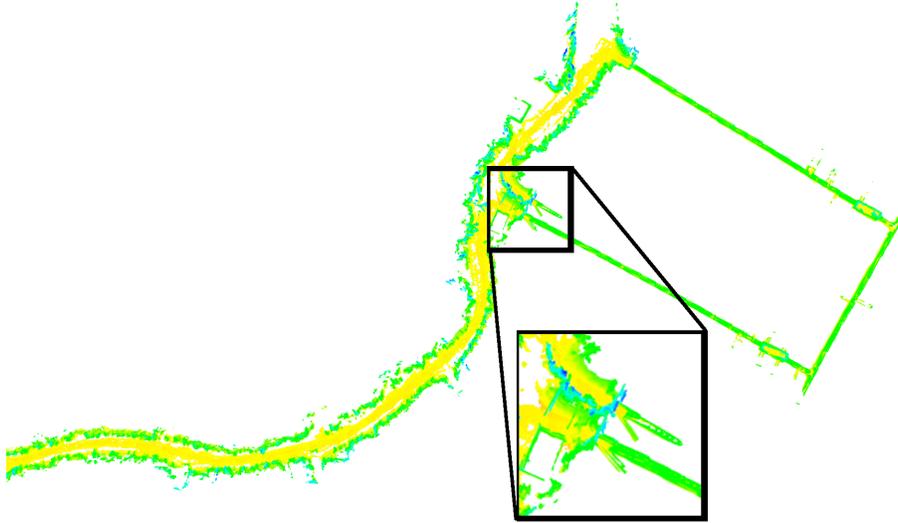


Fig. 1 The map generated from a run where a robot drives approximately 1 km along a road (on the left), then enters an underground facility (thin tunnels on the right), reemerges from the exit at the top, and reenters the first entrance near the middle. The robot has accumulated sufficient pose error during this 3-km trip that loop closure will be needed to correct it. The error can be seen in the zoomed in segment in the lower right.

We first present the mapping system, models, and training process of our approach in Section 2. Next, we describe our experimental design for evaluating our loop closure method in Section 3. The results of these experiments are detailed in Section 4, followed by our corresponding conclusions and future work in Section 5.

2. Approach

2.1 Mapping System

The mapping system used in this work is based on the *OmniMapper* library.^{4,5} This system is a front-end for the GTSAM nonlinear optimization engine,⁶ which provides measurements between places along a trajectory that is optimized by the GTSAM backend. There are two sources of measurements used to build maps using LiDAR: adjacent pose measurements and loop closure measurements. Both of these types of measurements are automatically determined via the generalized iterative closest point (GICP)⁷ implementation provided by the point cloud library (PCL).⁸

2.2 Appearance Model

The first component in the loop closure representation is the *Appearance* model. This model directly compares two places with the Open FAB-Map⁹ library included with OpenCV.¹⁰

Open FAB-Map is an open-source implementation of the fast appearance-based mapping technique.¹ This technique for global loop closure detection is popular, because it has a linear time complexity and is accurate; it achieves nearly 40% recall at 100% precision, which is more than sufficient for mapping. This is achieved through the use of visual features, together with a first-order estimate of the joint probability of observing combinations of visual features.

We use fast point feature histogram (FPFH) descriptors¹¹ computed at intrinsic shape signature keypoints¹² from three-dimensional (3-D) point clouds instead of the visual features that are typically used in FAB-Map. A codebook of representative FPFHs together with a Chow-Liu tree² is trained offline; this procedure is described in Section 3. A place descriptor vector is generated by vector-quantizing FPFH descriptors computed at keypoints found in a given place. Each entry in the place descriptor vector indicates the presence of that codeword in this place. This place descriptor vector is then compared to the ones already seen, as well as a *null place model* representing the average place. The Open FAB-Map library returns the likelihood of a loop closure to each previous place and the *null place*. If the largest likelihood of a previous place exceeds the likelihood of the *null loop closure hypothesis*, then this previous place is a putative loop closure hypothesis to the current place.

Cummins and Newman¹ were able to find a considerable number of loop closures without making any mistakes on very large data sets spanning an entire city using FAB-Map. This technique works very well on visual data; however, we were unable to get this level of performance on LiDAR data. This is likely due to the lower descriptiveness of purely geometric FPFH features over visual features, which benefit from image intensity variation due to texture and geometry. In the next section, we introduce an additional spatial technique to further refine and improve the putative loop closures to increase accuracy.

2.3 Spatial Model

The software presented in this report builds a metric map as loop closures are evaluated. This map has at its backbone a pose graph of values, which estimate the robot’s position at each place. Through the use of the GTSAM SLAM backend, we compute marginal distributions over each pose in the pose graph. These marginal distributions express the location uncertainty along the robot’s trajectory. We can also compute the joint marginal distribution over pairs of poses via the GTSAM backend; however, this is an expensive operation that requires marginalization over all other variables and must be performed judiciously.

