Improving Robot Navigation Through Self-Supervised Online Learning

Boris Sofman, Ellie Lin, J. Andrew Bagnell, Nicolas Vandapel, and Anthony Stentz
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
Email: {bsofman,elliel,dbagnell,vandapel,axs}@ri.cmu.edu

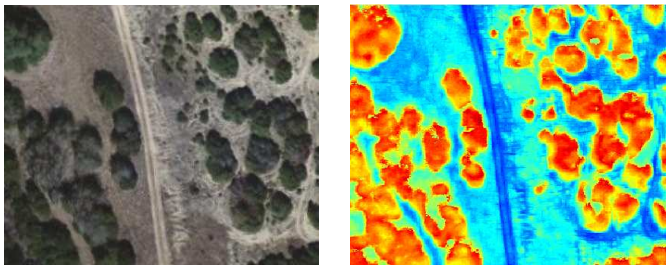
***Abstract*— In mobile robotics, there are often features that, while potentially powerful for improving navigation, prove difficult to profit from as they generalize poorly to novel situations. Overhead imagery data, for instance, has the potential to greatly enhance autonomous robot navigation in complex outdoor environments. In practice, reliable and effective automated interpretation of imagery from diverse terrain, environmental conditions, and sensor varieties proves challenging. We introduce an online, probabilistic model to effectively learn to use these scope-limited features by leveraging other features that, while perhaps otherwise more limited, generalize reliably. We apply our approach to provide an efficient, self-supervised learning method that accurately predicts traversal costs over large areas from overhead data. We present results from field testing on-board a robot operating over large distances in off-road environments. Additionally, we show how our algorithm can be used offline to produce a priori traversal cost maps and detect misalignments between overhead data and estimated vehicle positions. This approach can significantly improve the versatility of many unmanned ground vehicles by allowing them to traverse highly varied terrains with increased performance.**

I. Introduction

A common problem that arises in mobile robotics is that potentially powerful sensor data and features are often difficult to take advantage of because they are situation or location specific. For instance, camera imagery can potentially detect unpaved road in the desert significantly farther than some ladar-based systems can. Detecting such roads from a distance proved to be a crucial component in Stanford Racing's winning Grand Challenge entry [1]. Unfortunately, such data can prove very resistant to automated interpretation. In particular, classifiers that prove to be powerful indicators of road in a particular area often do not generalize to new conditions.

Similarly, outdoor robot navigation can benefit from the now widespread availability of high quality overhead imagery and elevation data from satellite and aircraft [2][3]. Presently, nearly the entire world has been surveyed at 1 m accuracy, with higher resolution (better than 25 cm accuracy) data available in certain areas. With this overhead data, many of the difficulties associated with autonomous robot operation are alleviated, even with the coarsest of terrain resolution. Systems can then dispense with myopic exploration and instead pursue routes that are likely to be effective.

Unfortunately, leveraging this tremendous resource on an autonomous robot proves difficult. Building systems that can reliably interpret overhead image data is far from easy. Additionally, such estimates must be well calibrated with other on-board perception estimates of the terrain, or system performance may suffer.



(a) (b)

Fig. 1. Sample results of terrain traversal cost predictions. 0.35 m resolution color overhead imagery used by our online learning algorithm (a) and corresponding predictions of terrain traversal costs (b). Traversal costs are color-scaled for improved visibility. Blue and red correspond to lowest and highest traversal cost estimates, respectively.

In this paper, we address the problem of learning and inferring between two heterogeneous data sources that vary in density, accuracy and scope of influence. The objective is to generalize from one data source, viewed as a reliable estimate, to be able to work with another, which may be high performance (e.g., long range or high accuracy) but difficult to generalize to new environments. We frame the problem as a simple, linear probabilistic model for which inference results in a self-supervised online learning algorithm that fuses the estimates from the two data sources. We discuss the advantages of this framework including reversible learning, feature selection, data alignment capabilities, reliable use of multiple estimates, as well as confidence-rated predictions.

Furthermore, we demonstrate the approach in the context of long range navigation of a large, unmanned ground vehicle during various field tests in complex natural environments. As it traverses an environment, the vehicle utilizes its on-board perception system and overhead imagery and elevation data to learn the mapping from overhead data features to computed terrain traversal costs in order to predict traversal costs elsewhere in the environment where only overhead data is available, effectively extending the range of the vehicle's local perception system and allowing more effective navigation

of the environment. This approach removes the necessity of human involvement and parameter engineering, which limits the versatility of many robotic systems.

II. Approach

A. Formalization

We approach the problem of leveraging the powerful, but difficult to generalize, features in a Bayesian probabilistic framework using the notion of scoped learning [4]. The scoped learning model admits the idea of two types of features: "global" and "local". Global features are generally useful, and their predictive power extends well to new domains, while local ones, which, although often very powerful, typically generalize poorly and are more difficult to take advantage of in a consistent way. These local features have *scope* that is limited to one particular domain. We wish to apply our system to extend the scope of such features to many possible domains. For our canonical problem of learning to leverage the extended range of overhead data, these names may prove counter-intuitive, so we refer to them instead as *general* and *locale-specific* features. Features generated from dense, vehicle-based ladar perception serve as our general features, while features generated from overhead based imagery and elevation data serve as our locale-specific features. The latter are particularly valuable to mobile robots because of their extended range and widespread availability.

