AIAA Guidance, Navigation, and Control Conference
13 - 16 August 2012, Minneapolis, Minnesota

AIAA 2012-4702

Combining Distance Measures to Classify Terrain Image Sequence Based on Dynamic Texture Model

Koki Fujita*

Kyushu University, Fukuoka, 819-0395, Japan

Naoyuki Ichimura†

National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, 305-8568, Japan.

Utilizing image sequences obtained from onboard cameras is important to improve autonomous mobility for planetary rovers. In this paper, we propose a method based on a linear dynamical system model called a Dynamic Texture to classify terrain image sequences which contain different types of properties. Among some physical properties included in image sequences, we focus on static properties such as a soil texture and dynamic properties such as a constant image velocity. Distance measures based on an observability space and a time-series model called an autoregressive (AR) model are used for classifying the properties. A cross-validation test using real image sequences shows that the latter distance measures are more advantageous than the former measures with respect to the dynamic properties. Thus we combine two distance measures specific to different properties to gain the classification performance. Experimental results demonstrate the effectiveness of combining distance measures.

I. Introduction

In order to improve autonomous mobility for next-generation planetary rovers, several investigators have recently focused on non-geometric features of surrounding terrain such as texture, color, and wheel-soil interaction mechanics.[1–5] In this paper, we propose a method using image sequences obtained from onboard cameras to recognize soil textures and steady-state velocities of a rover. While the past works discretely utilize image data from multi-spectral imager, CCD camera, and laser range sensor, our work aims at applying image sequences. From image sequences of surrounding terrain in which terrain texture moving in steady-state, a set of parameters of a linear dynamical system model called a Dynamic Texture can be computed. Then soil textures and velocities of a rover can be recognized through classification based on the model parameters.

Selecting distance measures used in classification of the model parameters is essential to distinguish different properties contained in terrain image sequences. The distance measures based on the observability space for the Dynamic Texture model were considered in our previous work.[6] Although a couple of the distance measures are fairly efficient to classify both static and dynamic properties, the other distance measures can be used in a complementary manner to improve classification performance. We introduce distance measures based on a time-series model called an autoregressive (AR) model. Subsequently a method combining different distance measures is proposed considering that the distance measures based on the AR model are supposed to be more advantageous as far as dynamic properties are concerned.

In the rest of this paper, an estimation and a recognition techniques for terrain image sequence utilizing Dynamic Texture model are briefly described. Next, performances of a couple of distance measures based on the AR model as well as the ones based on the observability space are discussed to address both the static and the dynamic properties. Finally, a method combining two distance measures is proposed to classify

*Assistant Professor, Dept. of Aeronautics and Astronautics, 744, Motooka, Nishi-ku, Fukuoka, Japan.

†Senior Researcher, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba Central 2, 1-1-1, Umezono, Tsukuba, Japan.