For a candidate loop closure selected by the *Appearance* module between pose X_i and X_j , we can compute via GTSAM the following:

1. μ_i and μ_j , the mean pose estimates
2. Σ_{ii} and Σ_{jj} , the unit marginal covariances
3. $\Sigma_{ij} = \Sigma_{ji}^T$, the joint covariance between X_i and X_j

From these values, we can express the conditional distribution over X_j with respect to X_i as

$$p(X_j | X_i) = N(\mu_{j|i}, \Sigma_{j|i}) \quad (1)$$

$$\mu_{j|i} = \mu_j - \mu_i \quad (2)$$

$$\Sigma_{j|i} = \Sigma_{jj} - \Sigma_{ij} \Sigma_{ii}^{-1} \Sigma_{ji}. \quad (3)$$

A weighting factor S is then computed based on the error function:

$$S = \exp(-\mu_{j|i}^T \Sigma_{j|i}^{-1} \mu_{j|i}). \quad (4)$$

S will have its largest value of 1 when $\mu_{j|i} = 0$, corresponding to the condition when X_i and X_j are precisely equal. S is dependent upon the shape of $\Sigma_{j|i}$; it will be large when the Mahalanobis distance between X_i and X_j is small.

2.4 Unified Model

Using either that *Appearance* or *Spatial* component by itself suffers from a number of potential problems, which limit loop closure detection reliability. The *Appearance*-based loop closure prediction technique described in Section 2.2 makes more mistakes than visual FAB-Map due to lower laser feature descriptiveness, and therefore, can only be used with very high thresholds, which limit the number of loop closures accepted to a small number. The *Spatial*-based loop closure prediction technique described in Section 2.3 is too expensive to compute at many locations due to the heavy cost of repeated marginalization of the pose graph for each candidate. Additionally, the *Spatial*-based technique would have very similar values for places adjacent to long-distance loop closures, requiring the use of iterative closest point (ICP)-based validation at many loop closure hypotheses.

Our proposed approach leverages the strengths of both techniques to mitigate the shortcomings presented above. For a given place corresponding to robot pose X_i , usually the robot's current pose as it is proceeding through the environment or operating on a log file, the place descriptor vector is computed and compared to all previously mapped places in addition to the *null place model* via Open FAB-Map. For every other place $\{X_j\}$ for which the likelihood of a loop closure exceeds that of the *null hypothesis*, typically no more than a few places, the *Spatial* model is evaluated. The *Appearance* model's likelihood is scaled by the *Spatial* model and compared to a threshold. If the scaled likelihood exceeds this threshold, then this loop closure candidate is accepted.

Determining the precise relative pose at a loop closure requires the use of a final estimation step. ICP-based techniques typically perform well when initialized close to the correct relative pose. Since a good initialization point is not available for a loop closure due to uncertainty, we have adopted a sampling-based strategy to test many initialization points and accept the ICP result that has the lowest residual error. The conditional distribution computed in the *Spatial* component is

used to generate samples for initialization conditions; the sample with the best residual error is accepted and the resulting relative pose is added as a constraint and solved by the GTSAM backend.

2.5 Training

Global loop closure correction involves the analysis of large data sets to find a compact set of salient and representative features that describe locations a robot will encounter and need to recognize. Selecting the right feature vocabulary is critical for systems aiming to consistently build accurate maps of an environment, because vocabularies compactly represent a place, which enables a robot to recognize the place when it is visited again and then execute the loop closure procedure. The optimization of this feature vocabulary requires the selection of salient features over a set of parameters that must be evaluated on a training set in a relevant environment. Ideally, this type of analysis would be conducted on several thousand individual test runs, and a separate trial would be used to evaluate each of the vocabularies to find the best parameter values for a given environment. This vocabulary could then be used for future mapping missions in comparable environments. In this work, we generated a feature vocabulary using the largest data set in each of the two testing environments. We then assessed the remaining trials in the specific environment using the respective vocabulary.

3. Experimental Design

Our novel global loop closure capability was evaluated using a customized iRobot PackBot, seen in Fig. 2. This man-portable robot was outfitted with a Velodyne HDL-32E LiDAR to capture 3-D point clouds at a rate of one per second. A MicroStrain 3DM-GX2 inertial measurement unit (IMU) captured odometry data to provide initial estimates of ego-motion. The robot also made use of a processing payload that consisted of an Intel Quad-Core i7 ICOM express board and an 802.11 wireless radio. A solid-state drive (SSD) was used to run Ubuntu 12.04, the open-source Robotics Operating System (ROS) and our experimental software, while a second SSD was used to record data. The software used in this report is a mapping system, described in Section 2.1 which uses a *global loop closure* detection method described in Sections 2.2 and 2.3.