1) Model: The scoped learning approach is a very simple probabilistic model (shown graphically in Figure 2) that captures this notion of features that have scope. The outer plate L represents in graphical model notation that there are independent locales in which the model will be applied [5]. These correspond to new areas of the world in which our robot will operate.

Within the plate, we see a sequence of locale-specific features and corresponding general feature-based estimates. At each point in the sequence, we wish to make predictions about c (either all or a subset of them.) Here, c is the true variable we wish to predict, and $\tilde{c}$ is an estimate of that variable coming from the general features, while x are our locale-specific features. The parameters β common to the locale (plate) capture the relationship between locale-specific features and the variables of interest c. The length of our sequence is n.

This learning model captures the idea of self-supervised learning [6] in a Bayesian framework and extends the idea to integrate both the general feature-based estimates and the self-supervised locale-specific estimates. Driven by our application, we are particularly interested in the online *regression* case[1] where the goal is to infer the true continuous values c_i in an online fashion as general feature-based estimates $\tilde{c}_i$ become available. We choose a simple model for c as a function of the k locale-specific features $x = (x^1, \ldots, x^k)$ by modeling the distribution for c given x as a Gaussian with mean a linear function of x and with a variance of σ_l^2 :

$$E(c|\beta, x) = \beta^T x \tag{1}$$

We assume that the estimates from the general feature-based predictors have Gaussian noise and thus are distributed:

$$\tilde{c} \sim \text{Normal}(c, \sigma_g^2)$$

We take the σ_g and σ_l to be hyper-parameters lying outside the locale-specific plate.

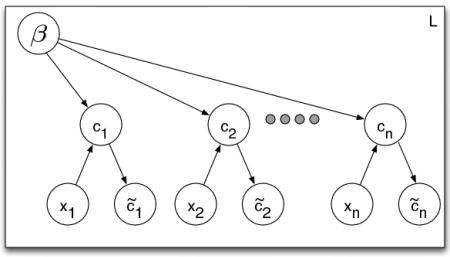


Fig. 2. Graphical depiction of the scoped learning model. Hyper-parameters, including priors on the locale-specific parameters β and noise variances, lie outside the plate indexed by L and are not depicted.

2) Inference: We develop the inference for the model in an online fashion. Given a new data point $\tilde{c}_i$ estimating the true variable c_i, our goal is to compute new estimates[2] of the variables c_j, assuming we have already seen data $D = \{\{x\}_{1\ldots n}, \{\tilde{c}\}_{1\ldots i-1}\}$. We can compute this by integrating over the uncertain parameters β which describe the relationship between the true variable and the local features.

$$p(c_j|\tilde{c}_i, x_i, D) = \int d\beta \; p(c_j|\beta, \tilde{c}_i, x_i) p(\beta|\tilde{c}_i, x_i, D)$$

We can compute the required distribution over β as:

$$p(\beta|\tilde{c}_i, x_i, D) \propto p(\beta|D) \int dc_i \; p(\tilde{c}_i|c_i) p(c_i|\beta, x_i)$$

In our linear-Gaussian model, this can be understood as revising the posterior distribution from $p(\beta|D)$ in light of a Gaussian likelihood that takes into account noise from both general and locale-specific features.

Our computation of the posterior distribution $p(\beta|\tilde{c}_i, x_i, D)$ is as follows. We first initialize our distribution to the prior distribution $p(\beta)$. Then, for every training example i, we multiply our distribution by $p(\tilde{c}_i|\beta, x_i)$. Since the prior distribution and $p(\tilde{c}_i|\beta, x_i)$ are normal, the posterior distribution is also normal. We use the notation $\hat{\beta}$ to represent the mean of the posterior distribution and V_β to represent the variance. Thus, computing $p(\beta|\tilde{c}_i, x_i, D)$ is performing a self-supervised learning using a Bayesian linear regression model with noise variance $\sigma_l^2 + \sigma_g^2$.

We use our current estimate of the posterior distribution when we want to predict a future outcome c_j. We are interested

[1]The original scope learning work [4] was developed in the context of classification using discrete features, generative descriptions of those features, and in batch.

[2]We assume that our prior on β is a priori independent of the features x so that inference will remain the same even in the case where the features become available in some sequence.

in predictions in two cases: first, when we have no general feature-based estimate $\tilde{c_j}$ for a particular c_j, and second, when such an estimate is available. In the first case, the predictive distribution $p(c)$ has mean $c_p = x^T\hat{\beta}$ and variance $\sigma_p^2 = \sigma_l^2 + \tilde{x}^T V_\beta \tilde{x}$ [7]. When we also have an estimate $\tilde{c_j}$, inference combines these two estimates:

$$p(c_j) = \text{Normal}(\sigma_p'^2(\frac{c_p}{\sigma_p^2} + \frac{\tilde{c_j}}{\sigma_g^2}), \sigma_p'^2)$$

where

$$\sigma_p'^2 = \frac{1}{\frac{1}{\sigma_p^2} + \frac{1}{\sigma_g^2}}.$$

We note that it is possible to compute the posterior distribution in batch, but we prefer to maintain an estimate of the posterior distribution as we receive general feature-based cost estimates so that we may immediately apply our algorithm to new data.

