

Planning Most-Likely Paths From Overhead Imagery

Liz Murphy and Paul Newman

Abstract—This paper is about planning paths from overhead imagery, the novelty of which is taking explicit account of uncertainty in terrain classification and spatial variation in terrain cost. The image is first classified using a multi-class Gaussian Process Classifier which provides probabilities of class membership at each location in the image. The probability of class membership at a particular grid location is then combined with a terrain cost evaluated at that location using a spatial Gaussian process. The resulting cost function is, in turn, passed to a planner. This allows both the uncertainty in terrain classification and spatial variations in terrain costs to be incorporated into the planned path. Because the cost of traversing a grid cell is now a probability density rather than a single scalar value, we can produce not only the *most-likely* shortest path between points on the map, but also sample from the cost map to produce a distribution of paths between the points. Results are shown in the form of planned paths over aerial maps, these paths are shown to vary in response to local variations in terrain cost.

I. INTRODUCTION

In this paper we deal with the problem of path planning outdoors using cost maps for mobile robots. Operating in such an environment brings a number of challenges, chief amongst which is that the environment is only partially known. From limited data, cost maps must be generated which represent a continuum of terrain costs wherein the cost of a particular terrain may vary over the course of path execution. Path derivation generally requires the cost map to be passed to a planner such as A*, D* or E* which, unlike in other planning approaches such as PRMs or RRTs, will seek to minimize the accumulated cost from start to goal of some navigation function.

Existing approaches to long range path planning typically generate deterministic cost maps where each grid cell is attributed a scalar value proportional to the estimated cost of traversing that terrain. This paper addresses two issues with deterministic cost maps, (i) terrain classification is an uncertain process, and (ii) even within a particular terrain class, the cost of traversing that terrain may not be the same all over the map. Figure 1 illustrates a type of terrain that pose difficulties for an overhead classifier - it interweaves traversable open terrain with woodland, and in planning through this area it would be useful to take this confusion into account. Figure 2 shows two examples of what would be classified as dirt road but which would have widely varying terrain costs in different parts of our map. Here we use Gaussian Processes - a non-parametric Bayesian technique - to create probabilistic cost maps which explicitly take into

account uncertainty in terrain classification whilst permitting the cost of a particular terrain type to vary over the map.

Underpinning this framework is a cost function that produces a probability density function (pdf) instead of a scalar value for each grid cell (as is common practice). The cost function combines the probability of terrain class membership with the expected terrain cost at that cell (also a probability density function). The overall cost of the cell is the result of integrating over all possible classes of terrain, taking into account the uncertainty that location. Each cell is first classified using a Gaussian Process Classifier, which gives the probability of membership of each of the n_c classes of terrain represented in the map. Independently of this, the cost of terrain class T at the grid cell is evaluated using a Gaussian Process Regressor for each of the n_c classes. Then, for each cell, we calculate a pdf over cost. The resulting cost map can then be sampled to produce a distribution of likely paths between points A and B in the map, or the mean of the cells can be used to produce the most-likely path. Of course, the vehicle’s knowledge of the environment is enhanced as it moves, and Gaussian processes give us the ability to easily retrain the cost function and come up with a more accurate planning mechanism on-the-fly.

The key contribution of this paper is a novel framework for cost map generation. Results are shown which demonstrate the ability of this technique to plan sensible paths from overhead imagery using only a limited number of features and a sparse manually-labeled training data set. We demonstrate the ability to sample from the cost map to generate a distribution of paths in differing terrain, and also show how the framework allows the path to adapt in response to spatial variance in the cost of terrain - such as in the case where a roadway may be blocked due to heavy traffic or normally traversable scrub is prohibitively expensive due to a locally steep gradient.