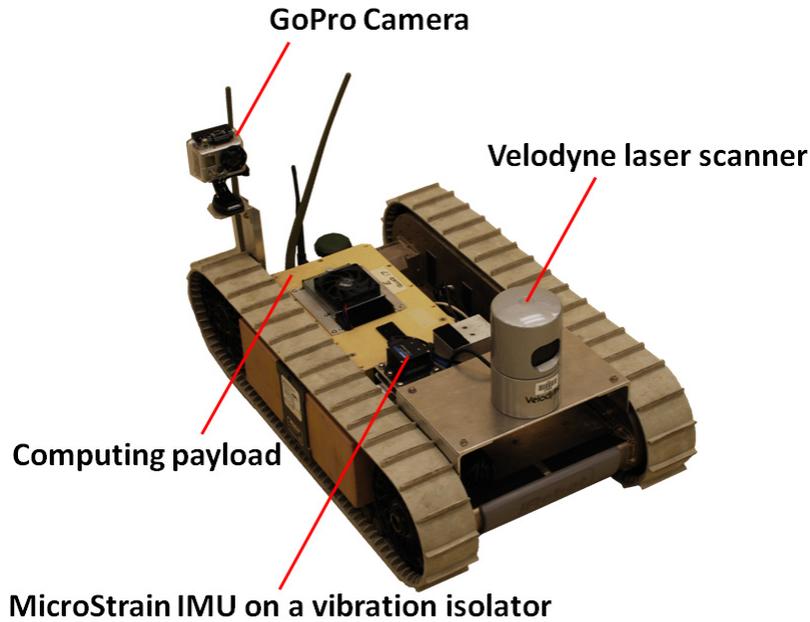


Fig. 2 The iRobot PackBot and Velodyne HDL-32E LiDAR used in this work

The robot was remote-controlled along carefully constructed routes through two complex environments, each of which present unique challenges for long-duration mapping. The routes within these environments start from the same location, traverse a complete loop, and then retrace virtually the same route to end in the same starting location. By collecting data in this fashion, the starting/ending location provided a concrete point of reference and navigating the same route twice ensured a large number of possible locations for executing loop closure. The data collected by the robot were post-processed and used for assessing the performance of the loop closure approach.

The first operational environment that this approach was employed in was an outdoor, urban setting. Designed to simulate a small city, this training facility provided buildings, vegetation, and realistic props that would be found in a town. The robot was driven along five routes, as shown in Fig. 3, ranging from 750 m to 2.2 km total distance traveled. Even though these test runs were completely outdoors, global positioning system (GPS) data were not used in these experiments.

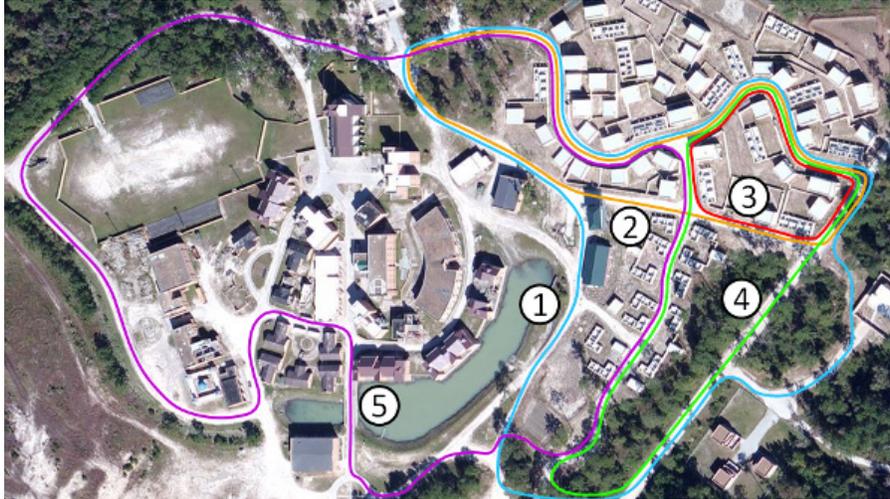


Fig. 3 A satellite photograph of the urban testing facility with each of the five routes labeled. Loop 1 is highlighted in cyan, loop 2 in orange, loop 3 in red, loop 4 in light green, and loop 5 in purple. Each loop was driven twice so that the loop closure method could be evaluated for the respective run.

The second environment used for testing was an underground tunnel complex. For these tests, the robot started outdoors near dense vegetation, maneuvered approximately 1 km, and then entered a long, straight tunnel. After navigating approximately 1 km through several tunnels in the facility, the robot exited the complex at a different location than it entered, drove outdoors to the starting location, and then repeated the route for a total of nearly 3 km of distance traveled. An overhead map of this environment can be seen in Fig. 1. These tunnels present a particularly challenging operating environment because of the lack of features throughout the route. In general, the tunnels were indistinguishable and only occasionally had salient features that would be useful for location recognition. An example of this austere environment can be seen in Fig. 4.



Fig. 4 A photograph of the near-featureless underground tunnel environment

In each environment, the data were collected using the Velodyne HDL-32E LiDAR shown in Fig. 2. This sensor provides point cloud data with a dense horizontal resolution; however, since it only has 32 lasers, the vertical resolution is too coarse for 3-D interest point detection and descriptor extraction. To overcome this limitation, we accumulate 10 s' worth of point cloud data into grouped point clouds while the robot is moving. These data are aligned via GICP to produce grouped places along the robot's trajectory that have sufficient resolution. This procedure can be seen in Fig. 5.