B. Advantages of the Bayesian Learning Approach

Using the online Bayesian scope learning model provides a number of important benefits.

1) Confidence Rated Prediction: The variance estimate provided by our algorithm for the probability of each c can be used as a metric of confidence in the prediction. If a situation arises in which we must choose which one of several predicted outcomes to trust, we could simply use the one with the smallest variance.

2) Learning of the Hyper-Prior and Feature Selection: Our algorithm depends on a number of hyper-parameter terms that may be chosen based on data from multiple locales. We discuss ways to choose the noise variance terms σ_l and σ_g in section II-A and the prior distribution on parameters β in section III-A. It is important to note that the prior on β may also be chosen as hyper-parameters on multiple locales by adapting Tipping's sparse hyper-parameter re-estimation procedure [3] to our setting [8]. In this way, we can both automate feature selection and bias our algorithm to prefer certain features for new locales.

3) Reversible Learning: A problem that often arises in online learning is the handling of multiple estimates of a particular quantity. For instance, in our canonical example, our general feature-based estimates $\tilde{c}_i$ may improve as we get closer and denser laser readings of the terrain. It is not appropriate to treat these as independent training examples: while they may differ in their variance, they are generally highly correlated. Neither is it useful to simply take the first estimate available: often this is a poor substitute for all the data. In our model, since we maintain an exact posterior distribution that lies inside the exponential family, we may effectively *remove* the effects of training on a data point by dividing out the likelihood term we had used to include it in the posterior [7]. In this way, we always have an estimate of the posterior distribution of β using the current best estimate $\tilde{c}_i$. Minka has developed an alternate use of this "removal trick" for approximate inference [9].

[3]The procedure finds the ML-II estimate for the precision (inverse variance) of each weight β_i.

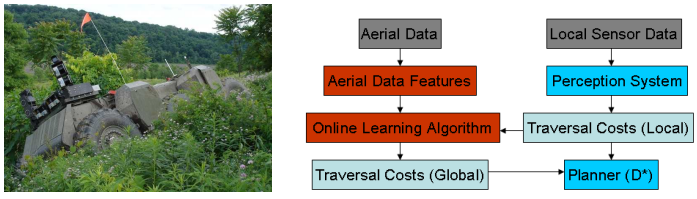
III. Application to Ground Robotics

A. Context

A natural application of our algorithm is to the improvement of autonomous navigation capabilities for unmanned ground vehicles. Our algorithm lends itself well to addressing many of the issues that arise due to the diversity in the environments and data that robotic systems must encounter. In many robotic systems, variables most relevant to an unmanned ground vehicle, such as terrain traversal cost, are computed directly by the vehicle's on-board perception system through the processing of high density ladar data gathered by on-board sensors. Techniques to process such data are often generalizable to many environments so that the vehicle's perception system does not require much adjustment when dealing with new terrain, yet the limited range for this type of data source is a major shortcoming. While data sources such as overhead imagery are widely available with high accuracy, developing fixed techniques to process such locale-specific data sources is difficult because even though they may be extremely useful in specific areas, they do not generalize well to new domains due to variations in terrain, lighting conditions, weather, and even time of data gathering.

We demonstrate how our algorithm can be applied to the domain of mobile robotics in order to derive the most benefit from the availability of both general and locale-specific data sources without any human involvement. The performance of our algorithm is evaluated through actual field testing in a complex off-road environment on-board a rugged, all-terrain unmanned ground vehicle. Our results show how combining the adaptive performance of our algorithm with the inherent mobility of such a vehicle leads to more efficient navigation of complex environments. Additionally, we show how our algorithm can be used to detect misalignments between overhead data and the vehicle's estimated position, a common problem in many such robotic systems.

For the results in section IV, we chose the hyper-parameter for noise variance σ_l^2 with ML-II and chose an isotropic Gaussian with high variance for the prior on β [7].



(a) (b)

Fig. 3. Robot used for all field tests in its typical operating environment (a) and an illustration of how the online learning algorithm runs on-board the robot (b). Algorithm learns mapping from local-specific overhead data features to locally computed terrain traversal costs (computed from general features) to make prediction elsewhere in the environment.

B. Terrain Traversal Cost Prediction

Our robot performs local sensing using ladar sensors, assigns traversal costs to the environment from features com-

puted by interpreting the position, density, and point cloud distributions of sensed obstacles (these features generalize across most domains and therefore serve as our *general* features as defined in Section II-A). The robot re-plans in real-time by finding minimum cost paths through the environment using the D* algorithm (see Figure 3) [10]. We demonstrate how our algorithm can learn to predict terrain traversal costs computed by the on-board perception system of our unmanned ground vehicle from overhead data. We also compare the predictive performance of our algorithm to that of a hand-trained classifier using superior data sources. We chose to predict traversal cost rather than intermediate results such as slope, density, or presence of vegetation because traversal cost is the metric that most closely governs a vehicle's navigation strategy through an environment. Our robot's perception system is proficient at effectively assessing terrain traversal costs, so it is desirable to be able to mimic its predictive abilities. We will therefore use estimates from the robot's perception system to evaluate the accuracy of traversal cost predictions.

