

Anytime Online Novelty Detection for Vehicle Safeguarding

Boris Sofman

J. Andrew Bagnell

Anthony Stentz

CMU-RI-TR-09-17

April 2009

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

© Carnegie Mellon University

Abstract

Novelty detection is often treated as a one-class classification problem: how to segment a data set of examples from everything else that would be considered novel or abnormal. Almost all existing novelty detection techniques, however, suffer from diminished performance when the number of less relevant, redundant or noisy features increases, as often the case with high-dimensional feature spaces. Additionally, many of these algorithms are not suited for online use, a trait that is highly desirable for many robotic applications. We present a novelty detection algorithm that is able to address this sensitivity to high feature dimensionality by utilizing prior class information within the training set. Additionally, our anytime algorithm is well suited for online use when a constantly adjusting environmental model is beneficial. We apply this algorithm to online detection of novel perception system input on an outdoor mobile robot and argue how such abilities could be key in increasing the real-world applications and impact of mobile robotics¹.

¹Most figures in this paper are best viewed in color.

Contents

1	Introduction	1
2	Novelty Detection	3
3	Approach	4
3.1	Formalization	4
3.2	Improved Dimensionality Reduction	5
3.3	Framing as Instance of NORMA	7
3.4	Query Optimization	7
4	Application to Mobile Robotics	9
5	Experimental Results	11
6	Conclusion	16

1 Introduction

Many autonomous unmanned ground vehicles (UGVs) have advanced to a level where they are competent and reliable a high percentage of the time in many environments [1, 2, 3]. Most of these systems, however, are heavily engineered for the domains they are intended to operate in. Any deviation from these domains often results in sub-optimal performance or even complete failure. Given the cost of such systems and the importance of safety and reliability in many of the tasks that they are intended for, even a relatively rare rate of failure is unacceptable. In many domains that are prime candidates for mobile robotic applications such as space exploration, transportation, military reconnaissance, and agricultural tasks, the risk of catastrophic failure, however small, is a primary reason why autonomous systems are still under-utilized despite already demonstrating impressive abilities.

One approach to addressing this limitation is for a UGV to be able to identify situations that it is likely untrained to handle *before* it experiences a major failure. This problem therefore becomes one of novelty detection: how a robot can identify when perception system inputs differ from prior inputs seen during training or previous operation. With this ability, the system can either avoid novel locations to minimize risk or stop and enlist human help via supervisory control or tele-operation (see Figure 1).

Two common limitations of novelty detection systems are particularly relevant to the mobile robotics domain. Autonomous systems often need to learn from their experiences and continually adjust their models of what is normal and what is novel. For example, if human feedback were to confirm that a certain type of environment selected as novel is actually safe to handle with the existing autonomy system or demonstrate to the system the proper way to handle the situation (as in [4]), the model no longer needs to identify such inputs as novel. Most novelty detection approaches, however, build a model of the normal set of examples a priori in batch in order to detect novel examples in the future but are unable to update that model online without retraining.

Furthermore, existing novelty detection techniques see diminished performance when using high-dimensional feature spaces, particularly when some features are less relevant, redundant, or noisy. These qualities are particularly common in features from many UGV perception systems due to the variety of sensors and uncertainty about how these features relate to novelty. For example, the relevance of camera-based features such as color and texture of an area of the environment to novelty (or similarity metrics in general) is difficult to understand as subsets of the features could contain redundant information or be mostly irrelevant. It is therefore important for novelty detection techniques to be resilient to such feature properties.

We present an online approach that addresses these common problems with novelty detection techniques. We approach the problem of novelty detection as one of online density estimation where seen examples generate an influence of familiarity in feature space. When prior class information is available, we show how using Multiple Discriminant Analysis (MDA) for generating a reduced dimensional subspace to operate in rather than other common techniques such as Principal Components Analysis (PCA) can make the novelty detection system more robust to issues associated with high-dimensional feature spaces. In effect, this creates a lower dimensional subspace

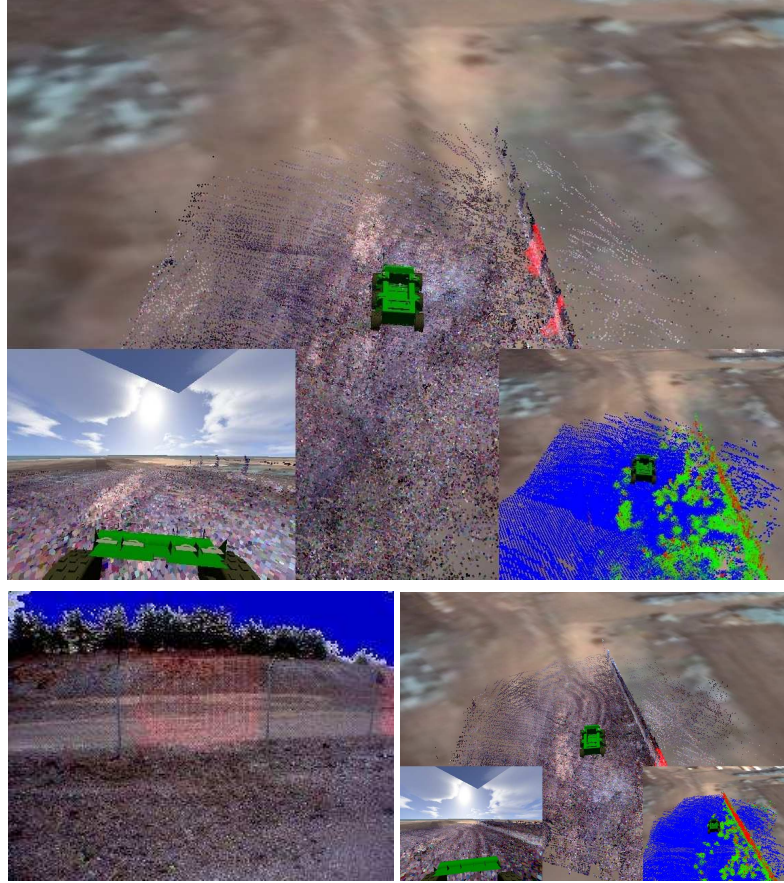


Figure 1: Sample result from online novelty detection algorithm onboard Crusher, a large UGV. Chain-link fence was detected as novel (top and left, novelty shown in red) with respect to the large variety of terrain and vegetation previously encountered. After an initial stretch being identified as novel, subsequent portions of the fence are no longer flagged (right) due to the algorithm’s online training ability. As with all future similar images, insets within the top image show a first-person view (left inset) and the classification of the environment by the perception system into road, vegetation, and solid obstacle in blue, green and red respectively (right inset).