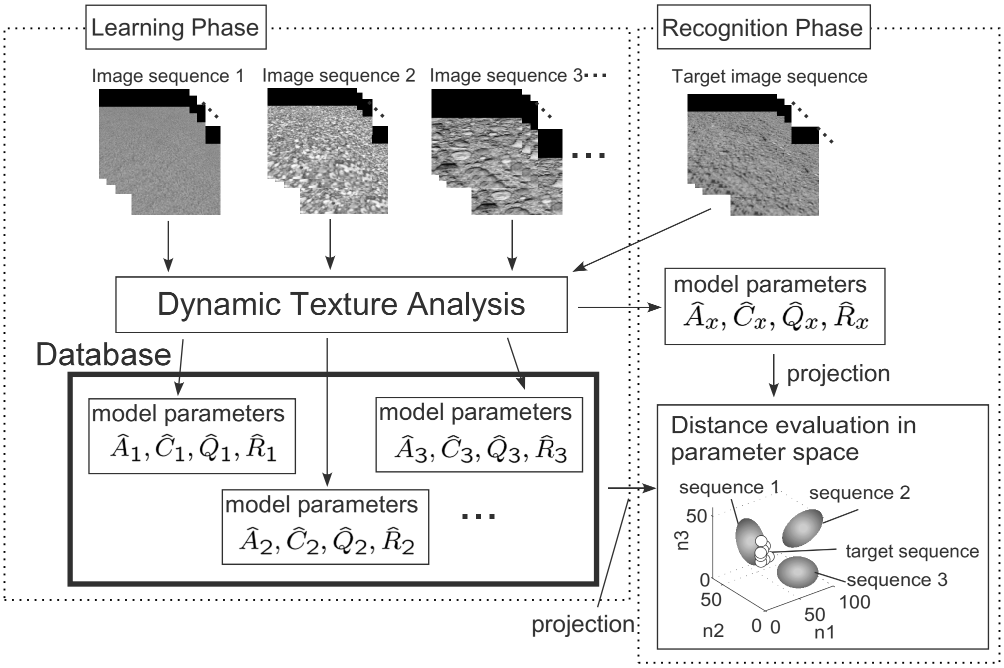

Learning Phase
Image sequence 1
Image sequence 2
Image sequence 3
Recognition Phase
Target image sequence
Dynamic Texture Analysis
model parameters $\hat{A}_x, \hat{C}_x, \hat{Q}_x, \hat{R}_x$
projection
Database
model parameters $\hat{A}_1, \hat{C}_1, \hat{Q}_1, \hat{R}_1$
model parameters $\hat{A}_3, \hat{C}_3, \hat{Q}_3, \hat{R}_3$
model parameters $\hat{A}_2, \hat{C}_2, \hat{Q}_2, \hat{R}_2$
Distance evaluation in parameter space
sequence 1
sequence 2
target sequence
sequence 3
n3
50
0
50
n2
0 0
50
100
n1
projection

Figure 1. Terrain classification based on Dynamic Texture.

terrain image sequences with respect to different physical properties, and the effectiveness of the proposed method is demonstrated through an ROC analysis applied to real image sequences.

II. Terrain Classification Utilizing Dynamic Texture Model

A terrain classification method in this work is schematically shown in Figure 1. It is divided into two phases called a Learning Phase and a Recognition Phase. Firstly, in the Learning Phase, a database of the Dynamic Texture model parameters is constructed for various types of soils on terrain surface and rover (or onboard camera) motions. Secondly, in the Recognition Phase, a terrain image sequence is obtained on site, and it is classified using some distance measures in a model parameter space. Classification is performed by applying a threshold value to the distance measures computed among a terrain image sequence and learned image sequences. If any of the distance measures is smaller than the threshold value assigned to a class of image sequences in the Learning Phase, the terrain image sequence is classified to the assigned class. Otherwise, the terrain image sequence is rejected.

III. Estimation and Recognition of Dynamic Texture Model

III.A. Estimation of the Dynamic Texture Model

Given constant linear motion of camera mounted on the vehicle and homogeneous and isotropic properties of the terrain texture, the motion image sequence captured from the camera can be reduced to a set of parameters in a linear dynamical system model as follows:

$$\begin{cases} x(k+1) = Ax(k) + v(k), & v(k) \sim \mathcal{N}(0,Q);\ x(0) = x_0, \\ y(k) = Cx(k) + w(k), & w(k) \sim \mathcal{N}(0,R), \end{cases} \tag{1}$$

where $k = 0, 1, 2, \cdots$ is the discrete time instant, $y(k) \in \mathbb{R}^m$ is a vector of measured pixel brightness values in the k-th image frame, m equals the number of pixels in an image frame, $x(k) \in \mathbb{R}^n$ is an n-dimensional state vector, and $v(k) \in \mathbb{R}^n$ and $w(k) \in \mathbb{R}^m$ are the noise vectors driven by white Gaussian noise. As seen in the equations, the above dynamical system is characterized by the parameter matrices $A \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{m\times n}$, $Q \in \mathbb{R}^{n\times n}$, and $R \in \mathbb{R}^{m\times m}$.

Whereas the previous works on Dynamic Texture as seen in a reference[7] applied a suboptimal estimation algorithm utilizing Principal Component Analysis (called PCA-ID) to estimate the parameter matrices, A, C, Q, and R, this work adopts another algorithm based on the components of 2-Dimensional Discrete Cosine Transform (2D-DCT) and an efficient system identification algorithm called N4SID.[8] The latter method has an advantage in that optimal solution for the dynamical model is obtained within less computational time.

The estimation algorithm in this work contains two steps as follows:

STEP1: Original $M \times N$-pixels image data from the terrain image sequence, $f_{i,j},\ i = 1, 2, \cdots, M,\ j = 1, 2, \cdots, N$ are transformed into $F_{k,l},\ k = 1, 2, \cdots, M,\ l = 1, 2, \cdots, N$ such that

$$F_{k,l} = C_k C_l \sum_{i=1}^{M} \sum_{j=1}^{N} f_{i,j} \cos\left(\frac{(2i-1)k\pi}{2M}\right) \cos\left(\frac{(2j-1)l\pi}{2N}\right) \tag{2}$$

$$\text{where} \quad C_{k \text{ or } l} = \begin{cases} 1/\sqrt{2}, & \text{if } k \text{ or } l = 1 \\ 1, & \text{else} \end{cases}$$