The characteristics of an environment change with varying conditions. However, even outdated data can be useful since most distinct areas in an environment will maintain uniformity in their characteristics despite these variations. By relaxing restrictions on the recency of overhead data, our algorithm further increases its impact on improving robot navigation. Overhead data is relatively inexpensive and available at various resolutions for the entire world, so as the quality of global surveying improves, the applications of such research will greatly expand.

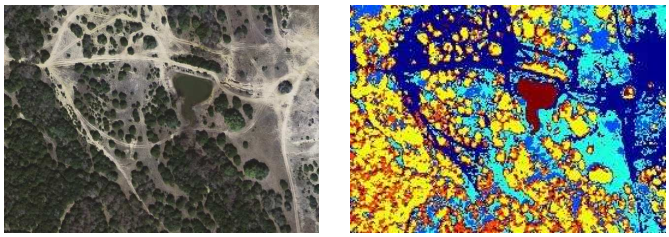
C. Features

A set of feature maps for the vehicle's environment was generated from each overhead data source for use as inputs to the algorithm (these are our *locale-specific* features as defined in Section II-A). In our implementation, HSV (hue, saturation, value) features were used to represent color imagery data while the pixel intensity of the black and white imagery data was used as a single feature. Raw RGB (red, green, blue) color data was inadequate for our approach due to its sensitivity to illumination variations.

A feature containing the maximum response to a set of Gabor filters of various orientations centered at each pixel was also generated to capture texture in each type of imagery. Additional features for the black and white imagery data were generated by computing the means and standard deviations in intensity within windows of various size around each pixel. Additional elevation-based features (similar to those described in [3]) were computed when such data was present. All features were rescaled to the $[-1, 1]$ range and a constant features was also included.

Finally, clustering of all previously computed features was performed that allowed the algorithm to identify patterns in the feature input space that are relevant to the output regression. The Gaussian Mixture Model algorithm was chosen to cluster the input data because of its ability to generate membership features by assigning each data point a fractional degree of membership in each output cluster (see Figure 4). Six clusters were used in our implementation.

Similar techniques may be used to generate features from any combination of data sources gathered through a variety of methods. We point out that our approach is quite robust to the number of clusters and the removal of features.



(a) (b)

Fig. 4. Sample clustering results from using the Gaussian Mixture Model algorithm on generated features. Overhead color imagery data (a) used to generate features and resulting clustering into six clusters (b). Membership features were generated by computing the fractional degree of membership of each pixel in each cluster.

D. Training and Prediction

Traversal cost is difficult to quantify, so choosing appropriate values often requires careful engineering. In order to produce desired behavior when used with a path planning algorithm such as D*, traversal costs for undesirable areas such as heavy vegetation must be higher than traversal costs for ideal areas such as roads (our robot works with traversal costs in the range of 16 to 65535). For example, the robot's on-board perception system assigns traversal costs of 16 (the minimum) to roads while grass is assigned a traversal cost of 48, implying the robot would be willing to take a detour of three times the distance in order to stay on a road as opposed to driving over grass. Meanwhile, dense vegetation is often assigned traversal costs of over 10000 in order to encourage the robot to traverse elsewhere except under extreme necessity. Because of these traversal distance ratios, errors in traversal cost estimates in low-cost areas are more detrimental than similar errors in high-cost areas. An error of 100 to an area of extremely high traversal cost would have negligible effect, while the same error at an area of desirable terrain would radically change the behavior of the robot.

In order to work with a linear model, we deal with traversal costs within our algorithm on a logarithmic scale, converting from the normal traversal cost space for the purposes of training and prediction. The Gaussian error assumption embedded in our probabilistic model is a much better approximation when we measure error on this scale. Unlike in the regular traversal cost space, small errors in the log space lead to small errors in the traversal distance ratios.

Training examples are constructed from a vector of overhead imagery feature values (x_i) and the average of all traversal cost estimates that have been calculated within the corresponding area ($\tilde{c}_i$). As with many robotic systems, the performance of our robot's on-board perception system quickly degrades as the

distance from the robot increases (due to the lowered accuracy and density of sensor data), so the quality of a training example is measured by its proximity to the robot. Rather than struggling to decide at which point to utilize an example for training, the reversible learning capabilities of our algorithm allow us to maintain an optimal level of predictive abilities by ensuring that only the highest quality data available impact its state. As the robot approaches locations that had previously been used for training, obsolete examples are *unlearned* in favor of higher quality training examples available for those areas. An example of this training process can be seen in Figure 5. Estimates greater than 12 meters from the robot are ignored since such estimates are unreliable and would only corrupt the quality of training in cases where they cannot be replaced with better estimates.

As the algorithm acquires more training data, its predictive performance improves, allowing it to revise previously made traversal cost estimates. The algorithm specifies a degree of confidence for each prediction based on the similarity of the example to past training data (as indicated by the variance estimate), so predictions in which the algorithm lacks confidence can be ignored in favor of an alternative source of predictions or a default value (see Figure 5b).