that truly captures *what makes things novel*. Additionally, our algorithm can be framed as a variant of the NORMA algorithm, an online kernelized Support Vector Machine (SVM) optimized through stochastic gradient descent, and therefore shares its favorable qualities [5]. Along with its anytime properties, this allows our algorithm to better deal with the real-time demands of online tasks.

While this work was targeted toward mobile robotics applications, the approaches here are more generally applicable to any domain which can benefit from online novelty

detection.

The next section presents background on novelty detection techniques and some example applications. Section 3 presents our novelty detection algorithm, followed by an explanation of how this technique can be applied to mobile robotics in Section 4, results from field testing on a large UGV in Section 5 and concluding remarks in Section 6.

2 Novelty Detection

Novelty detection techniques (also referred to as anomaly or outlier detection) have been applied to a wide range of domains such as detecting structural faults [6], abnormal jet engine operation [7], computer system intrusion detection [8], and identifying masses in mammograms [9]. In the robotics domain some have incorporated novelty detection systems within inspection robots [10, 11].

Novelty detection is often treated as a one-class classification problem. In training the system sees a variety of “normal” examples (and corresponding features) and later the system tries to identify input that does not fit into the trained model in order to separate novel from non-novel examples. Instances of abnormalities or novel situations are often rare during the training phase so a traditional classifier approach cannot be used to identify novelty in most cases.

Most novelty detection approaches fall into one of several categories. Statistical or density estimation techniques model the “normal” class in order to identify whether a test sample comes from the same distribution or not. Such approaches include Parzen window density estimators, nearest neighbor-based estimators, and Gaussian mixture models [12]. These techniques often use a lower-dimensional representation of the data generated through techniques such as PCA.

Other approaches attempt to distinguish the class of instances in the training set from all other possible instances in the feature space. Schölkopf et al. [13] show how an SVM can be used for specifically this purpose. A hyper-plane is constructed to separate the data points from the origin in feature space by the maximum margin. One application of this technique was document classification [14]. A noticeable drawback of this approach is that it makes an inherent assumption that the origin is a suitable prior for the novel class. This limitation was addressed by [15] by attracting the decision boundary toward the center of the data distribution rather than repelling it from the origin. A similar approach encloses the data in a sphere of minimal radius, using kernel functions to deal with non-spherical distributed data [16]. These techniques all require solutions to linear or quadratic programs with slack variables to handle outliers.

Another class of techniques attempts to detect novelty by compressing the representation of the data and measuring reconstruction error of the input. The key idea here is that instances of the original data distribution are expected to be reconstructed accurately while novel instances are not. A simple threshold can then be used to detect novel examples. The simplest method of this type uses a subset of the eigenvectors generated by PCA to reconstruct the input. An obvious limitation here is that PCA will perform poorly if the data is non-linear. This limitation was addressed by using a kernel PCA based novelty detector [17]. Benefits of more sophisticated auto-encoders,

neural networks that attempt to reconstruct their inputs through narrow hidden layers, have been studied as well [18].

Online novelty detection has received significantly less attention than its offline counterpart. Since it is often important to be able to adjust the model of what is considered novel in real-time, many of the above techniques are not suitable for online use as they require significant batch training prior to operation. While Neto et al. [10] replaced the use of PCA for novelty detection with an implementation of iterative PCA, performance was still largely influenced by the initial data set used for training. Marsland proposed a unique approach that models the phenomenon of habituation where the brain learns to ignore repeated stimuli [11]. This is accomplished through a clustering network called a Grow When Required (GWR) network. This network keeps track of firing patterns of nodes and allows the insertion of new nodes to allow online adaptation.

Markou and Singh have written a pair of extensive survey articles detailing many additional novelty detection applications and techniques [19, 20].

The performance of the above-mentioned novelty detection approaches, however, quickly deteriorates as the number of less relevant or noisy features grows. The disproportionately high variance of many of these features make it difficult for many of these algorithms to capture an adequate model of the training data and their effects quickly begin to dominate more relevant features in making predictions. Our algorithm addresses this crucial limitation in cases where class information is available within the training set while still being suitable for online use.

3 Approach

3.1 Formalization

The goal of novelty detection can be stated as follows: given a training set $D = \{\mathbf{x}\}_{1 \dots N} \in \mathcal{X}$ where $\mathbf{x}_i = \{x_i^1, \dots, x_i^k\}$, learn a function $f : \mathcal{X} \rightarrow \{novel, not-novel\}$. In the online scenario, each time step t provides an example \mathbf{x}_t and a prediction $f_t(\mathbf{x}_t)$ is made.

