Outdoor Path Labeling Using Polynomial Mahalanobis Distance

Greg Grudic
Department of Computer Science
University of Colorado
Boulder, CO
Email: grudic@cs.colorado.edu

Jane Mulligan
Department of Computer Science
University of Colorado
Boulder, CO
Email: Jane.Mulligan@colorado.edu

***Abstract*— Autonomous robot navigation in outdoor environments remains a challenging and unsolved problem. A key issue is our ability to identify safe or navigable paths far enough ahead of the robot to allow smooth trajectories at acceptable speeds. Colour or texture-based labeling of safe path regions in image sequences is one way to achieve this far field prediction. A challenge for classifiers identifying path and nonpath regions is to make meaningful comparisons of feature vectors at pixels or over a window. Most simple distance metrics cannot use all the information available and therefore the resulting labeling does not tightly capture the visible path. We introduce a new Polynomial Mahalanobis Distance and demonstrate its ability to capture the properties of an initial positive path sample and produce accurate path segmentation with few outliers. Experiments show the method's effectiveness for path segmentation in natural scenes using both colour and texture feature vectors. The new metric is compared with classifications based on Euclidean and standard Mahalanobis distance and produces superior results.**

I. Introduction

Autonomous robot navigation in outdoor environments remains a challenging and unsolved problem. A key issue is our ability to identify *safe* or *navigable* terrain in the near and far field for the robot to traverse. For vision-based navigation Stereo reconstruction can give us information about ground plane features in the near field, but to plan and execute smooth trajectories at reasonable speed we need to identify acceptable paths in the far field (50m+) as well. A common approach is to identify and cluster path regions in images based on the appearance of regions known to be traversable in the near field. Once a frame is classified we can project safe regions onto the ground plane and calculate trajectory and steering angle.

Using color to segment road or path regions in images for autonomous navigation has been deemed important since the '80s [1], [2], [3], [4]. Frequently colour segmentation is augmented by edge detection, or template matching for the expected shape of the road. Rasmussen [5] combines laser range data with colour and texture to address the kind of outdoor paths we are interested in. Lieb et al. [6] match a sample template from the road ahead to identify tracks in the dessert.

A key problem with any image segmentation scheme is how to compute similarity or *distance* between two regions or pixels in an image. The most common metrics are Euclidean (L_2) and absolute (L_1) distances [7]. In a feature spaces such as RGB where such differences do not reflect perceived differences between colour vectors, we need a measure which can describe more sophisticated and nonlinear relationships between feature vectors. Mahalanobis distance is another common metric which attempts to capture the non-isotropic properties of an d dimensional feature space. It weights the distance calculation according to the statistical variation of each component using the covariance matrix. Recently Generalized Principal Components Analysis has been introduced [8] as a method for automatically identifying linear subspaces in d dimensional datasets by solving for the normal vector b_i for each subspace. To date this approach has not been exploited as a distance metric in classifying points or regions.

We propose a new metric we call the Polynomial Mahalanobis Distance which attempts to capture more of the information available in an initial image sample of the path appearance. The motivation behind this new metric is based on the observation that, in high dimensional space, Euclidean Distance metrics mark the sphere of points at distance d_E from a reference point as being equally similar to it, while they may be far from observed neighbours in the sample. This is problematic because two completely unrelated regions of the state space can be equally "close" to a reference point. Figure 1b shows a 2 dimensional version of this property of the Euclidean distance metric. The non reference points in 1a have no effect on the resulting distance metric, and distance (shown as a grey scale image, where white is close and dark is far) from the reference point (shown as the point inside the blue rectangle). One can improve the distance metric by using the Mahalanobis distance, which takes into account the data structure by computing the covariance matrix for the observed sample (learning data). However, as we see in Figure 1c (and Figure 2b) this metric still does not tightly follow the data. However, when the Polynomial Mahalanobis distance is used, as shown in Figures 1d, 1e, 2c, 2d, and 2e, the learning data used to compute the coefficients for the polynomial distance metric are labeled as closer than other regions of the state space. In fact, the higher the polynomial order, the tighter the distance metric model.

The Polynomial Mahalanobis distance is obtained by mapping the feature space into higher order polynomial terms– in our experiments we often obtain best results (i.e. most

discriminative distance metrics) by mapping 300 dimensional feature spaces into 128 order polynomial terms. As we show in Section II-B, such mappings are not computationally feasible if done by simply taking the features and exhaustively finding all possible polynomial terms. In this paper, we present a framework for effectively and efficiently creating very high order polynomial mappings from a very high dimensional feature space. This is done by successively projecting high dimensional data into low dimensional column spaces, then constructing second order polynomial terms in this space. The column spaces are low dimensional because they are constructed using local neighborhoods containing a small number (i.e. a few hundred) of points. At the first step we construct second order polynomial spaces, these in turn have a low dimensional column space projection for which can compute a higher order polynomial space. By repeating these steps we can efficiently construct 128th order or higher polynomial spaces from data with thousands of features.