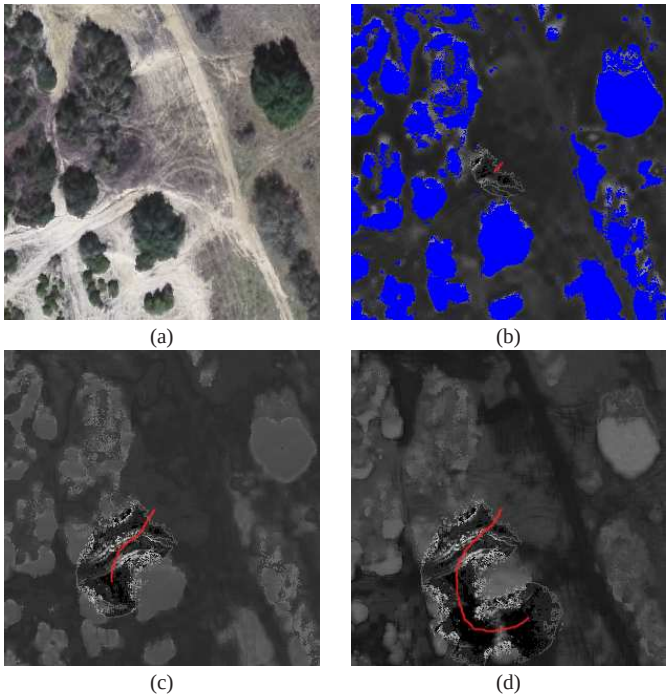


Fig. 5. Training progress of online learning algorithm using overhead color imagery data for traversal of environment shown in (a) is shown in (b) - (d). Dimensions of shown areas are 150 m $\times$ 150 m. Accumulated ground truth traversal costs computed by robot's on-board perception system and vehicle path (shown in red) are overlaid on estimated traversal costs generated by the algorithm. Lower costs appear as darker colors and predictions that the algorithm lacks confidence in (due to insufficient representative training examples) are shown in blue.

E. Applications of Trained Algorithm

This algorithm can be used in a variety of ways to aid in unmanned ground vehicle navigation, both in real-time on-board a robot and offline once it has been trained. In the case of online terrain traversal cost prediction, the algorithm can be used to periodically update traversal cost estimates within a region around the robot so that a real-time path planning algorithm such as D* can revise the vehicle's global path to account for the changes. As shown in the following section, the use of this algorithm to extend the robot's field of view results in significantly shorter and more intelligent paths.

Since the predictive state of the algorithm can be captured at any time by the vector β at that moment, one can make traversal cost predictions for a large area without ever having to traverse or acquire training examples from that area beforehand. Instead, as long as identical features are computed for the two areas, one can simply drive through a representative area for a short period of time in order to train the algorithm to make predictions in a much larger area. The following section will also show that a priori traversal cost maps produced by this technique are more accurate than even those produced from hand-trained techniques that utilize superior data sources.

Even slight mis-registration between overhead data and the estimated position of the robot can significantly hinder the performance of algorithms such as ours that are sensitive to such errors. An advantage of using an online Bayesian linear regression model is the ability to detect such misalignments.

When predicting a new traversal cost c_j, the model creates a predictive distribution $p(c)$ with a mean μ_p and a variance σ_p^2. Evaluating the predictive distribution at the traversal cost c_i of a training example gives the probability of having seen that traversal cost given its corresponding feature vector. We can use the probability of having seen all of our data, $p(\tilde{c}_1, \ldots, \tilde{c}_n)$, to detect map misalignments between overhead data sources and estimated vehicle positions by searching through a space of potential alignments for the one that maximizes the probability of the data.

Since $p(\tilde{c}_1, \ldots \tilde{c}_n)$, can be computed via the chain rule as the product of the predictive distributions evaluated at every $\tilde{c}_i$ used for training, an estimate of the log data probability is the cumulative sum of $-\log \sigma_p - \frac{(y-\mu_p)^2}{\sigma_p^2}$ for every predictive distribution. After all examples have been received, the average log data probability over all training examples can be used to compare the quality of an alignment against other alignments. As shown in the following section, correcting such misalignment produces traversal cost maps with better defined obstacles that more accurately reflect the true environment.

IV. Results

A. Field Test Results

The algorithm was tested in real-time on-board our unmanned ground vehicle to measure its impact on navigation performance. The test environments contained a large variety of vegetation, various-sized dirt roads (often leading through narrow passages in dense vegetation), hills, and

ditches. The vehicle traversed a set of courses defined by series of waypoints first by using only its on-board perception system for navigation and also with the help of our online learning algorithm with 40 cm resolution overhead imagery and elevation data to supplement the on-board perception system with traversal cost updates computed within a 75 m radius once every 2 seconds. The algorithm was initialized for each course with no prior training (see Figures 6 and 7 for sample results). As shown in Table I, our algorithm allowed the vehicle to complete the courses in 26.94% less time while traversing 7.38% less distance. We found that with our algorithm, the vehicle chose to drive further distance on more preferable terrain in order to avoid difficult or dense areas. Most importantly, while we were forced to manually intervene during the tests with only the perception system in order to correct the vehicle's heading when it became trapped in heavy vegetation and could not escape on its own, no manual interventions were necessary when using our algorithm.