We perform online novelty detection using the online density estimation technique shown in Algorithm 1. All possible functions f are elements of a *reproducing kernel Hilbert space* \mathcal{H} [21]. All $f \in \mathcal{H}$ are therefore linear combinations of kernel functions:

$$f_t(\mathbf{x}_t) = \sum_{i=1}^{t-1} \alpha_i k(\mathbf{x}_i, \mathbf{x}_t) \quad (1)$$

We make the assumption that proximity in feature space is directly related to similarity. Observed examples deemed as novel are therefore remembered and have an influence of familiarity on future examples through the kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$. A novelty threshold, γ , and a learning rate, η , are initially selected. For each example \mathbf{x}_t , the algorithm accumulates the influence of all previously seen novel examples (line 5). If this sum does not exceed γ then the example is identified as novel and is remembered for future novelty prediction (line 7). Non-novel examples are not stored as they

have minimal impact on future novelty computations (even though a coefficient of 0 is assigned in line 9 for clarity, these examples are not stored). We suggest simply using the Gaussian kernel with an appropriate variance σ^2 :

$$k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}} \quad (2)$$

Algorithm 1 Online novelty detection algorithm

```

1: given: A sequence of features  $S = (\mathbf{x}_i)_{1 \dots T}$ ; a novelty threshold  $\gamma$ ; a learning rate  $\eta$ 
2: outputs: A sequence of hypotheses  $\mathbf{f} = (f_1(\mathbf{x}_1), f_2(\mathbf{x}_2), \dots)$ 
3: initialize:  $t \leftarrow 1$ 
4: loop
5:    $f_t(\mathbf{x}_t) \leftarrow \sum_{i=1}^{t-1} \alpha_i k(\mathbf{x}_i, \mathbf{x}_t)$ 
6:   if  $f_t(\mathbf{x}_t) < \gamma$  then
7:      $\alpha_t \leftarrow \eta$ 
8:   else
9:      $\alpha_t \leftarrow 0$ 
10:  end if
11:   $t \leftarrow t + 1$ 
12: end loop

```

3.2 Improved Dimensionality Reduction

Especially if the number of features is large, it may first be necessary to project the high-dimensional input \mathbf{x}_t into a lower-dimensional subspace more suitable for novelty detection using distance metrics. The most common choice for this among dimensionality reduction (and novelty detection) techniques is PCA. PCA finds a linear transformation that minimizes the reconstruction error in a least-squares sense. If subsets of the features are redundant, noisy or are dominated disproportionately by a subset of the training set, however, applying techniques such as PCA, or any unsupervised dimensionality reduction technique for that matter, may yield disappointing results as precisely the most relevant directions for differentiation may be discarded in order to reduce reconstruction error of a less relevant portion of the feature space.

Rather than optimizing for reconstruction error, *discriminant analysis* seeks transformations that are efficient for discriminating between different classes within the data. Multiple Discriminant Analysis, a generalization of Fischer's linear discriminant for more than two classes, computes the linear transformation that maximizes the separation between the class means while keeping the class distributions themselves compact, making it useful for classification tasks [12].

We argue that when prior class information for the training set is available, using MDA to construct a lower dimensional subspace using labeled classes not only optimizes for known class separability but likely leads to separability between known classes and novel classes. In cases described earlier that result in poor performance

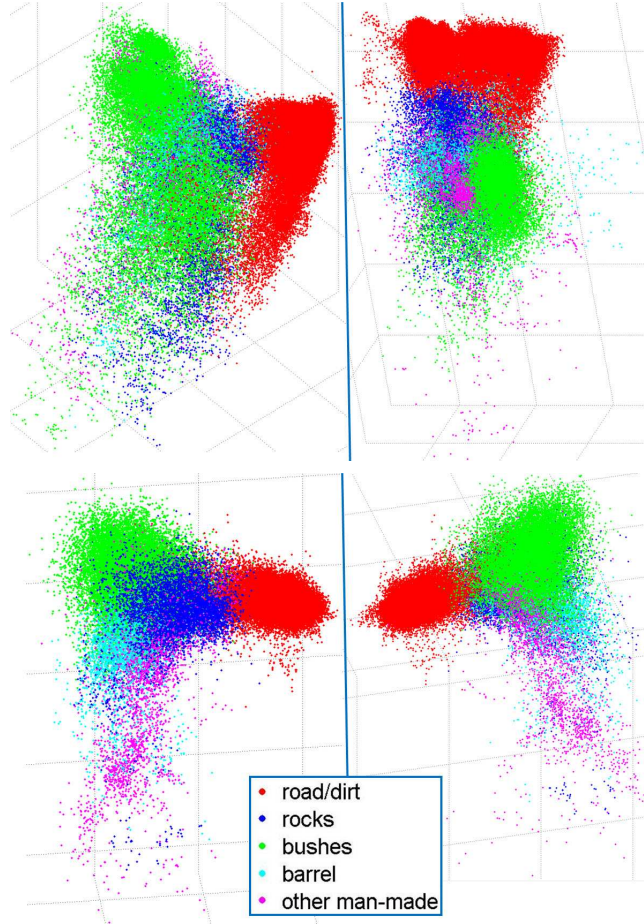


Figure 2: All data points projected onto the subspace defined by the first three basis vectors computed by PCA (top) and LDA (bottom). Only the first four classes were used to construct the subspaces ('other man-made' class was withheld as a test class). The LDA-based projection clearly shows significantly more separation between the new man-made class and the known classes, implying a more suitable subspace for novelty detection.

when using PCA, MDA will largely ignore features that do not aid in class discrimination, instead focusing on the obviously differentiating features. The key observation here is that novelty detection is about encountering new classes, so by using discriminating ability as the metric for constructing a subspace, one can capture the combinations of features that make known classes novel *with respect to each other* and likely generalize to previously unseen environments, in effect capturing *what makes things novel*.

Experimental validation of this theory within the domain of mobile robotics is pre-

sented in Sections 4 and 5.

3.3 Framing as Instance of NORMA

The NORMA algorithm is a stochastic gradient descent algorithm that allows the use of kernel estimators for online learning tasks [5]. As with our algorithm, f is expressed as a linear combination of kernels (1). NORMA uses a piecewise differentiable convex loss function l such that at each step t we add a new kernel centered at \mathbf{x}_t with the coefficient:

$$\alpha_t = -\eta l'(\mathbf{x}_t, y_t, f_t) \quad (3)$$

Our algorithm can easily be framed as an online SVM instance of NORMA using a hinge loss function as follows:

$$y_t = \gamma \quad (4)$$

$$l(\mathbf{x}_t, y_t, f_t) = \max(0, y_t - f_t(\mathbf{x}_t)) \quad (5)$$

Taking the derivative of (5), we get:

$$l'(\mathbf{x}_t, y_t, f_t) = \begin{cases} -1 & \text{if } f_t(\mathbf{x}_t) < \gamma \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

As before, the gradient of our loss is non-zero only when the accumulated contributions from stored examples are less than the novelty threshold γ , signifying that the example is novel. From (3) and (6) we then get:

$$\alpha_t = \begin{cases} \eta & \text{if } f_t(\mathbf{x}_t) < \gamma \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

This is equivalent to the update steps in lines 7 and 9 of Algorithm 1, showing that our algorithm can be framed as a specific instance of the NORMA algorithm.