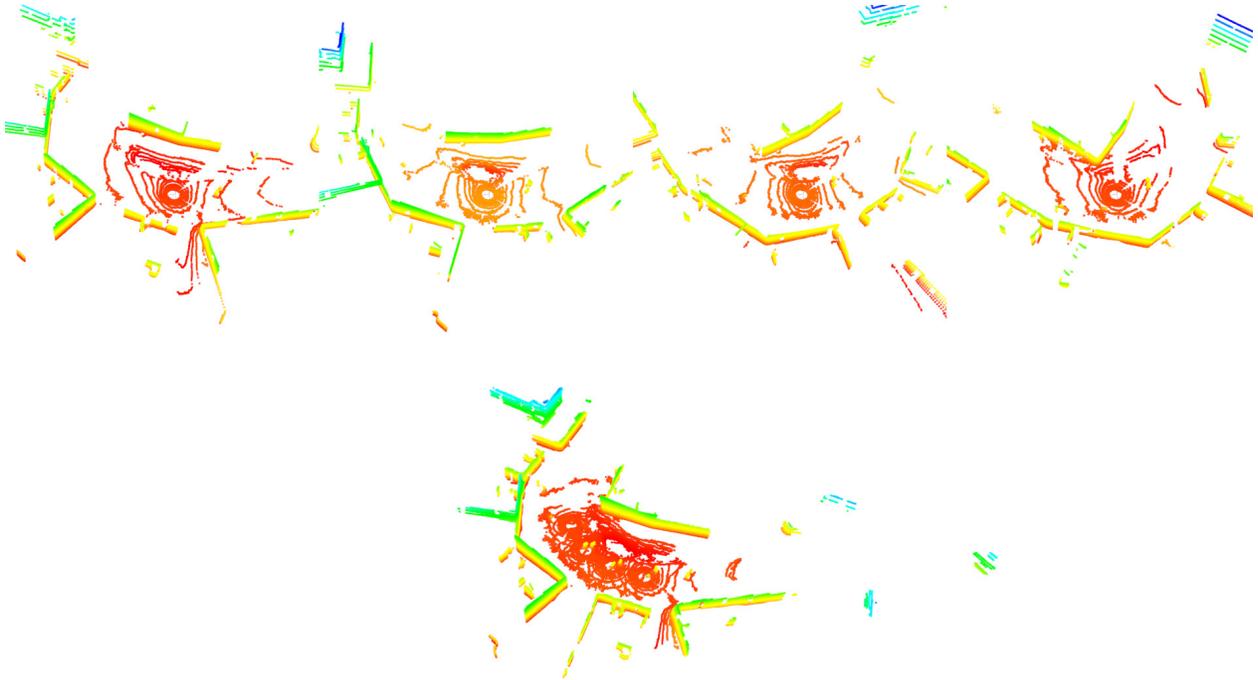


Fig. 5 The point cloud merge utility puts together adjacent point clouds into higher resolution places suitable for feature extraction. Four point clouds (top row) are combined into one place (bottom row). Four point clouds are shown for clarity; 10 are used in practice.

After the point clouds have been grouped into places, we then used one of the data sets for a given environment to develop a vocabulary of descriptors, as described in Section 2.5. These descriptors represent the occurrence of some feature in the scene observed by the LiDAR. The remaining data sets from the respective environment were then tested using the generated vocabulary. In order to evaluate whether the system determines the correct location to perform a loop closure, we generated ground-truth data by manually selecting the optimal pairs of poses in each data set. We developed a ground-truthing utility that allows us to step through each data set and choose the best previous pose that matches the location of the current pose. An example of this utility is depicted in Fig 6.