TABLE I

STATISTICS FOR COURSE TRAVERSALS WITH AND WITHOUT THE ONLINE LEARNING ALGORITHM

	Without Algorithm	With Algorithm
Total Traversal Time (sec)	1369.86	1000.82
Total Distance Traveled (m)	1815.71	1681.73
Average Speed (m/s)	1.33	1.68
Number of Interventions	1	0

Fig. 6. Comparison of paths executed by our unmanned ground vehicle for shown course when using only on-board perception (in solid red) and with our online learning algorithm used in real-time on-board the robot (in dashed blue). Notice how the online learning path quickly learns to avoid the difficult area near the cul-de-sac and instead chooses to follow the road to the goal.

B. Field Test Data Post-Processing Results

The algorithm was also used to produce a priori traversal cost maps for a multi-kilometer course over a large area of complex terrain with heavy vegetation and elevation obstacles. The algorithm was trained for about 7 minutes using overhead imagery data by driving the vehicle by remote control through

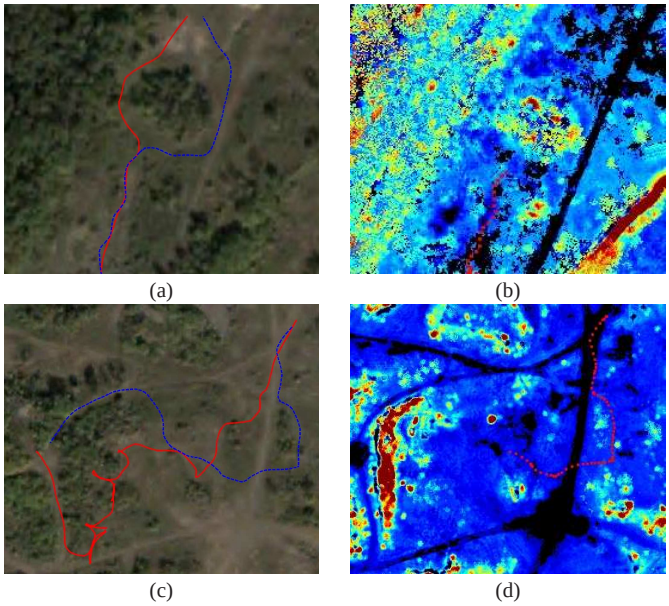


Fig. 7. Comparison of paths executed for shown situations when using only on-board perception (in solid red) and with our algorithm (in dashed blue) are shown in (a) and (c). Predictions of terrain traversal costs for the environment by our algorithm at the time the vehicle chose to avoid the large obstacles in front of it are shown in (b) and (d). Traversal costs are color-scaled for improved visibility. Blue and red correspond to lowest and highest traversal cost areas, respectively, with roads appearing in black. In (a) the online learning path chose to travel slightly further on road in order to avoid the difficult passage to the left and in (b) our algorithm helped the vehicle avoid the cul-de-sac by executing a path that is 42.89% shorter in 72.62% less time.

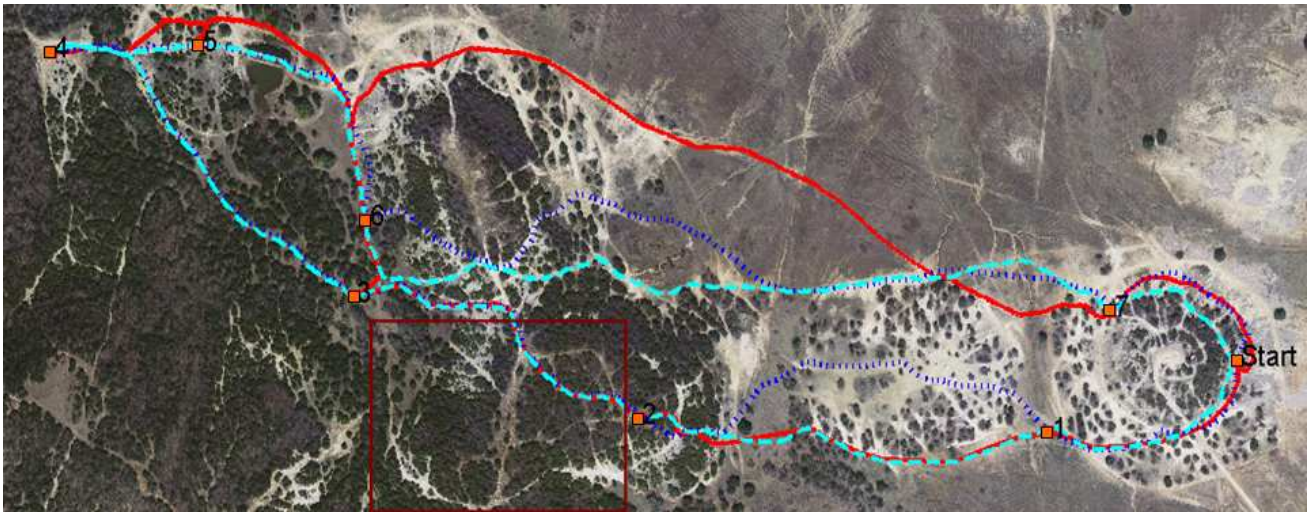
an entirely separate area that was similar to the main course. The trained algorithm was then used off-line to generate a traversal cost map and plan an initial path through the much larger course. Closeups of generated traversal cost maps and resulting planned paths can be seen in Figure 8. For comparison, we also included the resulting path from a traversal cost map generated by a supervised learning algorithm with human-picked examples from the actual course and features generated from both overhead imagery and high-density elevation data (see Table II for a description of data sources) [3].