In Section II we present the mathematical basics for our approach. We define the standard form of Mahalanobis distance in Section II-A as well as the details of the covariance calculations required. In Section II-B we present our new Polynomial Mahalanobis Distance and justify an efficient implementation technique. Experimental path labeling results and comparison to standard metrics are presented in Section III. The conclusion is given in Section IV.

II. Mathematical Formulation

A. The Mahalanobis Distance

Let $\mathbf{z}_i$ and $\mathbf{z}_j$ be d dimensional column vectors (i.e. $\mathbf{z}_i \in \Re^d$ and $\mathbf{z}_j \in \Re^d$ for all i, j). Then the Mahalanobis distance between $\mathbf{z}_i$ and $\mathbf{z}_j$ is defined by:

$$d_M(\mathbf{z}_i, \mathbf{z}_j) = \left((\mathbf{z}_i - \mathbf{z}_j)^\top C^{-1} (\mathbf{z}_i - \mathbf{z}_j) \right)^{\frac{1}{2}} \tag{1}$$

where C is a covariance matrix. Note that if C is a $d \times d$ identity matrix, the the Mahalanobis distance reduces to the Euclidean distance $||\mathbf{z}_i - \mathbf{z}_j||$. Given a set of N observations $\{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N\}$, we can estimate the covariance matrix as follows:

$$C = \frac{1}{N-1} \mathbf{A}^\top \mathbf{A} \tag{2}$$

where $\mathbf{A}$ is a N by d matrix defined by:

$$\mathbf{A} = \begin{pmatrix} \mathbf{z}_1^\top - \mathbf{z}_0^\top \\ \mathbf{z}_2^\top - \mathbf{z}_0^\top \\ \vdots \\ \mathbf{z}_N^\top - \mathbf{z}_0^\top \end{pmatrix} \tag{3}$$

where $\mathbf{z}_0$ is either the mean of the vectors $\{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N\}$, or a reference vector about which distances are being measured (in this paper $\mathbf{z}_0$ is a training input, and $\{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N\}$ are its N nearest neighbors).

From Equation 1, one can see that calculating the Mahalanobis distance requires the inversion of the covariance matrix C. Since this inverse often does not exist (especially for $d \ll N$ as is often the case for the problems studied in this paper), the covariance estimate must in practice be regularized. This regularization can take on many forms - in this paper it is done as follows. First, we perform a Singular Value Decomposition (SVD) of $\mathbf{A}^\top \mathbf{A}$ [9] giving:

$$\mathbf{A}^\top \mathbf{A} = \mathbf{U}\mathbf{W}\mathbf{V}^\top = \mathbf{U}\mathbf{W}\mathbf{U}^\top \tag{4}$$

(note that $\mathbf{U} = \mathbf{V}$ because $\mathbf{A}^\top \mathbf{A}$ is symmetric) where $\mathbf{W}$ contains the singular values $w_1, w_2, ..., w_d$ sorted in decreasing order:

$$W = \begin{pmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_d \end{pmatrix}$$

When there exist $w_i = 0$, the inverse of $\mathbf{A}^\top \mathbf{A}$, and hence the covariance matrix C, will not exist. We make C invertible by regularizing W by adding a small positive value value σ^2 to the diagonal:

$$W_{\sigma^2} = \begin{pmatrix} w_1 + \sigma^2 & & 0 \\ & \ddots & \\ 0 & & w_d + \sigma^2 \end{pmatrix}$$

Therefore our approximation of the covariance matrix takes on the form:

$$C_{\sigma^2} = \frac{1}{N-1} \mathbf{U}\mathbf{W}_{\sigma^2}\mathbf{U}^\top$$

which has the inverse:

$$C_{\sigma^2}^{-1} = (N-1)\, \mathbf{U}^\top \mathbf{W}_{\sigma^2}^{-1} \mathbf{U}$$

where

$$W_{\sigma^2}^{-1} = \begin{pmatrix} (w_1 + \sigma^2)^{-1} & & 0 \\ & \ddots & \\ 0 & & (w_d + \sigma^2)^{-1} \end{pmatrix}$$

The estimated Mahalanobis distance between $\mathbf{z}_1$ and $\mathbf{z}_2$ is thus given by:

$$d_{M_{\sigma^2}}(\mathbf{z}_i, \mathbf{z}_j) = \left((N-1) (\mathbf{z}_i - \mathbf{z}_j)^\top \mathbf{U}^\top \mathbf{W}_{\sigma^2}^{-1} \mathbf{U} (\mathbf{z}_i - \mathbf{z}_j) \right)^{\frac{1}{2}} \tag{5}$$