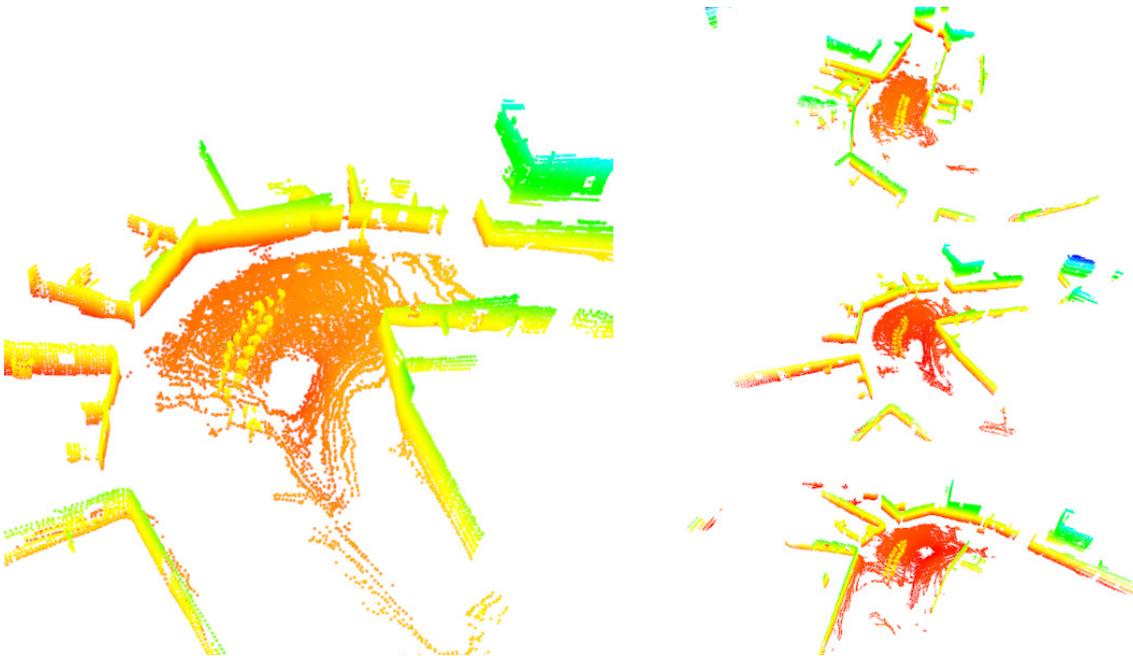


Fig. 6 A visualization of the ground-truthing utility. The leftmost point cloud is taken from the current pose of the robot. The point clouds on the right are three sequential point clouds that can be chosen if any match the location of the point cloud on the left. The user can iterate through all of the previous point clouds and choose the best match.

4. Results

Each environment was run through the three options for the loop closure detection setup: *Appearance*, *Spatial*, and *Appearance+Spatial*. The loop closure detection under consideration was used to choose the best previously visited place together with a confidence level, as log data were processed. These predicted loop closures were compared against ground truth to determine precision and recall values at various confidence levels. Precision is a measure of how accurate the predictions are; it indicates the rate of true positive results divided by total positive results. Recall is a measure of what proportion of positive results are identified. A conservative confidence threshold will maintain high precision and may have lower recall to avoid making any mistakes. Loop closure is particularly sensitive to false positives, so thresholds are purposefully selected to maintain 100% precision despite lower recall rates.

The first test environment consists of the five loop routes in the urban test facility shown in Fig. 3. All loop closure predictions with confidence were ground truthed and used to generate the precision/recall and receiver operating characteristic (ROC) curves shown in Fig. 7. Cummins and Newman¹ described recall rates around 30% to 40% with visual features at full precision. It can be seen that our recall rate is quite a bit lower at 12% with both components of the model and significantly lower at 2% with either component alone. The *Spatial* model achieves similar

precision to the combined model at various recall levels; however, it makes critical high-confidence mistakes after only a few loop closures are detected. These are suppressed until a higher recall level by using the combined model. It should also be noted that the mechanism for computing the *Spatial* model would be computationally prohibitive for real-time operation on large-scale mapping runs. In the combined model, the *Spatial* model is only computed for places where the *Appearance* model indicates that loop closure is possible with a higher likelihood than the null model.

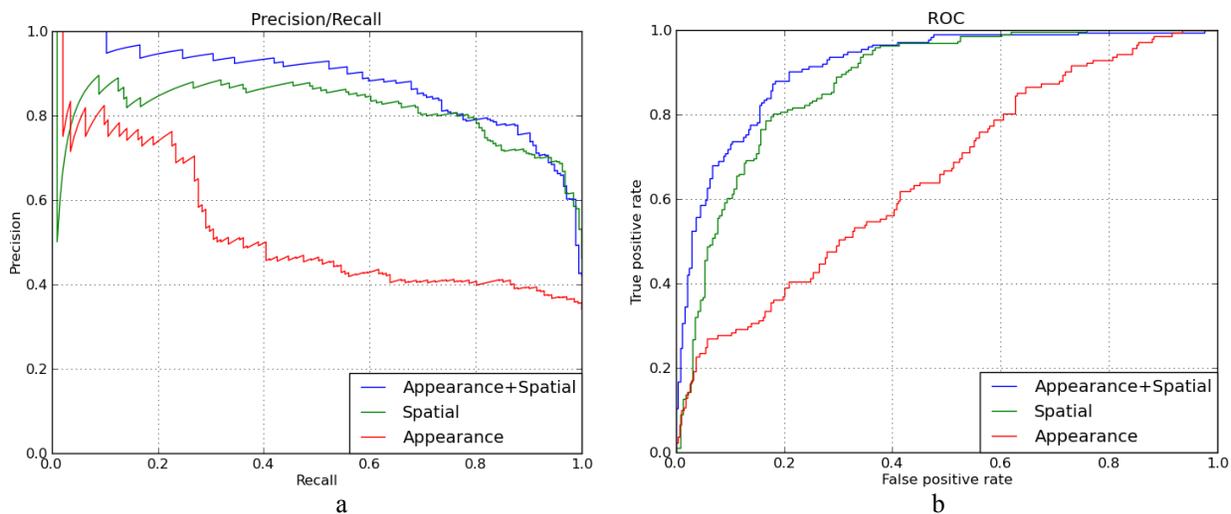


Fig. 7 Loop closure performance on all runs in the urban test complex, with three options for which components are used in proposing loop closures. a) Precision/recall graph: Loop closure is very sensitive to false positives and must be run at 100% precision. *Appearance+Spatial* can operate together at around 10% recall without sacrificing precision. Either technique alone has a significantly lower recall. b) ROC graph relating false and true positive rates.