II. RELATED WORK

The idea of explicitly accounting for uncertainty when planning paths is not new. Probabilistic Planners such as PRMs [1] and RRTs [2] can use uncertainty in the environment when sampling - instead of planning over a regular grid with a graph structure with a ‘node’ for each grid cell they only create nodes in places known to be free. However the jagged paths they create are generally unsuited to field robots and they are more readily used in planning over high dimensional spaces. More akin to the methods described in this paper [3] presents a randomized version of A* using machine learning to generate heuristics for the A* planner. Phillipsen [4] presents a probabilistic navigation



Fig. 1. Uncertainty in Classification: Forest or Open Terrain? Traversable or Not?



(a) High Traversal Cost



(b) Low Traversal Cost

Fig. 2. Spatial Variation in Terrain Cost: Dirt Roads

function for planning in unknown dynamic environments, Likhachev [5] introduces the notion of clear preferences on the missing information in the environment and combines this with multiple A*-like searches to produce an algorithm capable of dealing with uncertainty in large maps. Another method [6] is to frame uncertainty as a measure of the utility of the cell in exploring the environment and use this reward function to guide the robot.

The benefit of planning from overhead data gathered from aerial sources such as satellites and planes has been noted in [7] where aerial data is combined with laser data from a ground vehicle to classify terrain using a neural network. In [8] imitation learning is used to automate the building of a cost function from overhead data which may be used with different planners.

Gaussian processes themselves have found many successful applications in robotics. In [9] Gaussian process classification is used to exploit the structure of the environment in creating occupancy grids. Plagemann [10] uses learned terrain models and Gaussian process regression to predict elevations at unseen locations in planning paths for a legged robot. This builds on earlier work [11] which uses non-stationary covariance functions in the Gaussian process to accurately model terrain so that flat areas remain smooth while discontinuities caused by edges and corners can be preserved. Gaussian processes have long been used by Geostatistics community (where the technique is known as Kriging and usually restricted to modeling in physical spaces,

rather than the feature space representations commonly found in robotics applications). In [12] pixels within hyperspectral images are classified into land cover types using maximum-likelihood classification where the mean of each spectral band at a given location has been modeled by a spatially-varying Gaussian process.

III. GAUSSIAN PROCESSES

Gaussian process (GP) modelling is a non-parametric supervised learning technique which has recently found favour in robotics, principally due to its capability of providing uncertainty estimations at prediction points. A thorough treatment of Gaussian Processes is provided by [13]. A GP is a collection of random variables which forms a multivariate Gaussian distribution with mean $\mu(x)$ and covariance function $k(x, x')$. Given a training set $\mathcal{D} = \{(\mathbf{x}_i, y_i) | i = 1, \dots, N\}$ where \mathbf{x} denotes an input vector of dimension d and y represents a scalar output, we want to learn a model for the functional dependency $y_i = f(x_i) + \epsilon_i$. Under the GP paradigm, we view all target values y as jointly Gaussian distributed $p(y_1, \dots, y_n | x_1, \dots, x_n) \sim \mathcal{N}(\mu, K)$. Typically, we assume the mean μ is 0 and use the covariance matrix K to specify prior knowledge of $f(x)$ through a covariance function $\text{cov}(f(x_i), f(x_j)) = k_{ij} := k(x_i, x_j) + \sigma_n^2 \delta_{ij}$, where σ_n^2 is a global gaussian noise variance.

Hence we are able to write the joint distribution of the observed target values y and the function values f_* at the unknown target locations \mathbf{x}_* under this prior as:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix} \right) \quad (1)$$

By conditioning the prior on the observations y , we arrive at the following key equations for Gaussian Process Regression (GPR) for a single test point x_* .

$$\mathbf{f}_* | X, \mathbf{y}, \mathbf{x}_* \sim \mathcal{N}(\bar{\mathbf{f}}_*, \mathbb{V}[\mathbf{f}_*]) \quad (2)$$

where

$$\bar{\mathbf{f}}_* = \mathbf{k}_*^\top (K + \sigma_n^2 I)^{-1} \mathbf{y} \quad (3)$$

$$\mathbb{V}[\mathbf{f}_*] = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top (K + \sigma_n^2 I)^{-1} \mathbf{k}_* \quad (4)$$

for notational simplicity we have used $\mathbf{k}_* = k(x_*, x)$ to denote the covariance between the test point and the training points, and $K = K(X, X)$ to represent the matrix of covariances between the n training points.