The above equation is problematic from a computational standpoint when d is large (which is the case in this paper), because $\mathbf{U}$ is a d by d matrix. Therefore, we use the following equivalent equation instead:

$$d_{M_{\sigma^2}}(\mathbf{z}_i, \mathbf{z}_j) = \left((N-1) \left(\frac{1}{\sigma^2} \|\mathbf{z}_i - \mathbf{z}_j\|^2 + \sum_{k=1}^{K} \left(\frac{-w_k}{\sigma^2 (w_k + \sigma^2)} \left\| U_k^\top (\mathbf{z}_i - \mathbf{z}_j) \right\|^2 \right) \right) \right)^{\frac{1}{2}} \tag{6}$$

where K is the number of columns of $\mathbf{U}$ associated with non-zero singular values $w_1, ..., w_K$, and U_k is the kth column of $\mathbf{U}$. The advantage of Equation 6 over Equation 5 is that it is both computationally faster and has lower memory requirements - only the first K columns of $\mathbf{U}$ are used, which can be a significant advantage if $K \ll d$. One can show

that Equation 6 and Equation 5 are equivalent by deriving Equation 5 from Equation 6 as follows:

$$d_{M_{\sigma^2}}(\mathbf{z}_i,\mathbf{z}_j) = \Bigg((N-1)\Bigg(\frac{1}{\sigma^2}\|\mathbf{z}_i-\mathbf{z}_j\|^2 + \sum_{k=1}^{K}\left(\frac{-w_k}{\sigma^2(w_k+\sigma^2)}\left\|U_k^\top(\mathbf{z}_i-\mathbf{z}_j)\right\|^2\right)\Bigg)\Bigg)^{\frac{1}{2}}$$
$$= \Big((N-1)\Big((\mathbf{z}_i-\mathbf{z}_j)^\top\mathbf{U}^\top\mathbf{Q}_1\mathbf{U}(\mathbf{z}_i-\mathbf{z}_j) + (\mathbf{z}_i-\mathbf{z}_j)^\top\mathbf{U}^\top\mathbf{Q}_2\mathbf{U}(\mathbf{z}_i-\mathbf{z}_j)\Big)^{\frac{1}{2}}$$
$$= \Big((N-1)(\mathbf{z}_i-\mathbf{z}_j)^\top\mathbf{U}^\top(\mathbf{Q}_1+\mathbf{Q}_2)\mathbf{U}(\mathbf{z}_i-\mathbf{z}_j)\Big)^{\frac{1}{2}}$$

where

$$\mathbf{Q}_1 = \begin{pmatrix} (\sigma^2)^{-1} & & 0 \\ & \ddots & \\ 0 & & (\sigma^2)^{-1} \end{pmatrix}$$

and

$$\mathbf{Q}_2 = \begin{pmatrix} \frac{-w_1}{\sigma^2(w_1+\sigma^2)} & & 0 \\ & \ddots & \\ 0 & & \frac{-w_d}{\sigma^2(w_d+\sigma^2)} \end{pmatrix}$$

Finishing the derivation, we get

$$\mathbf{Q}_1+\mathbf{Q}_2 = \begin{pmatrix} (w_1+\sigma^2)^{-1} & & 0 \\ & \ddots & \\ 0 & & (w_d+\sigma^2)^{-1} \end{pmatrix} = W_{\sigma^2}^{-1}$$

1) Implementation Details: This implementation of the Mahalanobis distance requires two user defined parameters: the number of points in the neighborhood N and the minimum singular value ratio s_{min}, which is used to set a singular value w_i to zero when $w_i < w_1 s_{min}$. The chosen s_{min} value is also used to set σ^2:

$$\sigma^2 = w_1 s_{min} \tag{7}$$

When $d > N$, an efficient way of calculating U is to use $\mathbf{A}\mathbf{A}^\top$ which is an $N \times N$ matrix (in contrast to $\mathbf{A}^\top\mathbf{A}$ which is $d \times d$). Singular Value Decomposition is used to calculate $\mathbf{A}\mathbf{A}^\top = \mathbf{U}_N\mathbf{W}\mathbf{U}_N^\top$. Finally, $\mathbf{U}$ is calculated by normalizing to length 1 the first K columns of $\mathbf{B} = (\mathbf{U}_N^T\mathbf{A})^T$ (which are associated with the K nonzero singular values $w_1, ..., w_K$ in $\mathbf{W}$).

