AIAA Guidance, Navigation, and Control Conference
08 - 11 August 2011, Portland, Oregon

AIAA 2011-6580

A Terrain Classification Method for Planetary Rover Utilizing Dynamic Texture

Koki Fujita[*] Naoyuki Ichimura[†]

A novel terrain classification technique utilizing motion image sequence taken from planetary rover on-board camera is proposed. The proposed method has an advantage in that it can remotely estimate types of surrounding tarrains, and also has possibility to distinguish properties with dynamic interaction between rover body (wheels) and terrain surface. The characteristics of terrain image sequence is recognized based on a linear dynamical system model called Dynamic Texture. The Dynamic Texture is estimated as a set of parameter matrices, which construct a parameter space such as an observability space. In this paper, experimental results are shown to validate the proposed scheme based on real terrain image sequences obtained from a testbed. And the recognition rates for several distance measures computed from the estimated Dynamic Texture models are evaluated.

I. Introduction

In order to improve autonomous mobility of planetary rover, many works have recently focused on terrain classification and identification of wheel-soil interaction mechanics.[1−4]

Most of them propose to utilize on-board sensors such as multi-spectral imagers, CCD cameras, laser range sensor, and accelerometer to estimate “non-geometric” features of the rover’s surrounding terrain. In this paper, non-geometric feature is defined as textures of homogeneously-distributed terrain which are originated from the sizes of soil particles as well as mechanical interaction properties between rover body and soil.

As for the imaging sensors involved in the past works, while they utilize static images, this work proposes to utilize motion image sequences of terrain surface taken from the rover camera. Unlike the conventional techniques to classify terrain surfaces based on single or stereo camera images, the proposed method improves to measure visual saliences and has possibility to remotely estimate properties of dynamic interaction between rover body (wheels) and terrain surface, such as relative velocity, slippage, and sinkage.

Given constant linear motion of a camera and the homogeneous and isotropic properties of terrain texture, the motion image sequence can be reduced to a set of parameters of the Dynamic Texture model.[5] The estimated parameters contain unique properties not only with visual salience for the terrain surface but also with dynamics in camera (or vehicle) motion and terra-mechanics in surrounding terrain.

Aiming at validating the concept to classify terrain image sequences based on the Dynamic Texture model, this work shows experimental results for different types of soils and camera motion by using a testbed. Results of a cross validation test and a receiver operating characteristic (ROC) analysis shows feasibility of the proposed method, and issues to be improved in future works.

II. Overview of the Proposed Method

In this work, a terrain classification method is proposed as an online estimation scheme installed for planetary rover. The schematic view of the proposed method is shown in Fig. 1. As shown in the figure, the

[*] Assistant Professor, Dept. of Aeronautics and Astronautics, Kyushu University, 744 Mtooka, Nishi-ku, Fukuoka, 819-0395, JAPAN.

[†] Senior Researcher, National Institute of Advanced Science and Technology (AIST), Tsukuba Central 2, 1-1-1, Umezono, Tsukuba, Ibaraki, 305-8568, JAPAN.