When using Gaussian Processes the only ‘prior’ we place over the model of our dataset is the form of the covariance function. A commonly used covariance function is the *squared exponential*:

$$k(x_i, x_j) = \sigma_f^2 \exp \frac{-(x_i - x_j)^2}{2l^2} \quad (5)$$

which is stationary and hence invariant to translation in the input space while also leading to very smooth modelling of the data.

The quality of predictions produced by the Gaussian process regressor depends entirely on the suitable choice of covariance function. Typically a covariance function has

some free parameters (σ_f and l in (5)), which in the Gaussian process framework are referred to as *hyperparameters* (denoted by θ) to emphasize that they are parameters of a non-parametric model. If the choice of θ is inappropriate then poor results will be returned. To avoid this, and find the best model for our data, we train the Gaussian Process by computing the likelihood of the hyperparameters given the training data \mathbf{x}, \mathbf{y} . Under Bayes' theorem this can be done by maximizing the log-marginal likelihood $\log p(\mathbf{y}|\mathbf{x}, \theta)$ according to equation (6), or by integrating over all possible choices for θ .

$$\log p(\mathbf{y}|\mathbf{x}, \theta) = -\frac{1}{2}\mathbf{y}^T K^{-1}\mathbf{y} - \frac{1}{2}\log |K| - \frac{n}{2}\log 2\pi \quad (6)$$

Classification with Gaussian Processes is a natural progression from regression, but is slightly more complex as the values of the target y are no longer continuous. In the multi-class case our n training points $\{x_1, \dots, x_n\}$ are now related to a vector y of length $C \times n$:

$$\{y_{i1}, y_{i2}, \dots, y_{iC}, \dots, y_{n1}, y_{n2}, \dots, y_{nC}\}$$

with entries of 1 for the class which corresponds to the label of point \mathbf{x}_i and 0 for the remaining $C - 1$ entries for that point.

Again, we assume the existence of an underlying latent function $f(x)$

$$\{f_{i1}, f_{i2}, \dots, f_{iC}, \dots, f_{n1}, f_{n2}, \dots, f_{nC}\}$$

over which we place a GP prior with form $\mathbf{f}|\mathbf{X} = \mathcal{N}(\mathbf{0}, K)$. K is a block diagonal matrix as the C latent processes are uncorrelated: within each submatrix K_C we represent the correlations within that class C only.

Using Bayes' rule we can compute the posterior of the latent function:

$$p(\mathbf{f}|X, y) = \frac{p(\mathbf{f}|\mathbf{x})p(\mathbf{y}|\mathbf{f})}{p(X, y)} \stackrel{i.i.d}{=} \frac{\mathcal{N}(\mathbf{0}, K)}{p(X, y)} \prod_{i=1}^N p(\mathbf{y}_i|\mathbf{f}_i). \quad (7)$$

Unlike in regression, the likelihood term $p(\mathbf{y}|\mathbf{f})$ is not Gaussian, and in multi-class classification is often a *softmax* function:

$$p(y_i^c|\mathbf{f}_i) = \frac{\exp(f_i^c)}{\sum_{c'} \exp(f_i^{c'})}. \quad (8)$$

From (7) we can obtain the class prediction of a new test point \mathbf{x}_* in two steps.

$$p(\mathbf{f}_*|X, y, \mathbf{x}_*) = \int p(\mathbf{f}_*|\mathbf{f}, X, \mathbf{x}_*) \underbrace{p(\mathbf{f}|X, y)}_{(7)} d\mathbf{f} \quad (9)$$

$$p(\mathbf{y}_*|X, y, \mathbf{x}_*) = \int p(\mathbf{y}_*|\mathbf{f}_*) \underbrace{p(\mathbf{f}_*|X, y, \mathbf{x}_*)}_{(9)} d\mathbf{f}_* \quad (10)$$

A number of techniques exist for performing multi class Gaussian process classification. They vary in their choice of likelihood function (eg softmax, multinomial probit) and in their approach to handling the analytically intractable integrals of equations (9) and (10).