B. The Polynomial Mahalanobis Distance

In this section we first define the q-Order Polynomial Mahalanobis Distance, and then present an efficient implementation for calculating it.

Definition: *Let $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_N\}$ be a set of m dimensional column vectors (i.e. $\mathbf{x}_i \in \Re^m$). Then the q-Order Polynomial Mahalanobis Distance between any two points $\mathbf{x}_i$ and $\mathbf{x}_j$ is calculated by first mapping all points $\mathbf{x}_k = (x_{k1}, x_{k2}, ..., x_{km})$, for $k = 1, ..., N$, into all polynomial terms of order q or less. We designate these polynomial term mappings $\mathbf{z}_k$. The dimension d of vectors $\mathbf{x}_k$ is:*

$$d = \frac{(m+q)!}{m! \cdot q!} \tag{8}$$

Equation (6) then gives the q-Order Polynomial Mahalanobis Distance between any $\mathbf{z}_i$ and $\mathbf{z}_j$ (corresponding to the points $\mathbf{x}_i$ and $\mathbf{x}_j$ mapped into their polynomial terms).

As an example of mapping into polynomial terms, consider a two dimensional vector $\mathbf{x}_k = (x_{k1}, x_{k2})$. The mapping of this $m = 2$ dimensional vector into its order $q = 2$ polynomial terms gives the following $d = 5$ dimensional vector:

$$\mathbf{z}_k = \left(x_{k1}, x_{k2}, x_{k1}^2, x_{k2}^2, x_{k1}x_{k2}\right)$$

A *naive* way of obtaining the q-order polynomial Mahalanobis distance between $\mathbf{x_i}$ and $\mathbf{x_j}$ is to directly map all data points into their respective d polynomial terms. Equation 6 can then be used to directly calculate the distance. For large m and degree q, the dimension d of resulting mappings quickly makes this approach computationally infeasible. For example, a relatively *small* problem (considering the experimental results presented in this paper - see Section III-B) could have $m = 50$ and $q = 8$, which, using Equation (8), would require more than 10^9 polynomial terms. Clearly, for large m and q, this implementation is not practical. In this section, we propose a novel framework for calculating the Polynomial Mahalanobis distance which *is* computationally practical for large m and q.

We propose the following framework for calculating q-order Polynomial Mahalanobis (PM) distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. We assume a set of N points $\mathbf{x}_1, ..., \mathbf{x}_N$ and a reference point $\mathbf{x}_0$ used for covariance calculations, and denote the Polynomial Mahalanobis (PM) as $d_{PM}(\mathbf{x}_i, \mathbf{x}_j)$ which is given by:

$$d_{PM}(\mathbf{x}_i,\mathbf{x}_j) = d_{M_{\sigma^2}}(\mathbf{x}_i,\mathbf{x}_j) + \sum_{l=1}^{L} d_{M_{\sigma^2}}\left(\mathbf{g}_l^i, \mathbf{g}_l^j\right) \tag{9}$$

where $L > 1$ is a user defined parameter giving a $q = 2^L$ order polynomial Mahalanobis distance, $d_{M_{\sigma^2}}$ is the Mahalanobis distance as defined in Equation 6, and $\mathbf{g}_l^i$ and $\mathbf{g}_l^j$ are defined as follows. Let ${}^xU_1, ..., {}^xU_K$ be the K column space vectors calculated using SVD on the $N+1$ points $\mathbf{x}_0, ..., \mathbf{x}_N$ as defined by the covariance calculations in Section II-A (i.e. these U column space vectors are associated with the nonzero singular values $w_1, ..., w_K$). Using these column vectors, we can project the points $\mathbf{x}_i$ and $\mathbf{x}_j$ into this K dimensional column space, obtaining the vectors $\mathbf{f}_l^i$ and $\mathbf{f}_l^j$:

$$\mathbf{f}_1^i = \begin{pmatrix} {}^xU_1^\top \\ {}^xU_2^\top \\ \vdots \\ {}^xU_K^\top \end{pmatrix}\mathbf{x}_i \quad \mathbf{f}_1^j = \begin{pmatrix} {}^xU_1^\top \\ {}^xU_2^\top \\ \vdots \\ {}^xU_K^\top \end{pmatrix}\mathbf{x}_j$$

These K dimensional vectors, which for notational convenience we symbolize by $\mathbf{f}_l^i = (a_1, ..., a_K)$ and $\mathbf{f}_l^j = (b_1, ..., b_K)$, are projected into *second order polynomial space*,

giving, from Equation (8), the $d_1 = ((K+2)\cdot(K+1)/2)$ dimensional vectors $\mathbf{g}_1^i$ and $\mathbf{g}_1^j$:

$$\mathbf{g}_1^i = \left(a_1, ..., a_K, a_1^2, ..., a_K^2, a_1a_2, a_1a_3, ...\right)$$
$$\mathbf{g}_1^j = \left(b_1, ..., b_K, b_1^2, ..., b_K^2, b_1b_2, b_1b_3, ...\right)$$