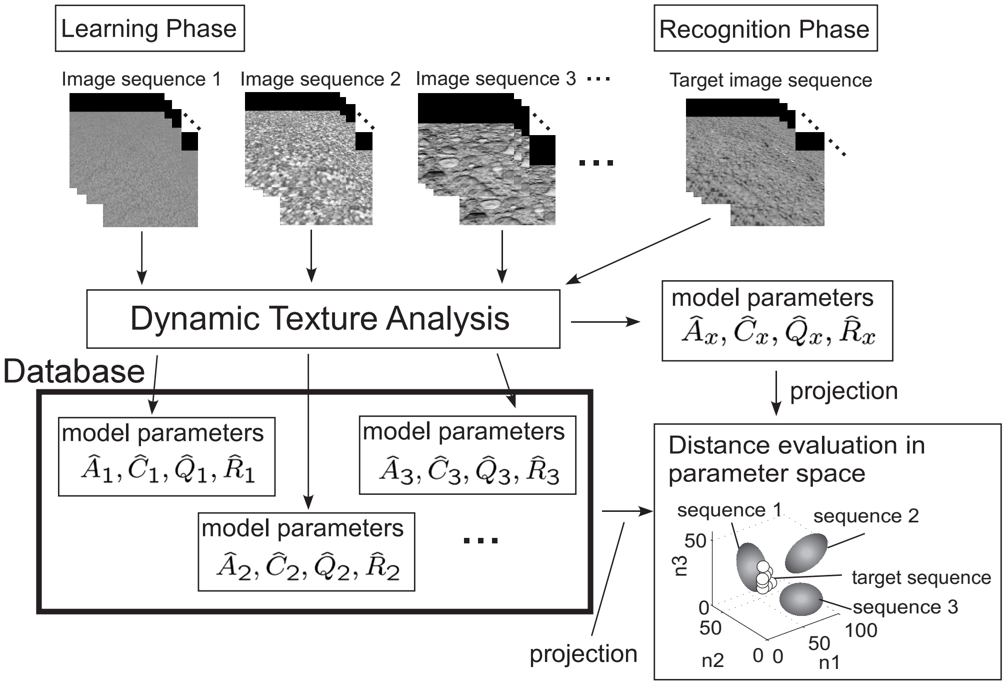

Learning Phase
Recognition Phase
Image sequence 1
Image sequence 2
Image sequence 3 ⋯
Target image sequence
Dynamic Texture Analysis
model parameters $\hat{A}_x, \hat{C}_x, \hat{Q}_x, \hat{R}_x$
projection
Database
model parameters $\hat{A}_1, \hat{C}_1, \hat{Q}_1, \hat{R}_1$
model parameters $\hat{A}_3, \hat{C}_3, \hat{Q}_3, \hat{R}_3$
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. Overview of terrain classification scheme utilizing Dynamic Texture.

shceme is divided into two phases called 'Learning Phase' and 'Recognition Phase'. Each phase is briefly described as follows.

(Learning Phase)

1. Acquire video sequences for various types of terrains (e.g. fine regolith, sand, gravel, etc.) taken from view points in the vehicle's steady-state motion.
2. Estimate the parameters of the Dynamic Texture model.
3. Construct a database of the estimated parameter sets for all the different types of terrain sequences.

(Recognition Phase)

1. Acquire a target image sequence.
2. Estimate the parameters of the Dynamic Texture model.
3. Compute the distances between the dynamical system model for the target sequence and the ones registered in the Learning Phase.
4. Classifying the target image sequence as the one closest to the terrain types in the database.

III. 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 these parameters can be estimated using a system identification theory such as N4SID,[7] the computational load tends to be large for video sequences which contain substantial data. Previous work on Dynamic Textures[5] proposed to apply a suboptimal estimation algorithm utilizing Principal Component Analysis (PCA-ID) in order to decrease the dimensionality of the state-space model. However, with this PCA-ID algorithm, not only the accuracy issue on the dynamical model still remains, but also computational load could be serious for relatively large size of the image frames due to the algorithm of PCA.

In this work, instead of the conventional PCA-ID algorithm, a new algorithm based on the components of 2-Dimensional Discrete Cosine Transform (2D- DCT) and a system identification algorithm, N4SID. The proposed method has an advantage in that optimal solution for the dynamical model is obtained within less computational time.

The proposed method contains two steps as follows:

STEP1: Original $M \times N$ pixel 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 vectors or matrices for the low-dimensional 2-D DCT components.

STEP2: N4SID algorithm[7] is applied to $y_c(k),\ k = 1, 2, \cdots, K$ in **STEP1**, and the linear dynamical system paraters such as $A_c,\ C_c$ are computed for given order of the system n.

IV. Recognition of Dynamic Texture Model

Since the estimated models for the linear dynamical system are characterized by the parameter matrices 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 order to recognize different Dynamic Texture models, the follwoing three typical metrics can be introduced for measuring the distances between the dynamical models in parameter space.

1. **Euclidean distance:** A simple but appropriate metric to represent variation of dynamical property is adopted. For the observability matrix of Eq. (5), a Eucledean distance between the models M_1 and M_2 is defined as

$$d_E(M_1, M_2) = \sqrt{\sum_{i=1}^{n} (\sigma_i(M_1) - \sigma_i(M_2))^2}, \tag{6}$$

where $\sigma_i(M_1)$ and $\sigma_i(M_2)$ are the i-th order sigular values of $\mathcal{O}_n(M_1)$ and $\mathcal{O}_n(M_2)$, respectively.

2. **Martin's distance:** A distance metric defined by Martin[8] is introduced. This metric is derived for a linear dynamical system model (ARMA model) equivalent to Eq. (1), and relation of the metric and the subspace angles between two ARMA models is given by De Cock and De Moor.[9] If p principal angles $\theta_k \in [0, \pi/2]$ between the ranges of the matrices A and B are recursively defined for $k = 1, 2, \cdots, n$ as

$$\begin{aligned} \cos\theta_1 &= \max_{x\in\mathbb{R}^p,\ y\in\mathbb{R}^q} \frac{|x^T A^T By|}{||Ax||_2||By||_2} = \frac{|x_1^T A^T By_1|}{||Ax_1||_2||By_1||_2}, \\ \cos\theta_k &= \max_{x\in\mathbb{R}^p,\ y\in\mathbb{R}^q} \frac{|x^T A^T By|}{||Ax||_2||By||_2} = \frac{|x_k^T A^T By_k|}{||Ax_k||_2||By_k||_2} \text{ for } k = 2, \cdots, q, \end{aligned} \tag{7}$$

subject to $x_i^T A^T Ax = 0$ and $y_i^T B^T By = 0$ for $i = 1, 2, \cdots, k-1$, the Martin's distance is derived as follows:

$$d_M(M_1,\ M_2) = \sqrt{\ln \prod_{i=1}^{n} \frac{1}{\cos^2\theta_i}}. \tag{8}$$

3. **Kernel density function (KDF) on Stiefel manifold:** While the above two metrics are derived to directly measure the distance between two subspaces spanned by the column vectors of the observability matrices, distance measures on special manifold such as Stiefel and Grassmann manifolds, on which the parameters of the dynamical system model lie have also been proposed.[6] In the previous work, a metric using a kernel density function based on a Procrustes representation for the distance metric is introduced. The Stiefel manifold $V_{n,m}$ is a space whose points are n-frames in $\mathbb{R}^m$, and each point on the manifold can be represented as a $m \times n$ matrices X such that $X^T X = I_n$, where I_n is $n \times n$ identity matrix.