NORMA produces a variety of useful bounds on the expected cumulative loss [5]. For novelty detection this directly relates to the number of examples that are expected to be flagged as novel. This means we are competitive with respect to the best $f \in \mathcal{H}$ in terms of representing our sample distribution with the fewest number of examples. This is to our advantage both from a computational perspective, since memory and prediction costs scale with the number of remembered examples, as well as performance since we want to minimize false positives that may be costly to handle.

3.4 Query Optimization

Without further measures, the potential number of basis functions stored by Algorithm 1 could grow without bound. NORMA deals with this issue by decaying all coefficients α_i and dropping terms when their coefficients fall below some threshold. This is unsuitable for our application since we do not want to repeatedly flag similar examples

Algorithm 2 Online novelty detection algorithm with query optimization

```
1: given: A sequence of features  $S = (\mathbf{x}_i)_{1..T}$ ; a novelty threshold  $\gamma$ ; a learning rate  $\eta$ ; a maximum example storage capacity  $N$ 
2: outputs: A sequence of hypotheses  $\mathbf{f} = (f_1(\mathbf{x}_1), f_2(\mathbf{x}_2), \dots)$ 
3: initialize:  $t \leftarrow 1$ ;  $n \leftarrow 0$ 
4: loop
5:    $i \leftarrow 1$ 
6:    $f_t(\mathbf{x}_t) \leftarrow 0$ 
7:   while  $f_t(\mathbf{x}_t) < \gamma$  and  $i \leq n$  do
8:      $f_t(\mathbf{x}_t) \leftarrow f_t(\mathbf{x}_t) + \alpha_i k(\mathbf{x}_i, \mathbf{x}_t)$ 
9:      $i \leftarrow i + 1$ 
10:  end while
11:  if  $f_t(\mathbf{x}_t) < \gamma$  then
12:     $\alpha_{n+1} \leftarrow \eta$ 
13:     $\mathbf{x}_{n+1} \leftarrow \mathbf{x}_t$ 
14:     $n \leftarrow n + 1$ 
15:     $i \leftarrow i - 1$  //  $i$  was incremented one extra time
16:  end if
17:  optimize sequence: Move  $(\alpha_i, \mathbf{x}_i)$  to front
18:  if  $n > N$  then
19:    Delete  $(\alpha_i, \mathbf{x}_i)_{i > n}$ 
20:     $n \leftarrow N$ 
21:  end if
22:   $t \leftarrow t + 1$ 
23: end loop
```

At line 17, if $f_t(\mathbf{x}_t) = \text{not-novel}$, i indexes the example that broke the novelty threshold. Otherwise, i indexes \mathbf{x}_t .

as novel. Instead, we propose a modified anytime version of our algorithm that ensures efficient and bounded computation (see Algorithm 2).

This algorithm takes advantage of the fact that familiarity contribution to new queries is often dominated by only a few examples. First, we can easily gain some efficiency by only processing stored examples until we have reached the novelty threshold (line 7). The key performance improvement, however, comes from the sequence optimization in line 17. For each prediction, the stored example that breaks the novelty threshold γ , or the new novel example itself, is moved to the front of the list as it is more likely to impact future queries². This is a slight variation of the traditional problem of dynamically maintaining a linear list for search queries for which the move-to-front approach was proven to be constant-competitive, meaning no algorithm can beat this approach by more than a constant factor [22]. As well as allowing us to bound the number of stored examples (line 19), this gives our algorithm an anytime property by enabling it to as quickly as possible classify as much of the environment as possible as not novel. When this algorithm is unable to run to completion due to time constraints,

²Another variant is to move stored example $\argmax_{j \in [1, i]} k(\mathbf{x}_t, \mathbf{x}_j)$ to the front of the list.

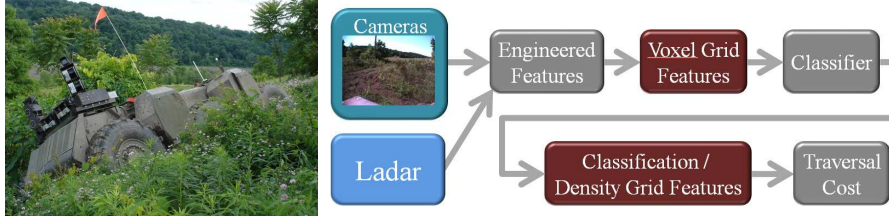


Figure 3: Robot used for novelty detection testing (left) and a high-level illustration perception system data flow (right). Features for novelty detection are taken from the steps highlighted in red.

it will fail intelligently by generating false positives but never potentially dangerous false negatives.

4 Application to Mobile Robotics

A natural application of our algorithm is to online novelty detection for a mobile robot. The Crusher UGV of the UPI Program (shown in Figure 3) that was used throughout our tests is intended for operation in complex, outdoor environments, performing local sensing using a combination of ladar and camera sensors [23]. The perception system assigns traversal costs by analyzing the color, position, density, and point cloud distributions of the environment [24, 25]. A large variety of engineered features that could be useful for this task are computed in real-time (see Figure 4) and the local environment is segmented into columns of 20 cm^3 voxels in order to capture all potentially relevant information (see Figure 5). Each voxel (tagged with its corresponding features) is passed through a series of classifiers and combined with additional density-related features to create a more compact set of intermediate features more suitable for traversal cost computation. The system then interprets these features through hand-tuned or learned methods to create a final traversal cost for that location in the world that can be used for path planning purposes.