Similarly, the neighborhood points $\mathbf{x}_k$, for $k = 1, ..N$, can be projected into this $\mathbf{g}_1^k$ polynomial space. SVD is now performed on these neighborhood points (as in Equation (4) where the data matrix $\mathbf{A}$ now contains the points $\mathbf{g}_1^1, ..., \mathbf{g}_1^N$). This SVD calculation gives us the column space vectors ${}^{g_1}U_1, ..., {}^{g_1}U_K$, allowing the distance $d_{M_{\sigma^2}}\left(\mathbf{g}_1^i, \mathbf{g}_1^j\right)$ to be calculated.

Continuing with this constructive algorithm, we calculate $d_{M_{\sigma^2}}\left(\mathbf{g}_l^i, \mathbf{g}_l^j\right)$ for $l = 2, 3, ..., L$ as follows. All points from the previous polynomial level $\mathbf{g}_{l-1}^{\cdot}$ (the "$\cdot$" simply refers to all points) are projected into level l giving $\mathbf{g}_l^{\cdot}$, by first defining the intermediate vectors $\mathbf{f}_l^{\cdot}$:

$$\mathbf{f}_l^{\cdot} = \begin{pmatrix} {}^{g_{l-1}}U_1^t \\ {}^{g_{l-1}}U_2^t \\ \vdots \\ {}^{g_{l-1}}U_K^t \end{pmatrix} \mathbf{g}_{l-1}^{\cdot}$$

where ${}^{g_{l-1}}U_1, ..., {}^{g_{l-1}}U_K$ are the K column space vectors calculated using SVD on the $N+1$ points $\mathbf{x}_0, ..., \mathbf{x}_N$ projected into g_{l-1} space (i.e. using by the covariance calculations defined in Section II-A). These K dimensional vectors are projected into *second order polynomial space*, giving the $d = ((K+2)\cdot(K+1)/2)$ dimensional vectors $\mathbf{g}_l^{\cdot}$. These forward mappings into progressively higher order polynomial spaces continue until the final polynomial order is 2^L.

Starting from the properties of the Mahalanobis distance, one can show that the q-Order Polynomial Mahalanobis Distance defined in Equation 9 has the following properties:

- $d_{PM}(\mathbf{x}_i, \mathbf{x}_j) = d_{PM}(\mathbf{x}_j, \mathbf{x}_i)$
- $d_{PM}(\mathbf{x}_i, \mathbf{x}_j) > 0$ if $\mathbf{x}_i \neq \mathbf{x}_j$
- $d_{PM}(\mathbf{x}_i, \mathbf{x}_j) = 0$ if $\mathbf{x}_i = \mathbf{x}_j$
- $d_{PM}(\mathbf{x}_i, \mathbf{x}_j) \leq d_{PM}(\mathbf{x}_i, \mathbf{y}) + d_{PM}(\mathbf{y}, \mathbf{x}_j)$

1) Computational Justification: There are two reasons why the polynomial Mahalanobis distance as defined in Equation (9) is computationally feasible. The first is that all of the $L+1$ SVD decompositions have at most N non zero singular values (where N is the number of points in the neighborhood used to estimate the column spaces U). Furthermore, each of the L polynomial projections have at most $d = ((N+2)\cdot(N+1)/2)$ number of terms. The result is that the polynomial Mahalanobis distance can be effectively applied to very large problem domains, as long as the number local neighborhood points is less than a few hundred (see Section III-B for details).

III. Experimental Results

All MATLAB code for implementing the Mahalanobis and Polynomial Mahalanobis distances is freely available from the authors for research purposes.