Naive techniques for classification and regression take $O(N)^3$ to compute during training, and $O(N^2)$ during prediction, due to the inversion of the covariance matrix K . N is the number of training data points. In this paper we have made use of toolkits which employ *sparse* approaches.

Typically, these sparse methods seek to reduce the number of training points by selecting $M \ll N$ points which are still *informative* enough (as measured by entropy) to adequately represent the training set. This reduces training cost to $O(NM^2)$ and prediction to $O(M^2)$.

IV. PROBABILISTIC PLANNING FRAMEWORK

Gaussian processes can be introduced into the planning domain to provide a means of dealing with uncertainty in the construction of cost maps for grid map planners. In our framework, we use data obtained from overhead imagery to plan long range paths over differing terrain. Due to the flexibility of the Gaussian process paradigm, the framework can accommodate differing inputs (color, hyperspectral data, principal direction) which aid in terrain classification and terrain cost.

Imagine we are given an aerial image of the area over which we wish to plan. We are going to use the aerial data to classify each grid cell as one of n terrain classes, so in order to do this we require some labelled training data which relates features of the image (eg x,y location; r,g,b color; hyperspectral data if available) to a single class label.

Independent of this we are given a priori cost data for the region for *each* of the n terrain classes. The reason for approaching the problem this way is that it frees the planner from having to know the class of a cell or area before placing a prior on traversability. We are hence able to incorporate vague information only relevant to a particular class(es) into our cost map. Examples of this would include '*the area to the south west is swamp land*', which would affect the cost of anything classed as open terrain, but not the cost of roads or impassable obstacles such as houses. Likewise '*all roads around London are congested*' affects only the cost of roads, not open terrain or buildings. Attributing a separate spatial GP to each class allows us to model the entire map from a much smaller representative set of training data than would be required if we were attempting to model all classes together using just one GP. As training data we assume some prior knowledge of the blanket cost of a particular terrain type together with some local spatial variations.

We begin by discretizing the environment into a grid, this can be done at the pixel or superpixel level. We classify the grid cell using the aerial imagery to obtain $p(T|x)$, the probability of terrain class membership at that location. Here we use T to represent the terrain class, and x is the location (x, y) in the map. Separately, we evaluate our spatial Gaussian process regressor for each of the n classes to obtain $p(c|T, x)$, the pdf representing the traversal cost c of *that particular class* at the given location. Because our spatial GPR can be viewed as modelling a cost-contour map for each class we assume that a smoothing between data points is appropriate and hence used a squared exponential covariance function for the regression.

Next, the cost function combines the terrain class membership and terrain cost at a point x in a probabilistic form, integrating out over the class types to produce the overall cost of the cell.

$$p(c|x) = \int_T \underbrace{p(c|T, x)}_{GPR} \underbrace{p(T|x)}_{GPC} dT \quad (11)$$

$$= \sum_{t \in T} p(c|t, x) p(t|x) \quad (12)$$

Finally, we pass the information in the cost function to a planner. Because our cost function is a pdf over grid cell locations $\{x, y\}$ we are able to not only produce the most likely shortest path between point A and point B, but by repeatedly sampling the cost function we may also produce a likely distribution of paths between the two points.

V. RESULTS

To perform the multi-class classification we used the Variational Bayes Gaussian Process (VBGP) MATLAB toolkit [14]. It uses sparse approximations and the variational bayes methodology together with a multinomial probit likelihood to estimate the posterior over class labels. Using this technique, classification scales linearly in the number of classes but is of order $O(NM^2)$ in training and $O(M^2)$ in prediction.