An example loop closure generated by the full system can be seen in Fig. 8. A threshold was selected from the precision/recall curve corresponding to the lowest confidence loop closure that was still correct across all urban loop data sets. The map shown in Fig. 8 corresponds to run 5 from Fig. 3. The loop closure shown in Fig. 8 is found at a very distinctive place. The loop closure is proposed by the combined *Appearance+Spatial* model, and the relative pose is solved through GICP with sampled initial conditions from the joint conditional pose distribution.

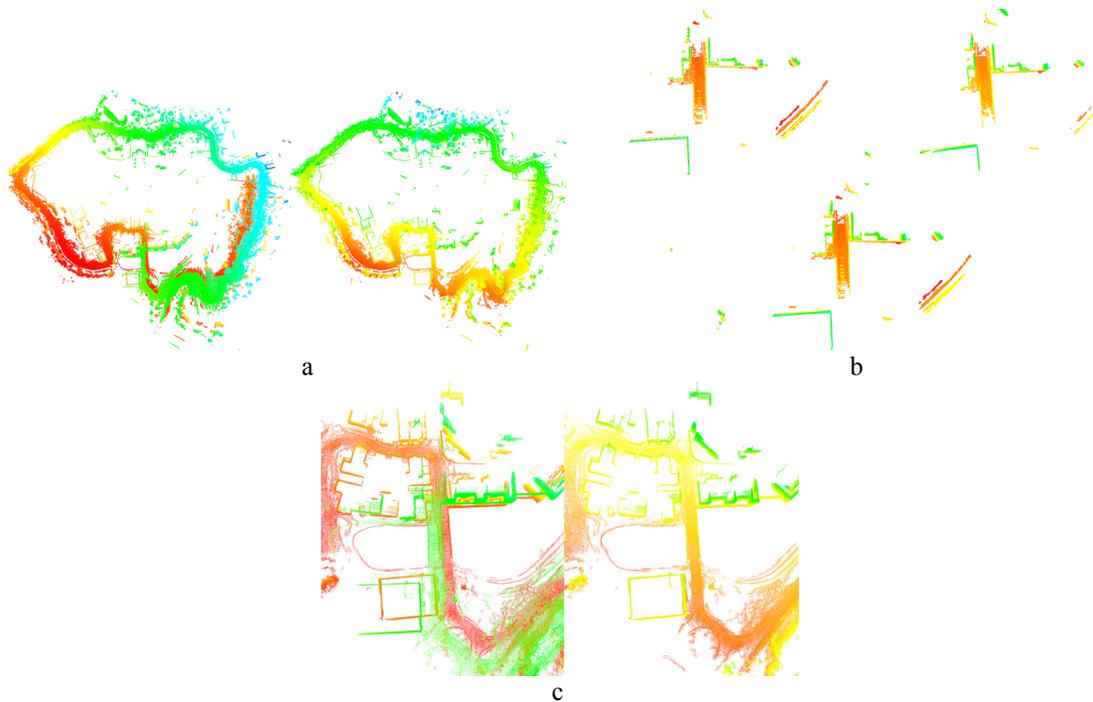


Fig. 8 Loop closure found in run 5 from Fig. 3 using the *Appearance* and *Spatial* models. a) Loop closure detected by *Appearance* and *Spatial* model: the left image corresponds to immediately before loop closure is inserted and the right image corresponds to after. b) Two candidate places shown in upper half of image have their relative poses computed through sampling initial conditions and running GICP; the resulting fused point cloud is shown in lower half of image. c) Detail view of loop closure.

The second type of environment tested was an underground tunnel complex. Precision/recall and ROC curves can be seen in Fig. 9. This result comes from a single data set with a short overlapping segment at the end of the run. In this run, the *Appearance* model is not nearly as useful as in the urban complex due to the fact that the portion of the trajectory that overlaps is almost entirely contained within the tunnel where there are no distinctive features. When it is used in the combined *Appearance*+*Spatial* model, however, it is able to improve recall over the *Spatial* model by itself. In this case, the *Spatial* model happens to have very high recall rates; this is due to the good performance of the mapping system in estimating the robot's trajectory in the absence of loop closure. If the mapping system had been a little further off, then the *Spatial* model would have had lower performance. This would have also affected the combined model's recall, but to a lesser extent since the *Appearance* model would have been unaffected.