A. Synthetic Data

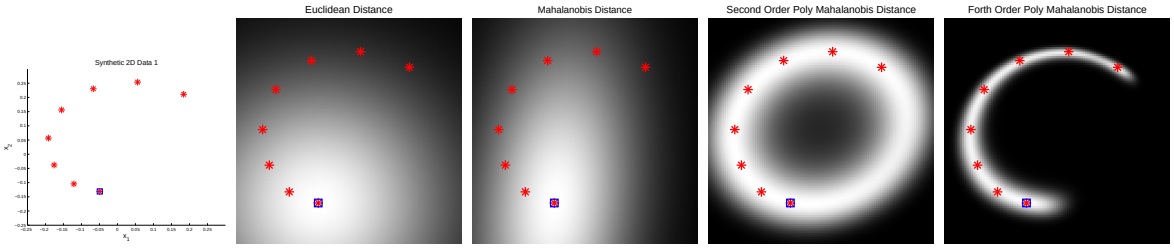
The 2 dimensional synthetic data in Figures 1 and 2 illustrate the specificity property of the distance metrics compared in this paper. All results were generated with $s_{min} = 1e-8$ (see Section II-A.1 for a definition of s_{min}). Figure 1a shows data obtained from a section of a spiral. Figure 1b depicts the Euclidean distance to a reference point (surrounded by a blue square). The more white the image, the closer that region is to the reference point. We can see that, as expected, the Euclidean distance does not follow the sample points in the plot, but rather simply radiates out equally in all directions. Figure 1c shows the Mahalanobis distance from the reference point. This distance is calculated using the sample data in 1a, and therefore does take into account the first order (or linear) structure of the data. Figure 1d, shows the second order Mahalanobis Distance. One can see that, since the data approximately lies in an ellipse, the second order Mahalanobis distance encompasses it, however, it still labels regions as near the reference point, even though no sample data appears there. Figure 1e shows the forth order polynomial Mahalanobis distance, and we see that the points used to compute it are considered "close" to the reference points, while other regions are labeled as far away (i.e. dark regions).

Figure 2a shows the second set of synthetic data used to estimate the Mahalanobis Distance and the Polynomial Mahalanobis Distance. Figure 2b shows the Mahalanobis Distance, and Figures 2c, 2d and 2c show the second, eighth, and thirty second order Polynomial Mahalanobis distances. Once more we see that as the order of the polynomial increases, the resulting distance metrics more tightly "fit" the data, and more effectively exclude regions that have no data.

B. Outdoor Image Data

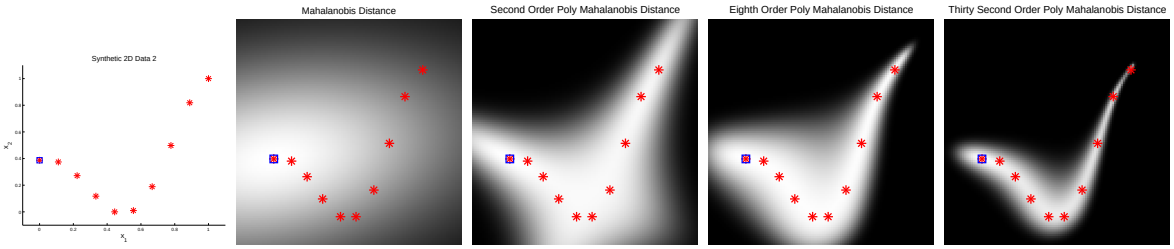
The outdoor image data was collected using a LAGR robot. Stereo is used to identify a ground patch, which is then used to build a model of what the ground looks like. This model is then used to label the remainder of the image. Note that no negative examples (or non-path examples) are used to construct these models. We build three types of models: 1) a Euclidean model defined by the mean value of the pixels in a region and a distance threshold for labeling paths; 2) a Mahalanobis model where a patch in the image is used to construct the distance metric and a distance threshold; and 3) a Polynomial Mahalanobis Model defined by an image patch and distance threshold. These models are constructed by using half the labeled patch to build a model, and the other half to find the threshold for labeling paths. In the case of the Polynomial Mahalanobis distance, the image patch is also used to pick s_{min} and the number of polynomial levels L. The search space for these parameters is $s_{min} = (10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7})$ and $L = 2, 3, 4, 5, 6, 7$. *In all images, each model is constructed using the same image patch.*

Figure 3 through Figure 9 show the results. The leftmost images in these figures show the starting image and the patch (a dark rectangle) used to build the models. Light coloured



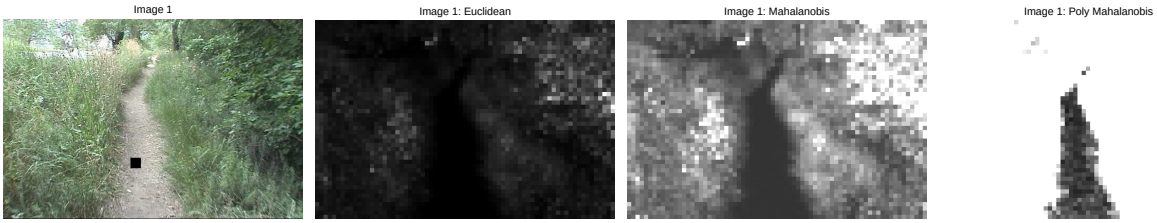
(a) (b) (c) (d) (e)

Fig. 1. Synthetic Data 1.