As we are potentially dealing with a large number of data points in the regression case, we also made use of a sparse gaussian process toolbox to perform the spatial regression. Namely the IVM toolkit [15]. As we assume a spatially smooth transition of values across the one terrain class, we use a stationary squared exponential covariance function.

In this work we used A* to generate results but the technique is equally applicable to any other planner reliant on a cost map.

A. Responding to Spatial Variations in Terrain Cost: Most Likely Paths

Figure 3 shows our framework operating on a simulated data set. The aerial image has a resolution of 15m per pixel, and each pixel becomes a grid cell for our planner. The image has been labelled with 3 classes, and trained using 50 points each for the *road*, *obstacle* and *scrub land* classes with input features corresponding to red, green and blue channels in the image. In figures 3(a) - 3(c) we see the results of the classification stage. Note how we get reasonably strong probabilities (approaching 0.6) in the Obstacle and Road classes, yet a muted response for the Open Terrain class. Looking at the original image we see greatest variation in colour amongst the open terrain class, there are various shades of green (NB These results are best understood in colour) and the light and dark extent of the colour range represented leaves pixels of this class more open to misclassification than road or obstacle pixels.

Figures 3(d) - 3(f) show the spatial cost maps for each terrain type masked by the terrain classification. The cost maps for the Obstacle and Scrub classes are almost uniform over the map, but we have added some local high cost areas to the road class to simulate areas of congestion, visible in figure 3(f). The blue lines in figure 3(k) show two different paths planned using this particular cost map.

Figures 3(g) - 3(i) show the convolution of spatial cost map and class probability for a different congestion situation.

The difference is most visible in 3(i) where we see two very high cost regions in the road cost map covering territory which the blue path of the previous instance passed through. Running the planner over this second cost map configuration produces the magenta paths shown in 3(k), so we see that by introducing some local spatial variation in the road class only we have been able to influence the path chosen by the planner.

B. Generating a Distribution of Paths

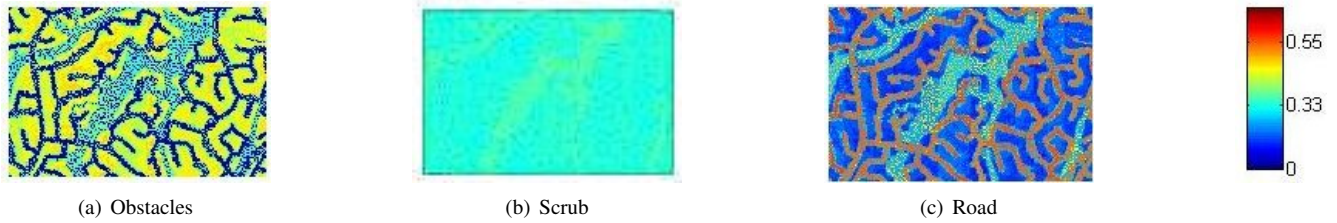
As mentioned, the pdf cost function can be sampled to generate a distribution of paths between points. Figures 4(a) to 4(c) illustrate the framework operating over 3 different aerial images taken from Google Maps. The first two show different urban environments, which we classified into *road*, *scrub* and *obstacle*. The third shows a portion of the Australian desert, which we classified into *dirt road*, *tree canopy* and *open terrain*. Again, 50 training data points were used for each class in classification. This time the cost maps were approximately uniform and scaled appropriately so that *road* was preferred over *scrub* which was in turn preferable to *obstacle*.

In figures 4(a) to 4(c) we see the results of running 50 different A* searches over the map, resampling the cost function each time. All paths have been plotted using an alpha-blended overlay. In each case we see a thickened *most-likely* path, together with other *feasible* paths that could result given that many of the grid cells on the most-likely path have uncertain classifications.