To perform novelty detection we used subsets of the initial raw features as well as the intermediate classification and density features for each voxel. This vertical voxelization approach is effective for mobile robots since the presence of specific features at certain vertical positions are highly relevant to their impact on traversal cost. For example, solid objects at wheel height are likely to be small rocks while similar features higher off of the ground are more likely to be trees or man-made objects. Similarly, such spatial information is vital to effective novelty detection. This forced us to deal with a relatively high-dimensional feature space (49 features) as well as with the associated issues described earlier.

We deal with this problem by using MDA with an extensive library of hand-labeled examples across many environments and conditions to compute a lower dimensional subspace more suitable for density estimation as described in the previous section. Of the available classes, four were used to construct a three-dimensional subspace:

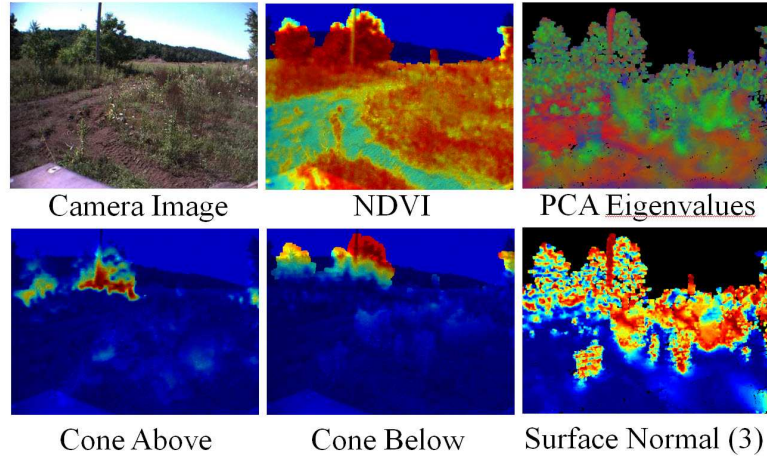


Figure 4: Example raw engineered features from the UGV's perception system used by the novelty detection algorithm. NDVI (normalized difference of vegetation index) is a useful metric for detecting vegetation.

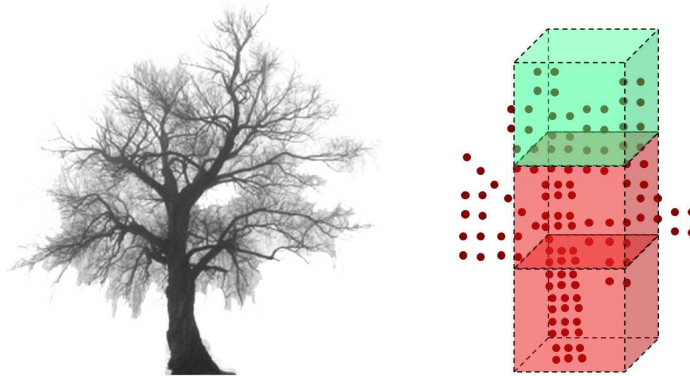


Figure 5: Illustration of the perception system's voxelization of vertical columns within the environment and subsequent classification. The voxels here are actually much smaller within the system but are enlarged for demonstration purposes. In the perception system, each voxel is a 20 cm^3 cube and due to the size of the vehicle, 10 voxels in the vertical direction are computed at each location in order to include all potentially relevant information.

road/dirt, rocks, bushes and barrels (see Figure 6). A fifth class of examples corresponding to various non-barrel man-made objects was withheld to verify the suitability of this subspace (see Figure 7).

Figure 2 shows the projection of all five classes onto the first three basis vectors computed by PCA and LDA using the first four classes³. The LDA projections clearly show better separation between the new set of man-made examples and the original four classes. As expected, the most overlap occurs with the barrel class as barrels share common properties with other man-made objects such as smooth surfaces, colors, etc. Since we would desire these new examples to be identified as novel relative to the rest of the classes, this separation implies that this is a more suitable subspace for use as a similarity metric within a novelty detection system.

Because our algorithm is efficient for online use, the novelty model can start uninitialized or can be seeded with a sampling of examples used during training so that it can identify areas that are novel and potentially unsafe to handle with the current perception system.

5 Experimental Results

Our novelty detection algorithm (with query optimization) was tested using our large UGV on an a natural outdoor environment to evaluate its online novelty detection performance (the algorithm ran in real-time on logged data). The test environment traversed by the robot consisted of combinations of road, grass and dirt, a large variety of vegetation, a series of small barrels, several ditches, large heavily-sloped piles of rocks and a long chain-link fence.

We projected all examples into the three-dimensional subspace generated by MDA as described in the previous section from the first four hand-labeled classes (not using the non-barrel man-made objects class). To best exhibit the online novelty detection abilities of our algorithm, the model was initialized to contain no prior examples. As the environment was explored, perception system features were averaged into 0.8 cm^2 grid locations for use as online batches of examples. Those that were identified as novel relative to the current model (composed of everything previously identified as novel) were incorporated into the model as described earlier.

The vehicle’s initial environment consisted of fairly open terrain with some light scattered vegetation scattered on both sides. As expected, instances of such vegetation were detected as novel the first few times they were seen (see Figure 8).

The vehicle then encountered areas of much denser, larger vegetation. Initially, a majority of such vegetation was identified as novel with respect to previous inputs (see Figure 9). As the vehicle continued navigating through similar vegetation, the model adapted and no longer identified such stimuli as novel (see Figure 10). Figure 11 demonstrates this learning process through a series of overhead images of this initial environment, identifying all future locations that are novel with respect to the *current* model. Output is shown at three points in time: near the beginning of navigation, just before initial encounters with dense vegetation and after sensing a small amount of

³All features were initially rescaled to zero-mean, unit-variance.



Figure 6: Examples of hand labeled class categories (bush, road / grass, rock, tree trunk, tree branches, etc.)