TABLE II

TYPES OF OVERHEAD DATA USED BY ONLINE LEARNING (OLL) AND HAND-TRAINED ALGORITHMS USED TO PRODUCE PRIOR COST MAPS

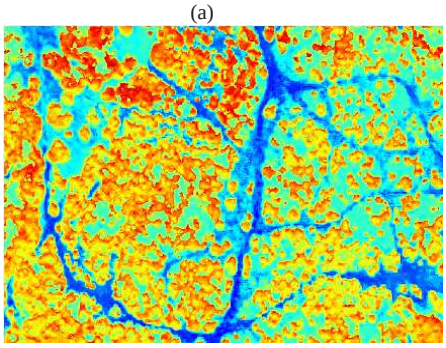
Algorithm	Data Used	Resolution
OLL (color)	Color imagery	$0.35\ m$
OLL (B & W)	Terraserver B & W imagery	$1.0\ m$
Human-Supervised	Color imagery Elevation	$0.35\ m$ $< 0.2\ m$

We evaluated the performance of our online learning algorithm against the human-trained technique by accumulating all the traversal costs generated by the vehicle's on-board perception system during a traversal of the course shown in Figure 8 and comparing those costs to the estimates from each of the generated prior cost maps. The average absolute error in traversal cost (on a log scale as described earlier) for each

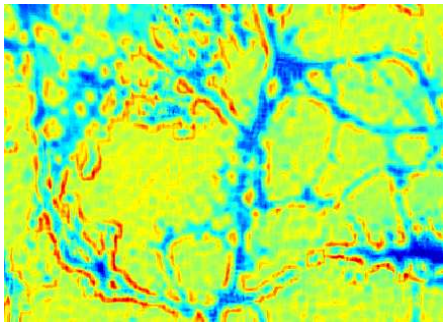


(a)

(b)



(c)



(d)

Fig. 8. Circular course with the GPS waypoints for which a priori paths were planned is shown in (a). Shown area is 2000 m $\times$ 750 m. Our algorithm was trained during a short traversal of a similar but entirely separate course. Paths generated by the human-trained algorithm (solid red), our algorithm using color imagery data (dashed cyan), and our algorithm using black and white imagery data (dotted blue) are shown. Traversal cost maps produced by our algorithm for the closeup area in (b) using overhead color imagery and black and white imagery are shown in (c) and (d) respectively. See Table II for description of data sources. Traversal costs are color-scaled for improved visibility where blue and red correspond to lowest and highest traversal cost areas respectively.

method is shown in Figure 9 as a function of training time. Our algorithm is shown to outperform the human-trained algorithm using only imagery data after only a short period of training. However, it should be pointed out that maintaining a tight correspondence from traversal costs assigned by the human-trained algorithm to those assigned by the perception system was difficult to strictly enforce. This highlights another advantage of the online learning approach over a human-trained approach: by relieving the need for manual manipulations of traversal cost values, the entire system is more adaptable to changes in both the environment and the perception system.

During post-analysis of this test, we discovered that the overhead imagery data and the estimated position of the vehicle were in fact misaligned by about 1.5 meters. While this result shows that our algorithm is robust to such map misalignment, this paper also demonstrates how our algorithm can be used to detect such errors in alignment in order to achieve optimal performance.

C. Offline Map Alignment

We applied our map alignment technique to a manually misaligned log of perception data and overhead imagery

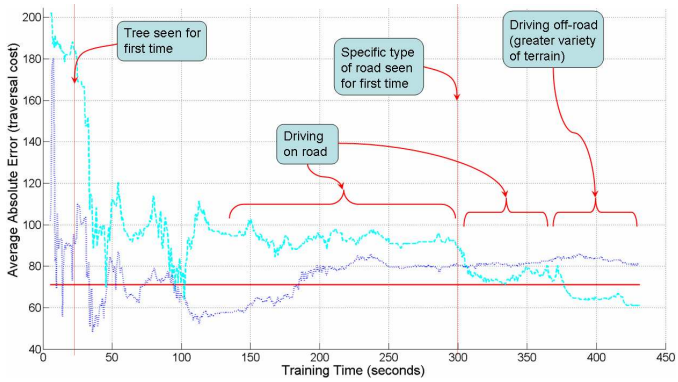
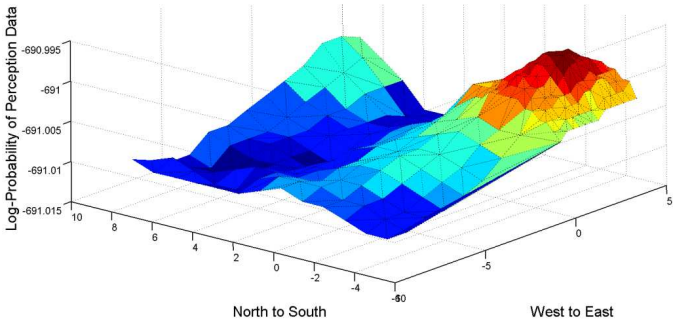