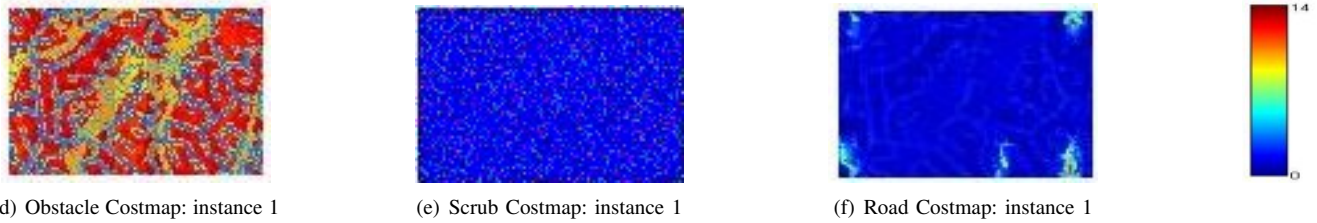
Figure 5 shows a histogram of the distribution of path costs that stems from running the same search over the desert image 200 times. Note that the distribution is multi-modal and that this is also reflected in 4(c) where a number of distinct clusters of similar paths are visible.

VI. CONCLUSIONS AND FUTURE WORK

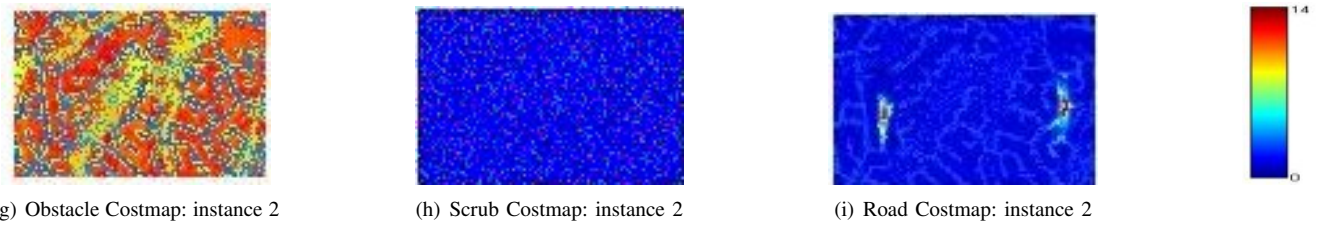
This paper presented an approach to probabilistic cost map construction and path planning. Using two separate Gaussian process techniques we were able to classify locations in an aerial image probabilistically into one of n_C classes. Independently of classification we produce a cost map for each of the n_C terrain classes using a sparse Gaussian process regressor. The benefits of this approach are multifarious: we allow the cost of a terrain type to vary spatially, this reduces the classification burden as we can abstract away variations within broader terrain types into the independent spatial cost map and hence require fewer classes to represent terrain. We are also able to incorporate vague local knowledge (like ‘avoid highways’) into the spatial cost maps without having to know the exact location of every instance of that terrain type in the affected region. The Gaussian process spatial regressor produces a cost and variance at each location in the map for each terrain type, this is combined with the probability of class membership at that point in a global probabilistic cost map. We can sample from the cost map to produce a distribution of probable paths between two points,



The results of classifying the image into 3 classes: Obstacle, Scrub and Road. Note the relatively good discrimination between high and low confidence regions obtained for both the Obstacle and Road classes, and compare this with the Scrub class, where confidence in actual regions of scrub (evident in (j)) barely reaches 0.4. The scrub class has the highest variation in colour (shown in (j)) and as we train and classify on colour information only, this leads to poorest results for this class.



(d)-(f) The cost maps for each of the 3 terrain types used for the first iteration of the planner. To enhance understanding we have masked the cost maps with the probability of class membership. Note that the Obstacle class is generally high cost everywhere, while the road class is almost uniformly low cost except for localised high cost regions which we have deliberately added around the edge of the image.