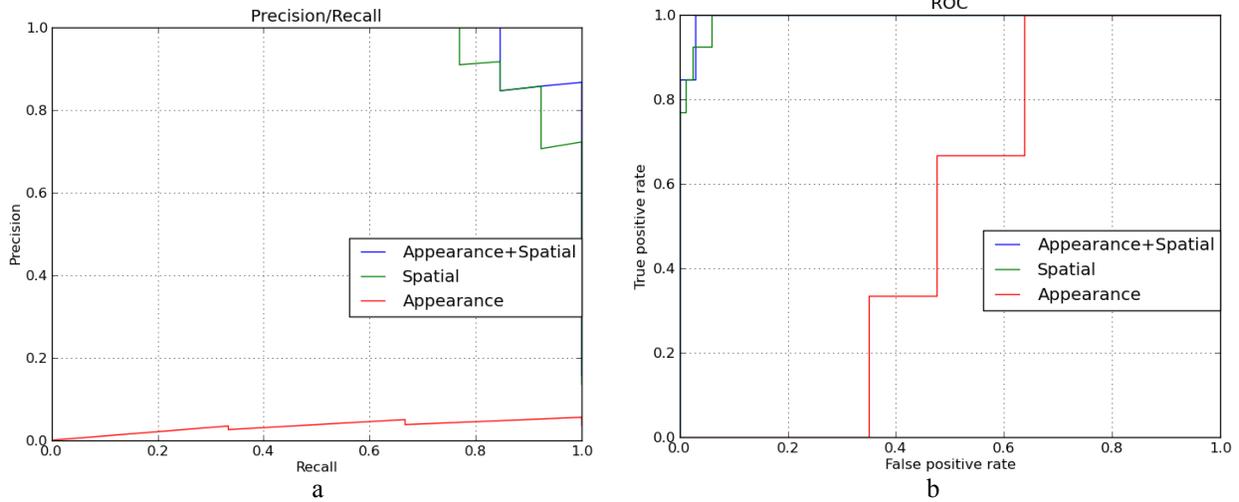


Fig. 9 Loop closure performance on a run in the underground tunnel complex, with three options used in the proposing loop closures. Data are from a single run with a short overlapping segment at the end, with only 17 possible loop closures, which occur at the end of the run as the robot reenters the tunnel. In this run, the *Spatial* model is clearly primarily responsible for high recall rates, but the addition of the *Appearance* model does improve recall while also reducing the number places for which the *Spatial* model must be evaluated. a) Precision/Recall graph: Loop closure is very sensitive to false positives and must be run at 100% precision. b) ROC graph relating false and true positive rates.

An example loop closure generated by the full system in the underground tunnel complex can be seen in Fig10. In this run, the robot starts 1 km away from the tunnel entrance, proceeds to enter the right tunnel, and exits the left tunnel before reentering the right tunnel. At this point, a loop closure is detected and solved by the ICP system, producing a corrected map.

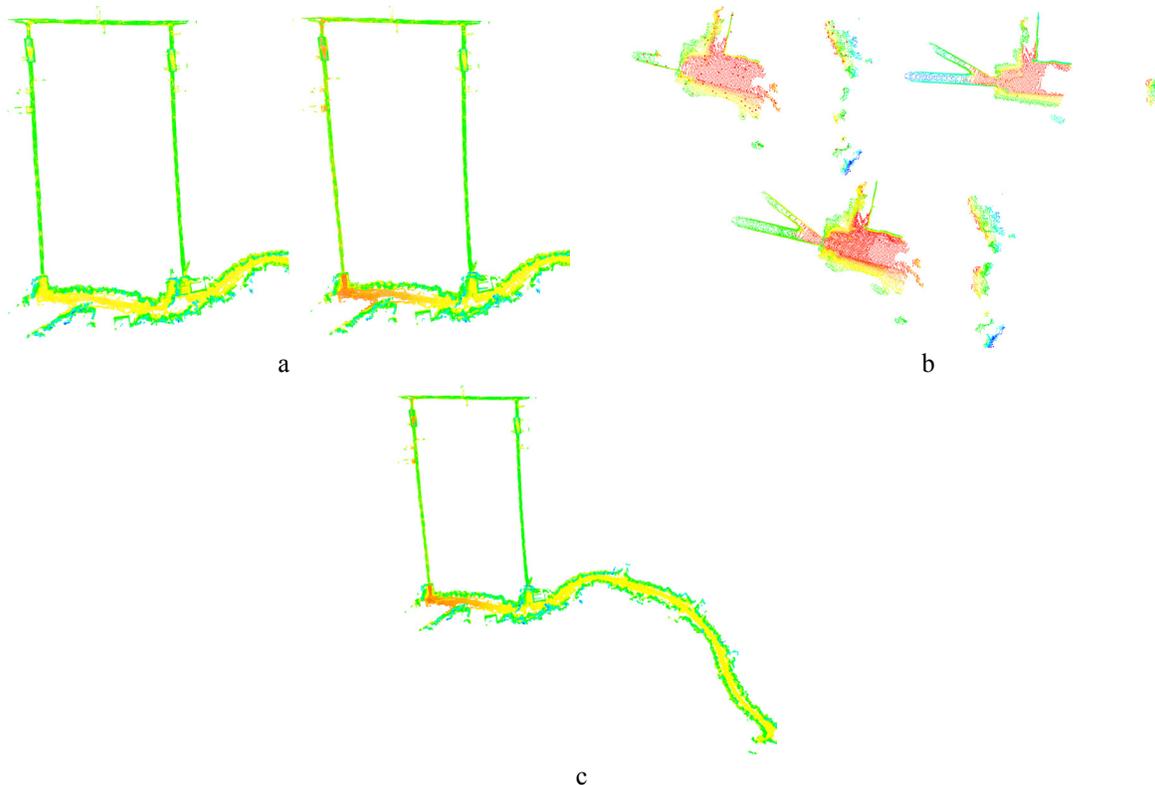


Fig. 10 Underground tunnel run with loop closure detected by *Appearance* and *Spatial* models. a) and b) Loop closures detected by *Appearance* and *Spatial* model, where the left image corresponds to immediately before loop closure is inserted and the right image corresponds to immediately after. c) Full map including approach trajectory.