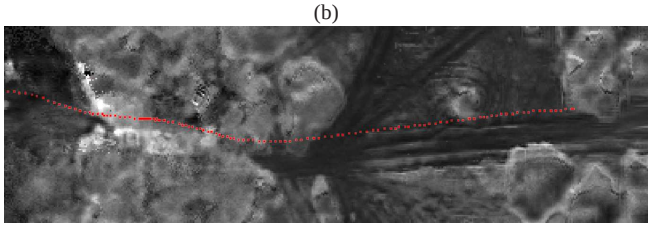
Fig. 9. Average absolute error between log scale traversal costs computed by unmanned ground vehicle's on-board perception system over the course of a multi-kilometer traversal of terrain and traversal cost estimates computed using three techniques: human-trained supervised learning algorithm using high resolution imagery and elevation data (solid red) and our algorithm using only color imagery (dashed cyan) and black and white imagery (dotted blue) as a function of training time by driving in a similar environment. See Table II for a description of data sources.

features. A brute force search across all potential map alignments in $0.35m$ increments in the four cardinal directions detected a misalignment of 3.85 m west and 4.9 m north. Computed probabilities of observed perception data and the corresponding improvement in traversal cost estimates can be seen in Figure 10. As expected, correcting the misalignment improved the definition of obstacles in the traversal cost maps and resulted in a stronger correspondence with the actual environment, correctly showing that the traveled path is clearly on the road.

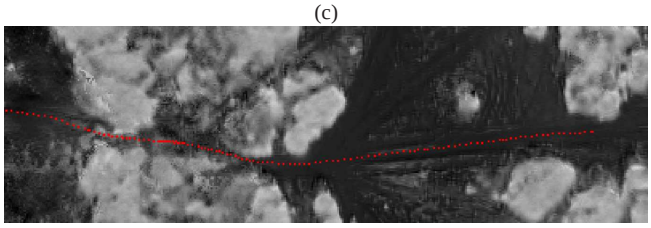


(a)

(b)



(c)



(d)

Fig. 10. Example of how misalignment between overhead data sources and estimated vehicle position can be detected using our algorithm. Computed log probability of the perception system sensor data encountered over a 12.6 m $\times$ 12.6 m search space of alignment shifts is shown in (a). Sample cost prediction for area shown in (b) before alignment correction and after correcting detected misalignment of 3.85 m west and 4.9 m north) appear in (c) and (d) respectively (best alignment is assumed to be that which produces the highest probability of seen perception data). Darker colors in the images correspond to lower traversal costs.

V. Conclusion

We have proposed a self-supervised online learning algorithm to learn and infer between different types of data sources that vary in density, reliability, and scope. By applying the scoped learning model, we were able to generalize from one type of data source to be able to work with another which may be difficult to generalize to new environments. As a result, we were able to extend the scope of such features to many possible domains.

We showed how the algorithm can be used to improve the navigation capabilities of unmanned ground vehicles by learning in real-time to interpret overhead imagery data to predict terrain traversal costs generated from an on-board perception system. We demonstrated this approach through actual field tests on-board a large robot in complex natural environments. Both online and offline results were given to demonstrate several applications of the algorithm. While performance could be hampered because of limitations in the available features or availability of representative training examples, the use of this algorithm was shown to improve the quality of robot navigation performance. Allowing robots to adapt to and improve their performance in diverse environments without human involvement through such techniques greatly expands effectiveness and potential applications of robotic systems.

Acknowledgment

This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) under contract "Unmanned Ground Combat Vehicle - PerceptOR Integration" (contract number MDA972-01-9-0005). 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.

We would also like to thank all the members of the UPI project team, without whom this work would not be possible.

References

[1] D. Lieb, A. Lookingbill, and S. Thrun, "Adaptive road following using self-supervised learning and reverse optical flow," in *Proceedings of Robotics: Science and Systems*, June 2005.

[2] A. Stentz, A. Kelly, P. Rander, H. Herman, O. Amidi, R. Mandelbaum, G. Salgian, and J. Pedersen, "Real-time, multi-perspective perception for unmanned ground vehicles," in *AUVSI*, 2003.

[3] D. Silver, B. Sofman, J. A. Bagnell, N. Vandapel, and A. Stentz, "Experimental analysis of overhead data processing to support long range navigation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[4] D. Blei, J. Bagnell, and A. McCallum, "Learning with scope, with application to information extraction and classification," in *Proceedings of the 2002 Conference on Uncertainty in Artificial Intelligence*, June 2002.

[5] M. I. Jordan and Y. Weiss, Eds., *Graphical models: Probabilistic inference*. MIT Press, 2002.

[6] S. M. Kakade and A. Y. Ng, "Online bounds for bayesian algorithms," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 641–648.

[7] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*. Chapman and Hall/CRC, 2004.

[8] M. E. Tipping, "Bayesian inference: An introduction to principles and practice in machine learning," in *Proceedings of the Advanced Lectures on Machine Learning*, 2003.

[9] T. Minka, "A family of algorithms for approximate bayesian inference," 2001.

[10] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the 1994 IEEE Conference on Robotics and Automation*, May 1994.