Since $F_{k,l}$ is obtained by linear transformation from the original image data, their principal properties should be preserved in the output components for the lower dimensional spatial frequencies. Thus, among $m(= M \times N)$ components of 2D-DCT output for the original image, only $m_c(= M_c \times N_c,\ m_c < m)$ ones are applied to the N4SID algorithm. If $y_c(k)$ is defined as $[F_{1,1}(k),\ F_{1,2}(k),\ \cdots F_{M_c,N_c}(k)]^T \in \mathbb{R}^{m_c}$, the dynamical system model corresponding to Eq. (1) is described such that

$$\begin{cases} x_c(k+1) = A_c x_c(k) + v_c(k), & v_c(k) \sim \mathcal{N}(0, Q_c);\ x_c(0) = x_{c0}, \\ y_c(k) = C_c x_c(k) + w_c(k), & w_c(k) \sim \mathcal{N}(0, R_c), \end{cases} \tag{3}$$

where the subscript c denotes variable vectors or parameter matrices corresponding to the low-dimensional 2-D DCT components.

STEP2: N4SID algorithm[8] is applied to $y_c(k),\ k = 1, 2, \cdots, K$ in **STEP1**, and the linear dynamical system paraters such as A and C are computed for given order of the system n.

III.B. Recognition of the Dynamic Texture Model

Since the estimated models for the linear dynamical system are characterized by the parameter matrices such as A and C, they can be identified by the column space of the extended observability matrix such that

$$\mathcal{O}_\infty(M) = \left[C^T\ (CA)^T\ (CA^2)^T\ \cdots\right]^T. \tag{4}$$

For a large enough number n, the above extended observability matrix is approximated by the finite observability matrix,

$$\mathcal{O}_n(M) = \left[C^T\ (CA)^T\ (CA^2)^T\ \cdots (CA^{n-1})^T\right]^T. \tag{5}$$

In the previous work,[6] based on the above observability space, three distance measures called "Euclidean distance", "Martin's distance", and "Kernel Density Function (KDF) on the Stiefel Manifold" were introduced. The performance of the classification for terrain image sequence was discussed based on the two different feature classes through an ROC analysis, which will be briefly shown in the later section.

IV. Distance Measures Based on AR Model

Aiming at data mining, clustering, and classifying for time-series data in the framework of a linear dynamical system model such as an autoregressive (AR) model, some metrics have been proposed in the past studies.[9,10] Focusing on the Dynamic Texture model, several typical distance measures were applied in a work[11] for image sequences containing natural scene such as waving trees in order to compare their performance. According to the above studies, there are three representative distance measures for the time-series model:

1. **Kullback-Leibler distance (d_{KL}):** A measure can be derived based on probability densities for time series of state variables, $\boldsymbol{x}(k)$ $(k = 1, 2, \cdots T)$, which are estimated by a certain length of real sequence $(\boldsymbol{y}(k),\ k = 1, 2, \cdots T)$. The Kullback-Leibler divergence or equivalently called the Kullback-Leibler distance is one of the measures to compare two probability distributions. Given two sets of the sequences of the state variables are described as $X_1 = \{\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \cdots \boldsymbol{x}_T\}$ and $X_2 = \{\boldsymbol{x}'_1, \boldsymbol{x}'_2, \cdots \boldsymbol{x}'_T\}$, as well as their probability density functions, $f_1(X_1)$ and $f_2(X_2)$, it is defined such that

$$d_{KL}(f_1,\ f_2) = E\left[\frac{f_1(X_1)}{f_2(X_2)}\right]. \tag{6}$$

If two sets of the model parameters for the Dynamic Textures, $M_1 = (A_1, Q_1)$, and $M_2 = (A_2, Q_2)$ are estimated from the different image sequences, the Kullback-Leibler distance is derived for M_1 and M_2 as follows:

$$d_{KL}(M_1, M_2) = \sum_{l=1}^{L/2}\left[\mathrm{Tr}\{F_1(l)F_2^{-1}(l)\} - \ln\frac{|F_1(l)|}{|F_2(l)|} - n\right], \tag{7}$$

where $F(l)$ $(l = 1, \cdots, L)$ is basically computed from Fourier transform coefficients of the state variables. If time series of the i-th component of n-diminsional state vector $\boldsymbol{x}(k)$ $(k = 1, \cdots, K)$ is described as $\boldsymbol{x}(i, 1 : K) \in \mathbb{R}^{1\times K}$ $(i = 1, \cdots, n)$, $F(l)$ is derived using L-dimensional row vector of Fourier coefficients for each time series (which is efficiently obtained by a Fast Fourier Transform algorithm), $f(i, 1 : L) = \mathrm{FFT}\left(\boldsymbol{x}(i, 1 : K)\right)$ $(i = 1, \cdots, n)$ such that

$$F(l) = \sum_{j=l-H}^{l+H} \boldsymbol{f}_j\boldsymbol{f}_j^*, \quad \boldsymbol{f}_j = f(1 : n, j) \in \mathbb{C}^{n\times 1} \tag{8}$$

for appropriate smoothing parameter, H.