5. Conclusion and Future Work

The motivation for incorporating appearance-based loop closure detection techniques into our mapping system was to find loop closures in difficult environments such as austere underground tunnels. We have applied FAB-Map to 3-D LiDAR data; however, by itself, the loop closure recall was much lower than the results shown for visual data in the literature. We believe that this is due to lower distinctiveness in 3-D LiDAR features as compared with visual features due to the absence of texture. We added a *Spatial* model to validate the loop closures coming from the *Appearance* model. This resulted in sufficient recall levels to perform loop closures, generating coherent maps from two diverse types of environments, austere underground tunnels and a simulated urban training facility.

We plan to add a *Locality* model to our loop closure method, which would share information about neighboring places, boosting marginal loop closures if they are locally consistent with their neighbors. The *Locality* model would represent the belief that if pose x_i has a loop closure to pose x_j , then the poses near x_i should tend to be good loop closures for places near x_j . This is

especially true for the type of environments we evaluated in this report, where long corridors admit only two approach directions that will be adjacent to in-between poses. In cases where this is not necessarily true, such as large open rooms or outdoors, where we might approach a loop closure at $x_i \rightarrow x_j$ from a direction that is not represented in the neighborhood of the trajectory chain leading through x_j , there would be a spontaneous loop closure model, which allows this type of loop closure to still be detected.

Our technique extracts FPFH features from 3-D LiDAR point clouds to generate place descriptor vectors. This choice of feature descriptor is preliminary and could benefit from further evaluation of other alternatives.

As mentioned in Section 2.5, the generation of a representative feature vocabulary is paramount to the performance of a global loop closure capability. The quality of the vocabulary, i.e., how accurately the vocabulary captures the features of an environment, can be drastically improved using high-performance computing. In future work, we plan to train thousands of vocabularies for a specific environment on a high-performance computer so that we can empirically determine the optimal parameter values for a robust vocabulary. Using this vocabulary, we will evaluate the performance of our global loop closure technique in a similar environment.

6. References

1. Cummins M, Newman P. FAB-MAP: probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*. 2008;27(6):647–665.
2. Chow C, Liu C. Approximating discrete probability distributions with dependence trees,” *Information Theory*. *IEEE Transactions on*. 1968;14(3):462–467.
3. Neira J, Tardós J. Data association in stochastic mapping using the joint compatibility test. in *Robotics and Automation (ICRA), 2001 IEEE International Conference on*. 2001;17(6):IEEE, 890–897.
4. Rogers, John G. III. Life-long mapping of objects and places in domestic environments. Ph.D. dissertation, Georgia Institute of Technology, 2013. [Online]. Available: <http://hdl.handle.net/1853/47736>
5. Trevor AJB, Rogers JG III, Christensen HI. Planar surface SLAM with 3D and 2D sensors. in *IEEE International Conference on Robotics and Automation (ICRA), 2012*.
6. Dellaert F. GTSAM download. [Online]. Available: <https://borg.cc.gatech.edu/download>
7. Segal A, Haehnel D, Thrun S. Generalized-ICP. in *Robotics: Science and Systems*. 2009;2:4.
8. Rusu RB, Cousins S. 3D is here: point cloud library (PCL). in *IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 2011*.
9. Glover A, Maddern W, Warren M, Reid S, Milford M, Wyeth G. OpenFABMAP: An open source toolbox for appearance-based loop closure detection. in *Robotics and Automation (ICRA). 2012 IEEE International Conference on*. IEEE. 2012;4730–4735.
10. Bradski G, Kaehler, A. *Learning OpenCV: computer vision with the OpenCV library*. O’Reilly, 2008.
11. Rusu RB, Blodow N, Beetz M. Fast point feature histograms FPFH for 3D registration. in *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE, 2009, 3212–3217.
12. Zhong Y. Intrinsic shape signatures: A shape descriptor for 3D object recognition. in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, 689–696.

List of Symbols, Abbreviations, and Acronyms

3-D	three-dimensional
FAB-Map	fast appearance-based mapping
FPFH	fast point feature histogram
GICP	generalized iterative closest point
GPS	global positioning system
JCBB	Joint Compatibility Branch and Bound
PCL	point cloud library
ROC	receiver operating characteristic
ROS	Robotics Operating System
SLAM	simultaneous localization and mapping
SSD	solid-state drive

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIRECTOR
(PDF) US ARMY RESEARCH LAB
RDRL CIO LL
IMAL HRA

1 GOVT PRNTG OFC
(PDF) A MALHOTRA

4 DIRECTOR
(PDF) US ARMY RESEARCH LAB
3 RDRL CII
(HC) BARBARA BROOME (1 HC, 1 PDF)
RDRL CII A
JOHN G ROGERS III (1 HC, 1 PDF)
JASON M GREGORY (1 HC, 1 PDF)
STUART H YOUNG (1 PDF)

INTENTIONALLY LEFT BLANK.