(g)-(i) The cost maps for each of the 3 terrain types used for the second iteration of the planner. Note the high cost regions added to the road costmap right where the path of instance 1 passed through.

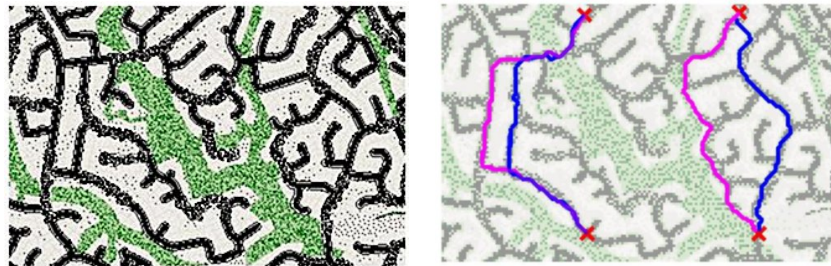


Figure (k) shows paths planned using costmap instances 1 (blue) and 2 (magenta). Note that the magenta path resulting from instance 2 now veers around the high cost regions added to the map, and that by comparing with the original image (j) the magenta path in the right hand side of the image actually prefers to traverse scrub rather than road for part of the route.

Fig. 3. Altering the path in response to spatial variation in terrain cost

as well as use the mean value of the cost function to find the most-likely path.

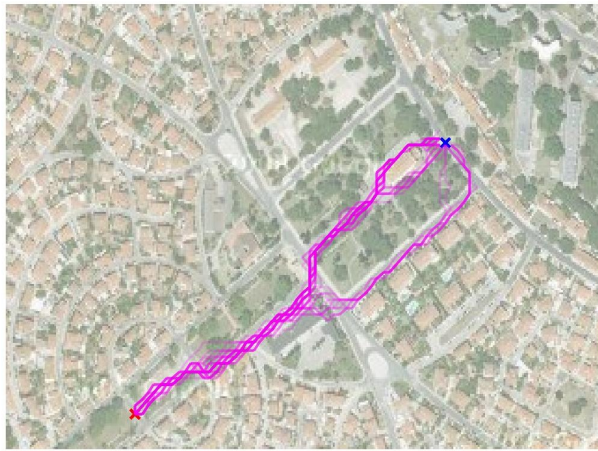
Although we chose to demonstrate the framework by planning over aerial images, its use could be applied to other domains (such as on a mobile robot) simply by altering the input feature set used for classification. We obtained decent results using only colour red, green and blue pixel intensities for classification into three classes. However, we would expect better results using more inputs, such as hyperspectral

data which would allow us to better discriminate between terrain such as vegetation or water. In turn, better paths could also be obtained by using more classes, this will not introduce much extra computational burden as the Gaussian process multi-class classification technique we employed scales linearly with the number of classes.

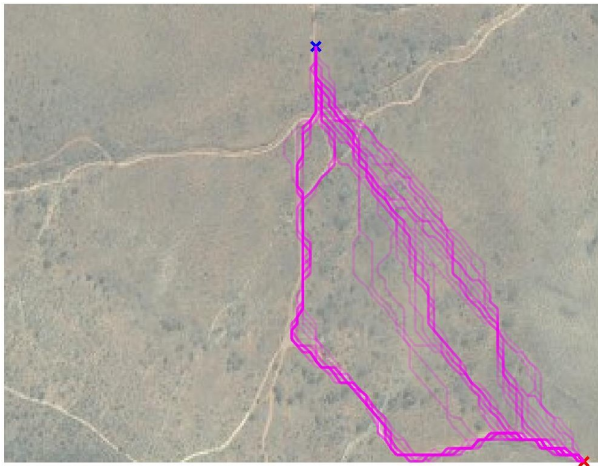
In a similar vein, we have used standard squared-exponential covariance functions in both the GPC and the GPR. Future work would encompass experimenting with dif-



(a) Planning Paths Across Tempe, Az



(b) Planning Paths Across Lisbon, Portugal



(c) Planning Paths Across the desert, Australia

Fig. 4. Path Distributions

ferent covariance function. One would expect the use of non-stationary covariance functions in modelling the spatially varying cost map to produce better results, as these may better capture the abrupt local variations in terrain cost that we are interested in modelling.