Figure 7: Sample hand labeled examples in the 'other man-made' class used for validation of dimensionality reduction effectiveness. This category excluded instances of barrels which were used as a separate class.

dense vegetation. It is clearly visible how the system adapts quickly, causing future similar instances to no longer be flagged as novel.

Proceeding through the environment, the vehicle then encounters a series of plastic barrels (see Figure 12). As desired, the first several appear as novel with respect to the large variety of vegetation previously seen while later barrels are no longer novel due to their strong similarity to the initially seen barrels. Similarly, a long stretch of a chain-link fence is identified as novel late in the course (see Figure 1). Again, the initial portions of the fence triggered the novelty detection algorithm while later portions were no longer novel due to the algorithm's adaptation. Additional examples of novel instances identified during traversal appear in Figure 13.

Overall, the novelty detection algorithm was able to identify all major unique objects (vegetation, barrels, fence, etc.) with a relatively small amount of false positives



Figure 8: After initialization with no prior novelty model, various small vegetation was detected as novel (identified in red).

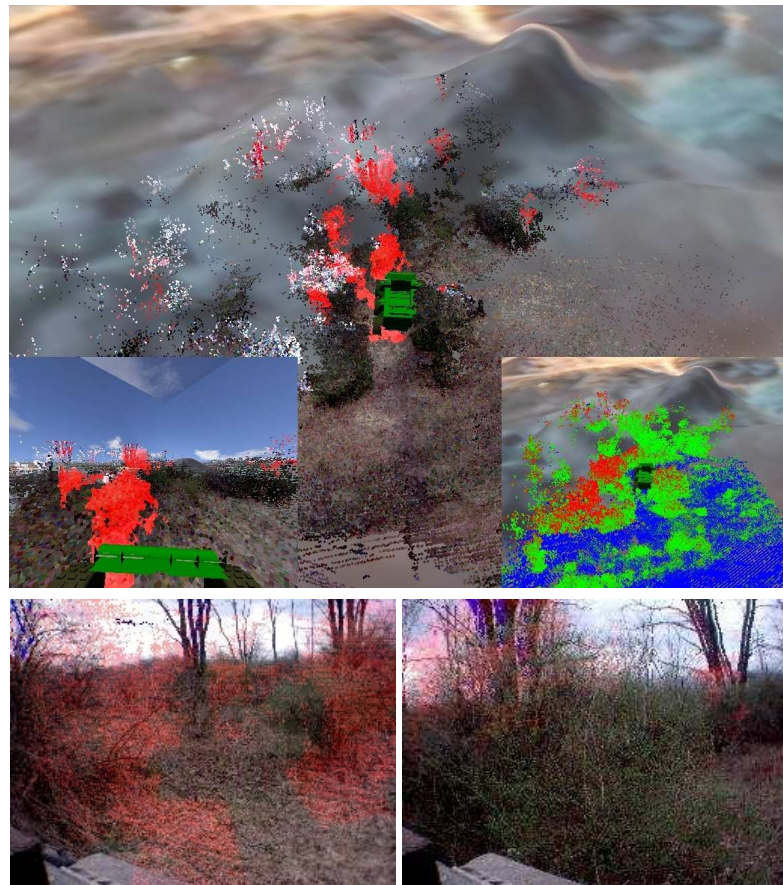


Figure 9: Initial encounter with larger and denser vegetation results in a significant amount of detected novelty (identified in red).

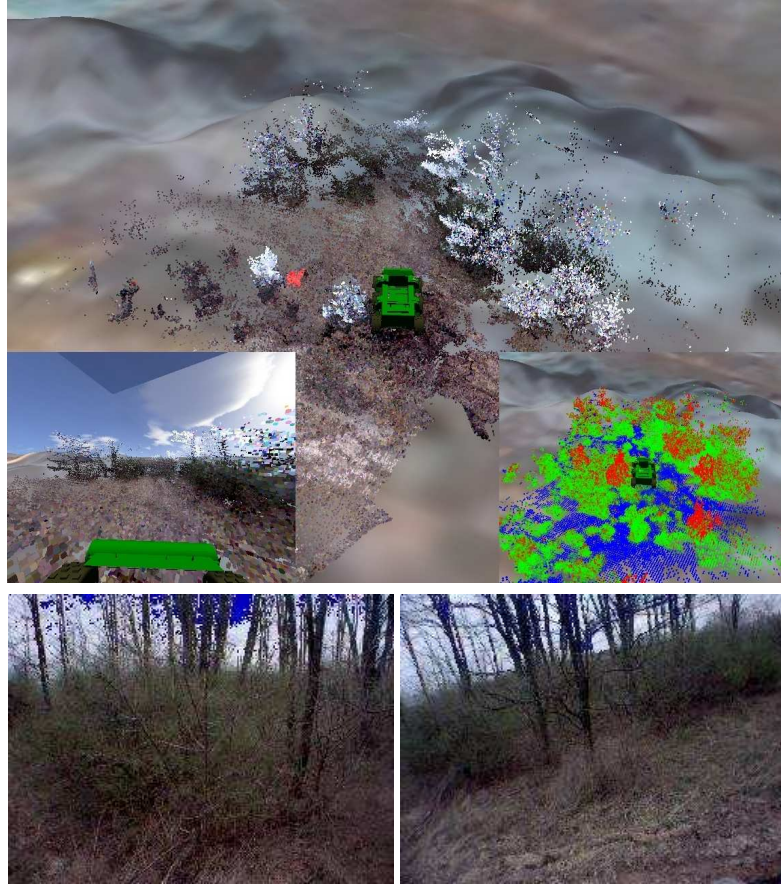


Figure 10: Similar vegetation as that shown in Figure 9 encountered a short time later. Notice how almost all vegetation is no longer novel due to similarity to previous stimuli.

due to effective adaptation to the environment. When PCA was used to create the feature subspace, the lack of separability between classes resulted in either unacceptably many false positives or false negatives, depending on parameter choices. As with any algorithm, the success of this approach is heavily dependent on the quality of features.