2. **Cepstral distance (d_{cep}):** A cepstrum is a popular measure to represent features in speech signal and derived from Fourier transform of the logarithm of the spectrum for original time-series data. While it is originally defined for one-dimensional signal, a cepstrum for multidimensional AR model is derived as the following equations.

$$\begin{aligned} c_x(k) &= \quad 0, & \text{if } k \leq 0,\\ &= \tfrac{1}{k}\textstyle\sum_{i=1}^{n} p_i^k, & \text{if } k > 0, \end{aligned} \tag{9}$$

where p_i $(i = 1, 2, \cdots, n)$ describe eigen values of a parameter matrix of the Dynamic Texture model, A. A distance measure for the two different models can be defined such that

$$d_{cep}(M_1, M_2) = \sum_{k=0}^{K_t} |c_{x_1}(k) - c_{x_2}(k)|^2 \tag{10}$$

by appropriately choosing K_t to terminate the summation.

3. **Distance based on feature vector (d_{fv}):** The other measure for the linear dynamical system model can be directly derived from the model parameters such as A, C, Q, and R. Considering dynamical properties should appear in the matrices A and Q of the state equation in Eq. (1) (or in Eq. (3)), a feature vector is defined as $\boldsymbol{v}(M) = [\lambda_1^{(A)}, \cdots, \lambda_n^{(A)}, \lambda_1^{(Q)}, \cdots, \lambda_n^{(Q)}]$ by using n eigen values of A and Q, respectively. A distance measure is defined as a Euclidean distance for the feature vectors of different models as follows:

$$d_{fv}(M_1, M_2) = ||\boldsymbol{v}(M_1) - \boldsymbol{v}(M_2)||_2 = \sqrt{\sum_{i=1}^{n}\left(\lambda_i^{(A_1)} - \lambda_i^{(A_2)}\right)^2 + \left(\lambda_i^{(Q_1)} - \lambda_i^{(Q_2)}\right)^2}. \tag{11}$$

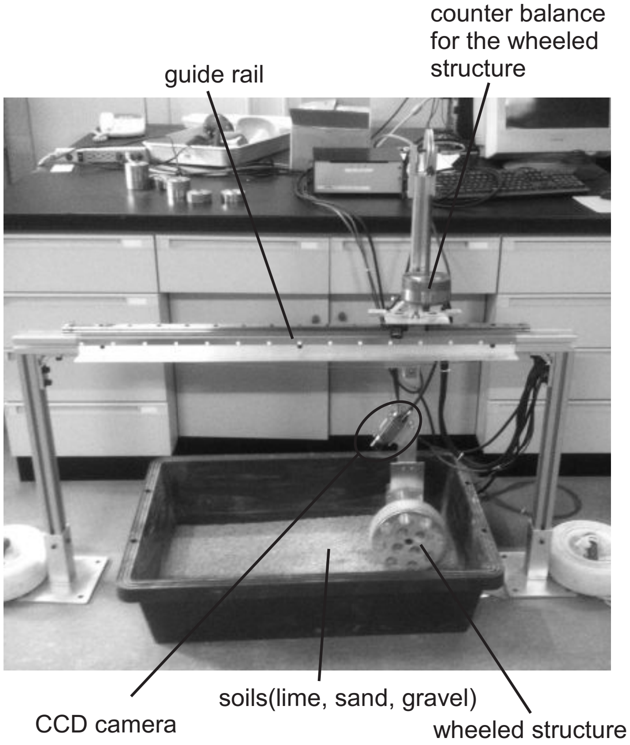


Figure 2. Testbed for acquiring terrain image sequences.

While the above three distance measures can be computed from A and Q for the assumed AR model,[11] the Kullback-Leibler distance is more efficiently obtained from the state vectors $\boldsymbol{x}(k)$, $k = 1, \cdots, K$, which are directly estimated by the N4SID algorithm.

In order to see performances of the above distance measures for the AR model, a 2-fold cross-validation test and a Receiver Operating Characteristic (ROC) analysis[12] were conducted for an experimental data using a testbed shown in Figure 2. Real image sequences for four types of the terrain textures taken from a camera moving in steady-state translational motions were applied to estimate the Dynamic Texture model parameters, A_c, C_c, Q_c, R_c, and $\boldsymbol{x}_c$ in Eqs. (3). Then recognition rates were computed from the results and the corresponding ROC plots are depicted for various threshold values of each distance measure to classify the image sequences.