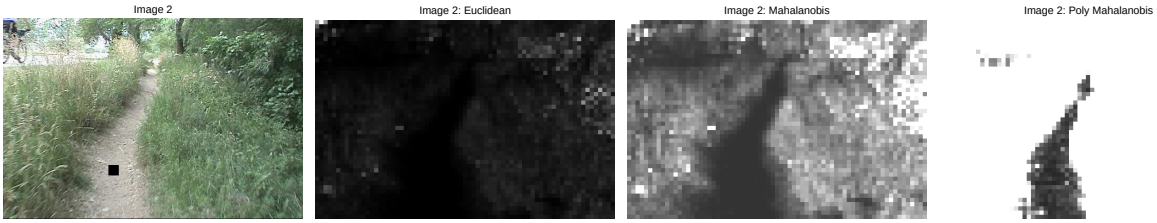
(a) (b) (c) (d) (e)

Fig. 2. Synthetic Data 2.



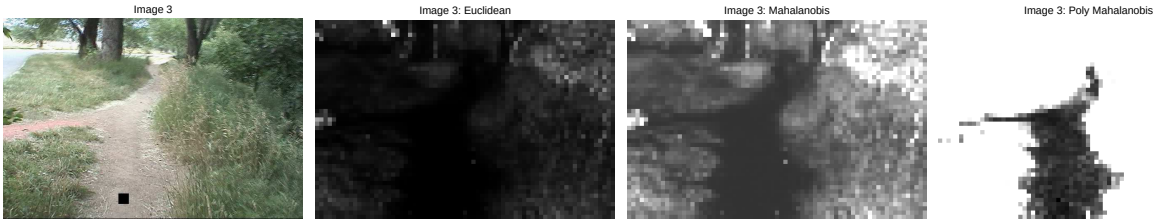
(a) (b) (c) (d)

Fig. 3. Image 1 Using Normalized RGB.



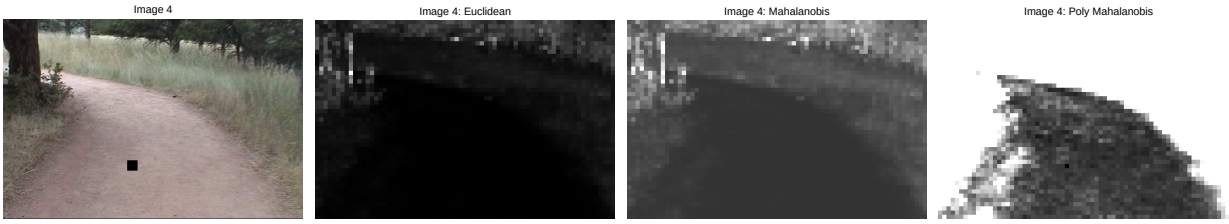
(a) (b) (c) (d)

Fig. 4. Image 2 Using Normalized RGB.



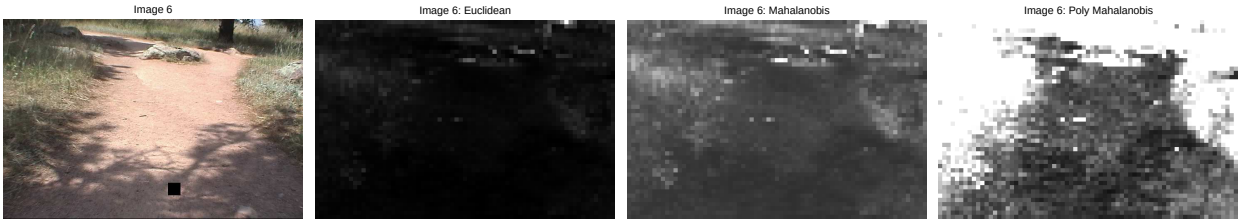
(a) (b) (c) (d)

Fig. 5. Image 3 Using Normalized RGB.



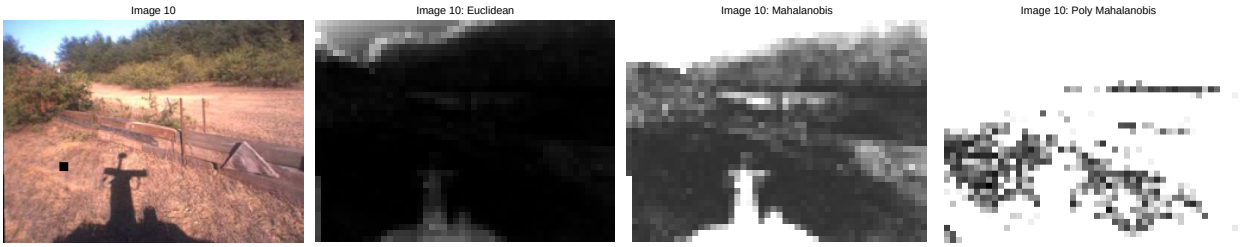
(a) (b) (c) (d)

Fig. 6. Image 4 Using Normalized RGB.