Computation time comparisons between the two algorithms on this course highlight the effectiveness of query optimization (see Figure 14). While the average computation time required per novelty query using Algorithm 1 grows with the number of stored examples, Algorithm 2 experiences temporary spikes in computation time as novel areas are encountered but query optimization allows the algorithm to quickly adapt its ordering of examples in order to maintain a bounded computation throughout navigation and allow effective anytime novelty prediction.

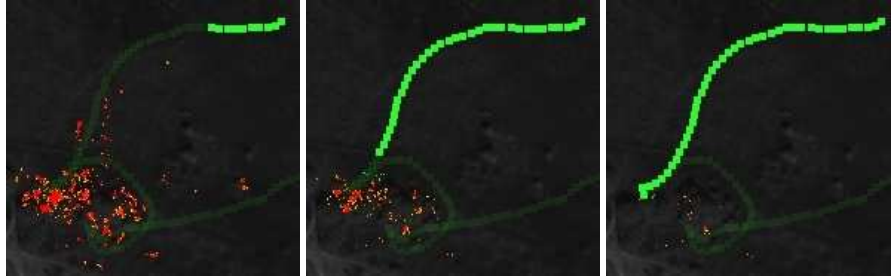


Figure 11: Novelty of all future perception input using current novelty model on vegetation-heavy terrain shown in Figures 8, 9 and 10 at three points throughout traversal. Robot’s past and future path is shown in light and dark green respectively and novelty of terrain is indicated by a gradient from yellow (moderately novelty) to red (high novelty). Robot is initialized without a prior novelty model.

6 Conclusion

Our algorithm addresses two significant limitations of most novelty detection approaches. By using MDA for supervised dimensionality reduction rather than unsupervised techniques such as PCA, this algorithm operates on a subspace that is more conducive to viewing novelty as a distance metric and is therefore more resistant to many of the issues associated with high-dimensional feature spaces. Additionally, this algorithm’s adaptive abilities, computational bounds and anytime properties make it a logical choice for many online novelty detection tasks. As robotic systems continue to improve, such approaches can help capitalize on their abilities by acting as a safeguard against the inevitable dangers of unfamiliar situations.

Acknowledgment

This work was partially sponsored by DARPA under contract Unmanned Ground Combat Vehicle - PerceptOR Integration (contract number MDA972-01-9-0005) and by the U.S. Army Research Laboratory under contract Robotics Collaborative Technology Alliance (contract number DAAD19-01-2-0012). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

Boris Sofman is partially supported by a Sandia National Laboratories Excellence in Engineering Fellowship. The author gratefully acknowledges Sandias Campus Executive Laboratory Directed Research and Development Program (LDRD) for this support. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energys National Nuclear Security Administration under contract DOE. No. DE-AC04-94AL85000.

We would also like to thank all the members of the UPI project team, without whom this work would not be possible. Particular thanks go to David Bradley, David Silver, Cristian Dima, Cliff Olmstead and Thomas Pilarski.

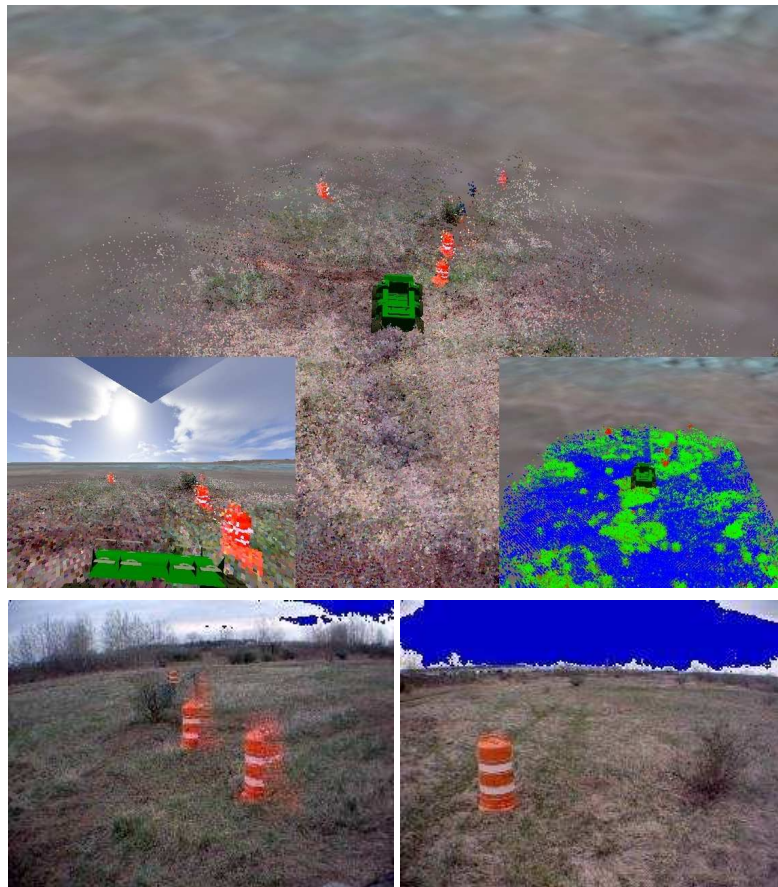


Figure 12: Series of barrels encountered later in the course. The initial barrels are detected as novel (red shade) even after significant exposure to a large variety of vegetation (top and left). Later barrels are no longer identified as novel due to online training.



Figure 13: Additional examples of novel instances identified during later traversal (red shade): first encounter with a ditch (left) and a large, heavily-sloped pile of rocks (right).