As shown in Figure 3, the terrain sequences contain four types of soils (lime, two types of sand, and gravel) identically-distributed in the image frames, and three constant and translational image velocities

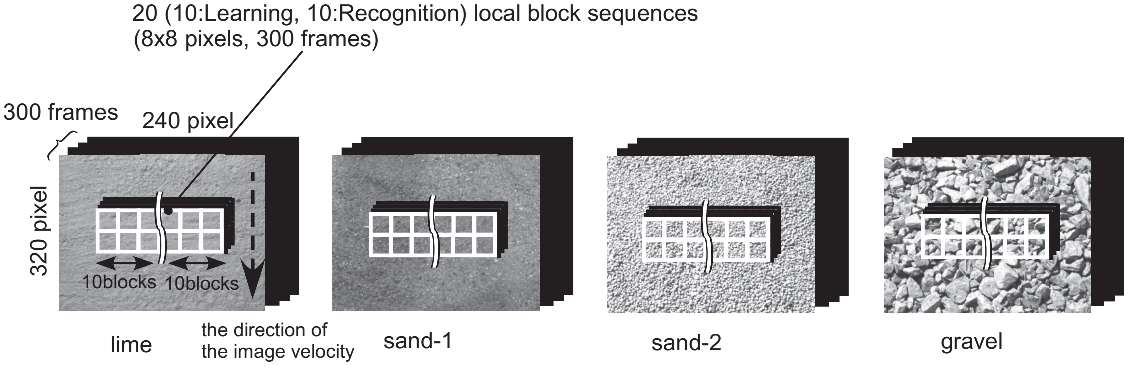


Figure 3. Real image sequences applied to the proposed methods.

Table 1. Table of combination

		Soil texture			
		lime (A)	sand-1 (B)	sand-2 (C)	gravel (D)
Image velocity	$\boldsymbol{v_1}$ **(a)**	Aa	Ba	Ca	Da
	$\boldsymbol{v_2}$ **(b)**	Ab	Bb	Cb	Db
	$\boldsymbol{v_3}$ **(c)**	Ac	Bc	Cc	Dc

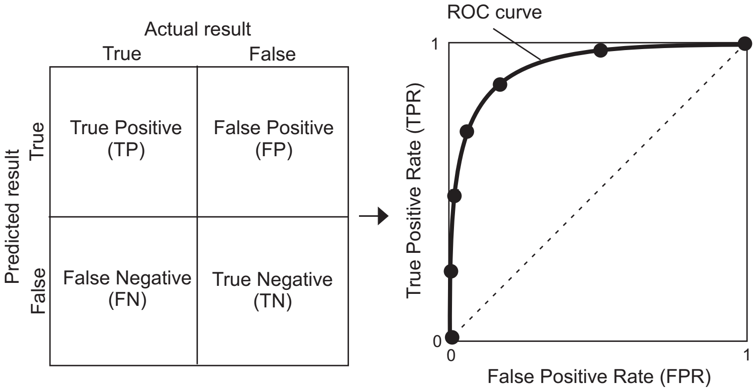


Figure 4. Receiver Operating Characteristic (ROC).

(named v_1, v_2, and v_3), which are generated from the constant torques, $T_0, 2T_0$, and $3T_0$ (T_0: reference torque) of the testbed wheel. Hence, they can be classified into a maximum of twelve image sequences as shown in Table 1.

Applying the 2-fold cross-validation test, twenty local block images of 8 × 8 pixels are cropped from the original images, and ten block image sequences consiting of 300 frames are used for the Learning Phase, and the rest of the ten block image sequences are used for the Recognition Phase. Note here that the dimension of the AR model in the state equation in Eq. (3) is fixed so that $n = 10$, and the number of the 2D-DCT components applied for the N4SID algorithm is fixed so that $M_c = N_c = 8$. We consider two types of feature classes in this experiment. One is a class consisting of twelve dynamical properties caused from the image velocities as well as the static property caused from the soil textures (corresponding to the twelve matrix elements such as "Aa" in Table 1), and another is a class consiting of the four static properties caused only from the soil textures (corresponding to the four columns "A", "B", "C", and "D" in Table 1). In this work, the former is called a dynamic texture class, and the latter is called a static texture class.

ROC plots are depicted for various threshold values for each distance measure through computing two types of recognition rates, the true positive rate (TPR) and the false positive rate (FPR). They are defined as follows based on the relationship between predicted and actual results as shown in the left-hand side of Figure 4:

$$\mathrm{TPR} = N_{\mathrm{TP}}/(N_{\mathrm{TP}} + N_{\mathrm{FN}}), \quad \mathrm{FPR} = N_{\mathrm{FP}}/(N_{\mathrm{FP}} + N_{\mathrm{TN}}). \tag{12}$$