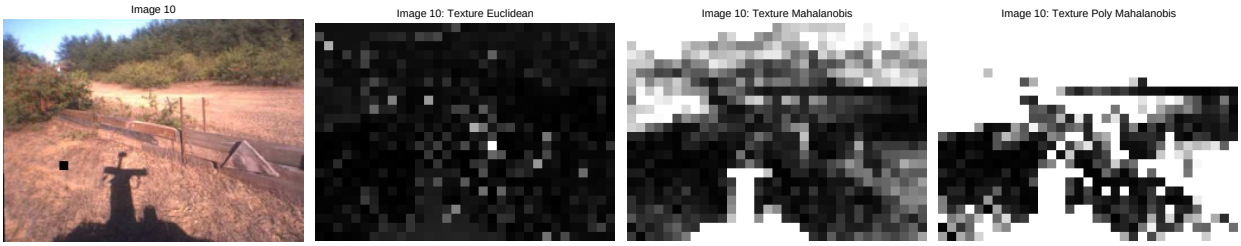
(a) (b) (c) (d)

Fig. 7. Image 6 Using Normalized RGB.



(a) (b) (c) (d)

Fig. 8. Image 10 Using Normalized RGB.



(a) (b) (c) (d)

Fig. 9. Image 10 Using Texture Features.

areas in the image indicate a non-path, while the darker the image patch is, the more confident the model is that that region belongs to a path. In all images presented, we can see that the Euclidean metric is the least effective at segmenting paths, the Mahalanobis distance is slightly better, and the Polynomial Distance is best. In general, the more discriminative the distance metric, the better the segmentation of the path.

In Figures 3 through 8, models are built using normalized pixel values within a 10 by 10 pixel image patch. In Figure 9 (based on the same image as Figure 8) models are built using image texture features. One can see the the texture features are more effective in segmenting this image, when compared to the normalized pixel value. Furthermore, we see that the Polynomial Mahalanobis distance is effective in two very different feature spaces.

IV. Conclusion

The problem of planning smooth trajectories for mobile robots traveling at relatively high speed in natural environments, depends on being able to identify navigable terrain a significant distance ahead. Labeling *safe* or *path* regions in an image sequence is a common way to achieve this far field classification. Many pixelwise classification techniques fail at this task because their similarity metric is not powerful enough to tightly separate path from nonpath, resulting in outliers distributed across the image.

In this paper we have introduced a new and more powerful Polynomial Mahalanobis distance metric for use in classifying path regions in images of natural outdoor environments. Our approach uses only an initial positive sample of a path region to capture the relationships in the data which are most discriminative for path/nonpath classification. We have compared its performance to Euclidean and standard Mahalanobis distance for illustrative synthetic data as well as for challenging outdoor scenes. For both normalized colour and texture features the new metric produced significantly better results.

Acknowledgment

This work has been partially funded by National Science Foundation Awards IIS-0535269 and CNS-0430593. The authors are grateful to the reviewers for their feedback.

References

[1] R. Wallace, K. Matsuzaki, J. Crisman, Y. Goto, J. Webb, and T. Kanade, "Progress in robot road-following," in *Proc. of 1986 IEEE International Conference on Robotics and Automation*, vol. 3, 1986, pp. 1615–1621.

[2] J. Crisman and C. Thorpe, "Unscarf, a color vision system for the detection of unstructured roads," in *Proc. 1991 Int. Conf. on Robotics and Automation*, Sacramento, CA, April 1991, pp. 2496–2501.

[3] D. Kuan, G. Phipps, and A.-C. Hsueh, "Autonomous robotic vehicle road following," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 5, pp. 648 – 658, 1988.

[4] M. Turk, D. Morgenthaler, K. Gremban, and M. Marra, "Vits-a vision system for autonomous land vehicle navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 342–361, May 1988.

[5] C. Rasmussen, "Combining laser range, color, and texture cues for autonomous road following," in *Proc 2002 Int Conf on Robotics and Automation*, 2002.

[6] 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*, Cambridge, USA, June 2005.

[7] P. Fieguth and S. Wesolkowski, "Highlight and shading invariant color image segmentation using simulated annealing," in *Proc. EMMCVPR*, vol. LNCS 2134, 2001, pp. 314–327.

[8] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (PAMI)*, vol. 27, no. 12, pp. 1945– 1959, Dec 2005.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge, UK: Cambridge University Press, 1992.