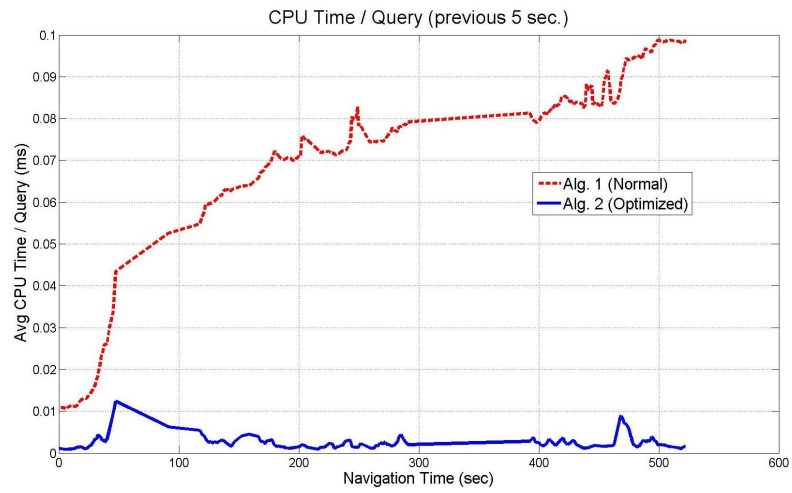


Figure 14: Average computation in milliseconds per novelty query on 3.2 GHz CPU for Algorithm 1 (dashed red line) and Algorithm 2 (solid blue line) over the previous 5 seconds throughout navigation. Computational complexity of Algorithm 2 remains bounded due to the order optimization step (line 17). These timings do not include feature computation and projection costs as they are identical under both algorithms.

References

- [1] C. Urmson, J. Anhalt, D. Bagnell *et al.*, “Autonomous driving in urban environments: Boss and the urban challenge,” *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.
- [2] A. Kelly, A. Stentz, O. Amidi *et al.*, “Toward reliable off road autonomous vehicles operating in challenging environments,” *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 449–483, 2006.
- [3] S. Thrun, M. Montemerlo, H. Dahlkamp *et al.*, “Stanley: The robot that won the darpa grand challenge,” *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, June 2006.
- [4] N. D. Ratliff, D. Silver, and J. A. Bagnell, “Learning to search: Functional gradient techniques for imitation learning,” *Autonomous Robots, special issue on Robot Learning*, 2009.
- [5] J. Kivinen, A. J. Smola, and R. C. Williamson, “Online learning with kernels,” in *NIPS*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. MIT Press, 2001, pp. 785–792.
- [6] K. Worden, “Structural fault detection using a novelty measure,” *Journal of Sound and Vibration*, vol. 201, no. 1, pp. 85–101, 1997.
- [7] P. Hayton, B. Schölkopf, L. Tarassenko *et al.*, “Support vector novelty detection applied to jet engine vibration spectra,” in *NIPS*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2000, pp. 946–952.
- [8] J. Ryan, M.-J. Lin, and R. Miikkulainen, “Intrusion detection with neural networks,” in *Advances in Neural Information Processing Systems*, M. I. Jordan, M. J. Kearns, and S. A. Solla, Eds., vol. 10. The MIT Press, 1998.
- [9] L. Tarassenko, P. Hayton, N. Cerneaz, and M. Brady, “Novelty detection for the identification of masses in mammograms,” in *Proceedings of the Fourth International IEEE Conference on Artificial Neural Networks*, vol. 409, 1995, pp. 442–447.
- [10] H. V. Neto and U. Nehmzow, “Visual novelty detection with automatic scale selection,” *Robotics and Autonomous Systems*, vol. 55, no. 9, pp. 693–701, 2007.
- [11] S. Marsland, U. Nehmzow, and J. Shapiro, “On-line novelty detection for autonomous mobile robots,” *Robotics and Autonomous Systems*, vol. 51, no. 2-3, pp. 191–206, 2005.
- [12] R. O. Duda and P. E. Hart, *Pattern Classification*. John Wiley and Sons, 2000.
- [13] B. Schölkopf, R. C. Williamson, A. J. Smola *et al.*, “Support vector method for novelty detection,” in *NIPS*, S. A. Solla, T. K. Leen, and K.-R. Müller, Eds. The MIT Press, 1999, pp. 582–588.

- [14] L. M. Manevitz and M. Yousef, “One-class SVMs for document classification,” *Journal of Machine Learning Research*, vol. 2, pp. 139–154, 2001.
- [15] C. Campbell and K. P. Bennett, “A linear programming approach to novelty detection,” in *NIPS*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. MIT Press, 2000, pp. 395–401.
- [16] D. M. J. Tax and R. P. W. Duin, “Support vector domain description,” *Pattern Recognition Letters*, vol. 20, no. 11-13, pp. 1191–1199, 1999.
- [17] H. Hoffmann, “Kernel PCA for novelty detection,” *Pattern Recognition*, vol. 40, no. 3, pp. 863–874, Mar. 2007.
- [18] N. Japkowicz, C. Myers, and M. A. Gluck, “A novelty detection approach to classification,” in *IJCAI*, 1995, pp. 518–523.
- [19] M. Markou and S. Singh, “Novelty detection: a review - part 1: statistical approaches,” *Signal Processing*, vol. 83, no. 12, pp. 2481–2497, 2003.
- [20] ———, “Novelty detection: a review - part 2: neural network based approaches,” *Signal Processing*, vol. 83, no. 12, pp. 2499–2521, 2003.
- [21] B. Schölkopf and A. J. Smola, *Learning with Kernels*. Cambridge, MA: The MIT Press, 2002.
- [22] D. Sleator and R. Tarjan, “Amortized efficiency of list update and paging rules,” *CACM: Communications of the ACM*, vol. 28, 1985.
- [23] A. Stentz, J. Bares, T. Pilarski, and D. Stager, “The crusher system for autonomous navigation,” in *AUVSI's Unmanned Systems North America*, August 2007.
- [24] D. M. Bradley, R. Unnikrishnan, and J. Bagnell, “Vegetation detection for driving in complex environments,” in *ICRA*. IEEE, 2007, pp. 503–508.
- [25] J.-F. Lalonde, N. Vandapel, D. Huber, and M. Hebert, “Natural terrain classification using three-dimensional ladar data for ground robot mobility,” *Journal of Field Robotics*, vol. 23, no. 1, pp. 839 – 861, November 2006.