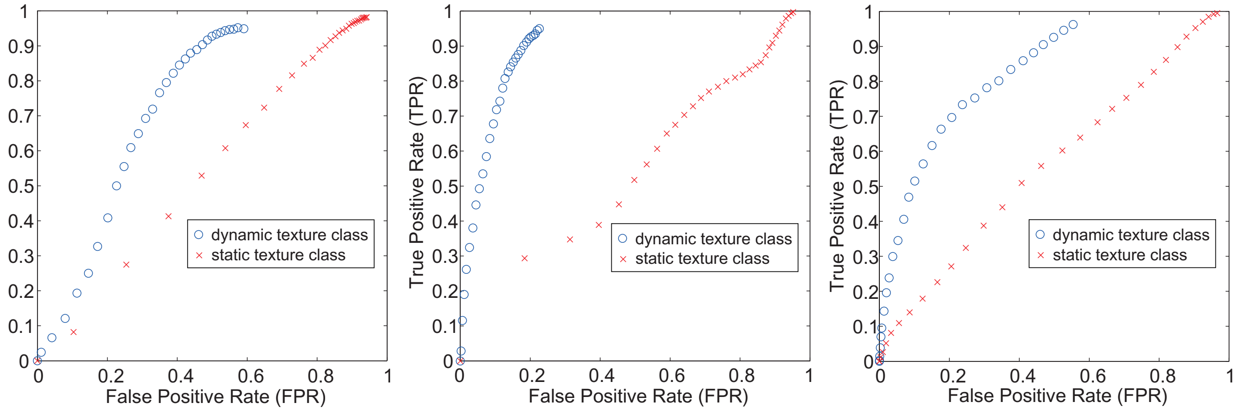
Since optimal classifier should obtain enough high true positive rate relative to the corresponding false positive rate, curve in ROC space is desirable to be skewed to upper left corner as shown in the right-hand side of Figure 4.

In this work, for each distance measure thirty threshold values were set such that they are equally distributed in the range of the distances obtained for the same feature class (the static texture or the dynamic texture classes) of the block image sequences during the Learning Phase. ROC plots shown later in this section are depicted for those different threshold values.

TPRs computed for particular threshold values for each distance measure are shown in Table 2. ROC plots for the same cross-validation tests are shown in Figure 5. In the case of Table 2, the threshold value is fixed at the maximum value in the range of the distances obtained for the same feature class in the Learning Phase, which correspond to the plots for the highest values of TPR in Figure 5. Table 3 and Figure 6 are

Table 2. The true positive rates based on the AR model.

Dynamic texture class				Static texture class			
	1st	2nd	mean		1st	2nd	mean
K-L dist.	96.0%	96.8%	96.4%	**K-L dist.**	98.2%	99.8%	99.0%
Cepstral dist.	95.0%	94.3%	94.6%	**Cesptral dist.**	99.7%	99.1%	99.4%
The dist. based on the feature vector	96.3%	90.8%	93.5%	**The dist. based on the feature vector**	99.5%	99.8%	99.6%



(a) K-L distance

(b) Cepstral distance

(c) The distance based on the feature vector

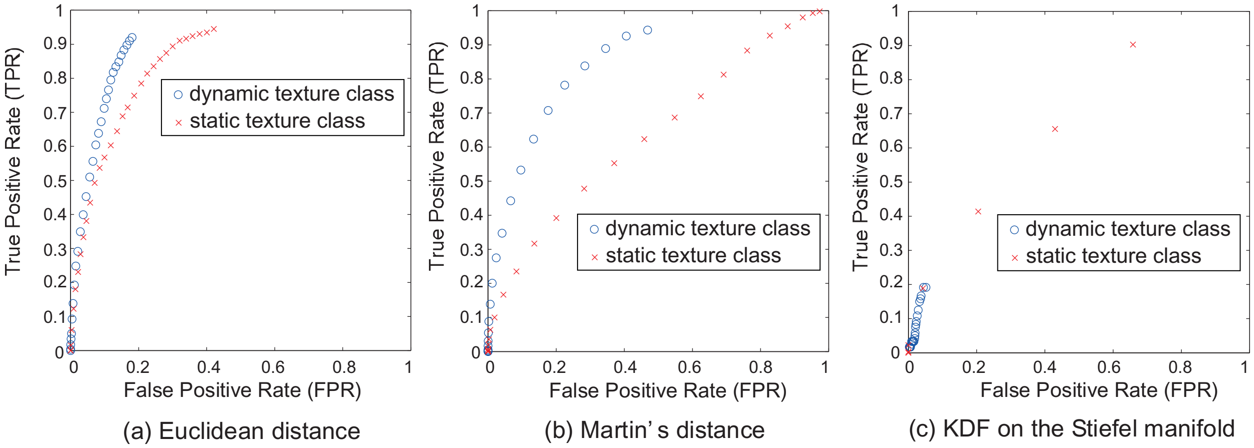
Figure 5. ROC plots for the distance measures based on the AR model (for the 1st test).

shown to compare with the results for the distance measures based on the observability space in the previous work.[6]

Table 3. The true positive rates based on the observability space.