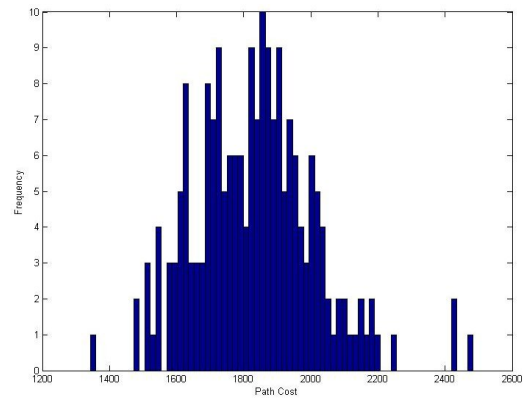


Fig. 5. The multi-modal distribution of path costs resulting from 200 iterations of planning over the desert image of Figure 4(c).

REFERENCES

- [1] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [2] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Computer Science Dept., Iowa State University, Tech. Rep. TR 98-11, October 1998.
- [3] R. Diankov and J. Kuffner, "Randomized statistical path planning," in *Proceedings of IEEE/RSJ 2007 International Conference on Robots and Systems (IROS07)*, October 2007.
- [4] R. Philippsen, S. Kolski, K. Macek, and B. Jensen, "Mobile robot planning in dynamic environments and on growable costmaps," in *Workshop on Path Planning on Costmaps at the IEEE International Conference on Robotics and Automation*, Pasadena, California, USA, May 19-23 2008.
- [5] M. Likhachev and A. Stentz, "Probabilistic planning with clear preferences on missing information," *Artificial Intelligence*, vol. 173, no. 5-6, pp. 696 – 721, 2009, advances in Automated Plan Generation.
- [6] J. K. Rosenblatt, "Optimal selection of uncertain actions by maximizing expected utility," *Auton. Robots*, vol. 9, no. 1, pp. 17–25, 2000.
- [7] B. Sofman, J. A. D. Bagnell, A. T. Stentz, and N. Vandapel, "Terrain classification from aerial data to support ground vehicle navigation," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-39, January 2006.
- [8] D. Silver, J. A. D. Bagnell, and A. T. Stentz, "High performance outdoor navigation from overhead data using imitation learning," in *Robotics Science and Systems*, June 2008.
- [9] S. OCallaghan, F. T. Ramos, and H. Durrant-Whyte, "Contextual occupancy maps using gaussian processes," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009.
- [10] C. Plagemann, S. Mischke, S. Prentice, K. Kersting, N. Roy, and W. Burgard, "Learning predictive terrain models for legged robot locomotion," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, 2008.
- [11] T. Lang, C. Plagemann, and W. Burgard, "Adaptive non-stationary kernel regression for terrain modeling," in *Robotics: Science and Systems (RSS)*, Atlanta, Georgia, USA, June 2007.
- [12] G. Jun and J. Ghosh, "Spatially adaptive classification of hyperspectral data with gaussian processes," in *IEEE International Geoscience and Remote Sensing Symposium*, 2009.
- [13] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [14] M. Girolami and S. Rogers, "Variational bayesian multinomial probit regression with gaussian process priors," *Neural Computation*, vol. 18, no. 8, pp. 1790–1817, 2006. [Online]. Available: <http://www.dcs.gla.ac.uk/people/personal/girolami/pubs.2005/VBGP/>
- [15] N. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse gaussian process methods: The informative vector machine," *Neural Information Processing Systems*, vol. 15, pp. 609–616, 2003. [Online]. Available: <http://www.cs.manchester.ac.uk/neill/ivm/>