Dynamic texture class				Static texture class			
	1st	2nd	mean		1st	2nd	mean
Euclidean dist.	90.5%	82.4%	86.5%	**Euclidean dist.**	98.9%	97.3%	98.1%
Martin's dist.	96.4%	95.8%	96.1%	**Martin's dist.**	99.9%	99.7%	99.8%
KDF on the Stiefel manifold	5.0%	9.2%	7.1%	**KDF on the Stiefel manifold**	83.3%	87.5%	85.4%

As shown in Table 2, for the both feature classes, all the results show relatively high performances compared to the previous results seen in Table 3. On the other hand, focusing on the dynamic texture class, the ROC plots in Figure 5 show relatively high performances, and the cepstral distance shows the results competitive to the one for the Euclidean distance based on the observability space, as seen in Figure 6(a). One of the reasons for these results is considered that the distance measure based on the AR model effectively represents the dynamic property of a state-space model, which should appear in the parameter matrices, A (or A_c) and Q (or Q_c) in Eq. (1) (or in Eq. (3)). Meanwhile the distance measures based on the observability space depict more desirable ROC curves for the static texture class, and the Euclidean distance (Figure 6(a)) shows the best performance among six distance measures. The distance measures based on the observability space have a distinct advantage in that they are contributed from the parameter matrix, C, which presumably contains non-dynamical properties in the Dynamic Texuture model.



(a) Euclidean distance

(b) Martin’s distance

(c) KDF on the Stiefel manifold

Figure 6. ROC plots for the distance measures based on the observability matrix (for the 1st test).

Table 4. The true positive rates for the stepwise terrain classification.

1st classification (for soil texture)				2nd classification (for image velocity)			
	1st	2nd	mean		1st	2nd	mean
The combined dist. (proposed)	94.5%	98.7%	96.6%	The combined dist. (proposed)	100.0 %	100.0%	100.0%
The cepstral dist.	99.7%	99.1%	99.4%	The cepstral dist.	100.0%	100.0%	100.0%
The Euclidean dist.	94.5%	98.7%	96.6%	The Euclidean dist.	47.4%	45.0%	46.2%

V. A Terrain Classification Method Combining Distance Measures

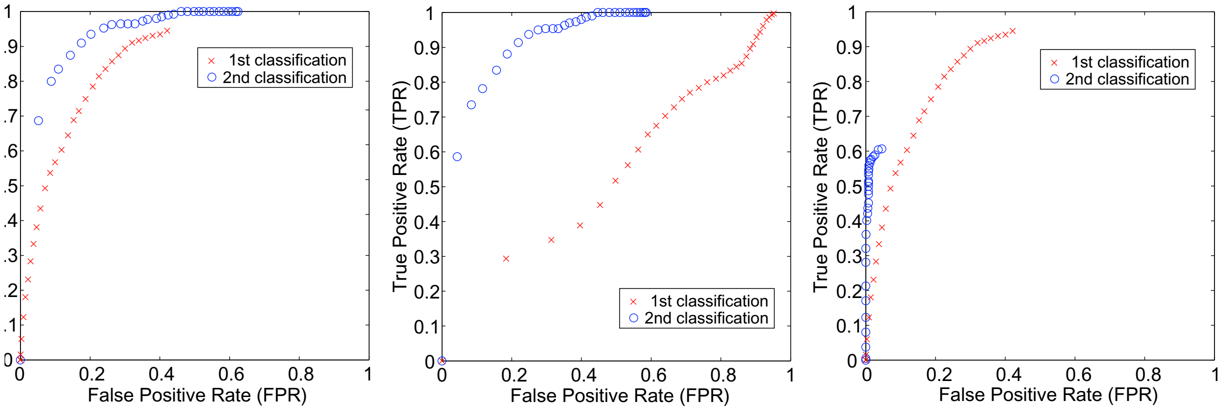
Considering the results in the previous section, a stepwise method combining different distance measures is proposed. In the proposed method, two distance measures are applied to improve the performance of distinguishing different type of the features containing in the image sequences.

After estimating the Dynamic Texture models in the Learning Phase, in the following Recognition Phase, first step of the static feature classification is proceeded using a measure based on the observability space, then second step of the dynamic feature classification is applied using a measure based on the AR model.

To validate the effectiveness of the proposed method, 2-fold cross-validation tests were conducted for the same image sequences as in the previous section. For the 1st step, the Euclidean distance was applied to classify the static feature (“A”, “B”, “C”, and “D” in Table 1), and for the 2nd step, the cepstral distance was applied to classify the dynamic feature (“a”, “b”, and “c” in Table 1). The 2nd step was implemented for each static feature class obtained from the 1st step. The cross-validation tests using only one measure (the Euclidean or the cepstral distance) for the both classification steps were also conducted for comparison. The results of the TPR using certain threshold values of the distance measures are shown in Table 4, and ROC plots for various threshold values of the distance measures are shown in Figure 7.

As for Table 4, the threshold values were fixed in the same way as in the result of Table 2. On the other hand, in contrast to the ROC analysis in the previous section, the recognition rates such as TPR and FPR in the proposed stepwise classification can be affected by results of the preceding classification step. Therefore, the threshold value for the 1st step (to classify the static texture) was fixed at the maximum value in the range of the distances obtained for the same feature class in the Learning Phase. In the 2nd step, using the threshold values set in the same way as in Figure 5, TPR and FPR were computed only for the block sequences which were predicted to be true (belonging to TP and FP in Figure 4) in the previous step.

As seen in the results of Table 4, all the TPRs show relatively high performances except for the 2nd classification using only the Euclidean distance. It is also validated in the results of ROC plots in Figure 7, the highest TPR for the 2nd classification in Figure 7(c) is less than the other highest values of the TPRs



(a) The combined distance (proposed)

(b) Cepstral distance

(c) The Euclidean distance

Figure 7. ROC plots for the stepwise terrain classification (for the 1st test).

using the different sets of distance measures.

On the other hand, ROC plots clearly show the performance of the proposed method. As shown in Figure 7(a), only the ROC plots for the proposed method depict desirable curves, that is, they achieve convincing high TPRs compared to the corresponding FPRs for the both classification steps.

VI. Conclusion

This work proposes to combine different distance measures in a model parameter space, aiming at classifying terrain image sequences taken by rover onboard cameras with respect to static and dynamic properties. Utilizing the parameter estimation and the recognition techniques based on the Dynamic Texture model, performances of a couple of the distance measures based on AR model as well as the previously introduced measures based on the observability space are discussed. Subsequently, a stepwise method is proposed to classify the terrain image sequences with respect to each feature caused from soil textures and image velocities. The effectiveness of the proposed method was validated through a 2-fold cross-validation test using a set of real image sequences.

In future work, various terrain image sequences obtained in more realistic planetary environment should be applied to the proposed method.

Acknowledgments

The authors would like to thank the Japan Society for the Promotion of Science for supporting this study as a part of Grants-in-Aid for Scientific Research (C) (No. 22500176).

References

[1] Helmick, D., Angelova, A., and Matthies, L., "Terrain Adaptive Navigation for Planetary Rovers," *Journal of Field Robotics*, Vol. 26, No. 4, 2009, pp. 391–410.

[2] Halatci, I., Brooks, C. A., and Iagnemma, K., "Terrain Classification and Classifier Fusion for Planetary Exploration Rovers," *Proceedings of the IEEE Aerospace Conference*, 2007, pp. 1–11.

[3] Ishigami, G., Nagatani, K., and Yoshida, K., "Path Planning for Planetary Exploration Rovers and Its Evaluation based on Wheel Slip Dynamics," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 2361–2366.

[4] Dima, C. S., Vandapel, D., and Hebert, M., "Classifier Fusion for Outdoor Obstacle Detection," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004, pp. 665–671.

[5] Iagnemma, K., Brooks, C., and Dubowsky, S., "Visual, Tactile, and Vibration-Based Terrain Analysis for Planetary Rovers," *Proceedings of the IEEE Aerospace Conference, Vol. 2*, 2004, pp. 841–848.

[6] Fujita, K. and Ichimura, N., "A Terrain Classification Method for Planetary Rover Utilizing Dynamic Texture," *Proceedings of the AIAA Guidance Navigation, and Control Conference*, 2011, pp. 184–196.

[7] Saisan, P., Doretto, G., Wu, Y. N., and Soatto, S., "Dynamic Texture Recognition," *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01), Vol.2*, 2001, pp. 58–63.

[8] Overschee, P. V. and Moor, B. D., "N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems," *Automatica*, Vol. 30, No. 1, 1994, pp. 75–93.

[9] Corduas, M. and Piccolo, D., "Time Series Clustering and Classification by the Autoregressive Metric," *Computational Statistics and Data Analysis*, Vol. 52, 2008, pp. 1860–1872.

[10] Bagnall, A. J. and Janacek, G. J., "Clustering Time Series from ARMA Models with Clipped Data," *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '04)*, 2004, pp. 49–58.

[11] Woolfe, F. and Fitzgibbon, A., "Shift-Invariant Dynamic Texture Recognition," *Proceedings of the 9th European Conference on Computer Vision (ECCV), Part II*, 2006, pp. 549–562.

[12] Provost, F. and Fawcett, T., "Robust classification for imprecise environments," *Machine Learning*, Vol. 42, 2001, pp. 203–231.