   By singular value decomposition of $\mathcal{O}_n(M)$ such that

$$\mathcal{O}_n(M) = U\Sigma V^*, \tag{9}$$

the orthonormal matrix $U \in \mathbb{R}^{m\times n}$ is regarded as a point on the Stiefel manifold retaining the column space property of the original observabiliry matrix, and the class conditional density can be estimated by using a following function:

$$\hat{f}(M_1,\ M_2) = \frac{1}{n} C(P_s) \sum_{i=1}^{n} K(P_s^{-1/2}(I_n - X_{2,i}^T X_1 X_1^T X_{2,i})P_s^{-1/2}), \tag{10}$$

where X_1 is a matrix on the Stiefel manifold constructed by the model M_1, and $X_{2,i}$ $(i = 1, \cdots, n)$ are the sample matrices on the Stiefel manifold from a class of the model M_2. $K(A)$ is the kernel function for a matrix A, P_s is $n \times n$ positive definite matrix as a smoothing parameter, and $C(P_s)$ is a normalizing factor selected so that the estimated kernel density integrate to unity.

   In this paper, X_1 and $X_{2,i}$ correspond to U which are derived from the above singular value decomposition, and the exponential kernel $K(A) = \exp(-\mathrm{tr}(A))$ is treated. $\hat{f}(M_1,\ M_2)$ has the values between 0 and 1, and it is larger if the structure of the manifold of M_1 is close to the one of M_2. Therefore, in this study, the following function is defined as a distance metric:

$$d_K(M_1,\ M_2) = 1.0 - \hat{f}(M_1,\ M_2) \tag{11}$$

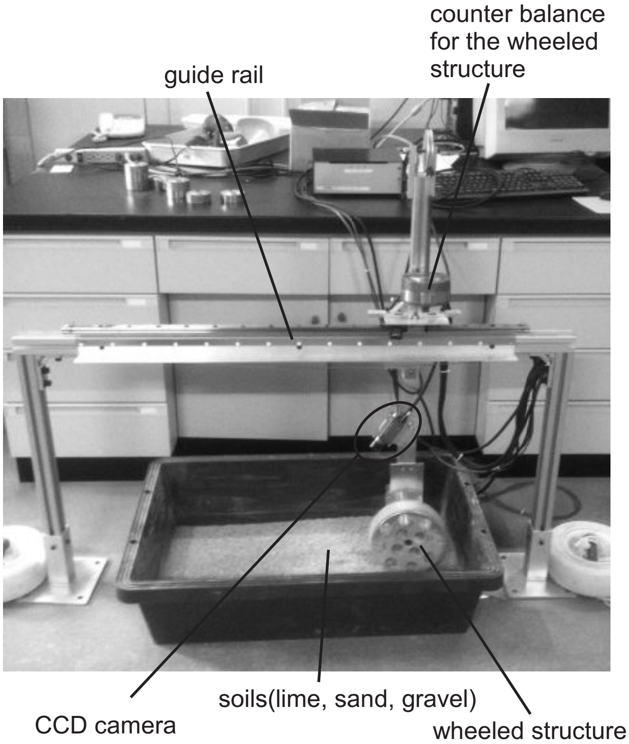


Figure 2. Testbed for acquiring terrain image sequences.

V. Recognition Test for Real Image Sequences

In order to validate the effectiveness of the proposed methods, an experiment is conducted by using a testbed as shown in Fig. 2. Real image sequences for four types of the terrain textures (magnesium lime, fine and coarse sand, and gravel) are obtained using a CCD camera (SONY XCD-V60CR). On this testbed, translational motions are given to the camera which is fixed on a wheeled structure. The wheel is driven by constant torque with a brushless DC motor, which averagely gives constant velocity to the CCD camera on flat surface.

Figure 3 shows real image sequences applied to the proposed methods. Each terrain sequence captures different soil particles identically-distributed in the image frame. In order to see discriminative ability not only for terrain textures but also for rover translational motion, three constant torques, T_0, $2T_0$, and $3T_0$ (T_0:reference torque) are given to the DC motor for each type of terrain texture, which generate different image velocity fields in the image frames, v_1, v_2, and v_3. The table of combination for all the experimental parameters is shown in Table 1. As shown in Fig. 3, 20 local block images of 8×8, 16×16, and 32×32 pixels are cropped from the original images, and the block image sequences consisting of 300 frames (for about 10sec) are applied to the proposed scheme.

Considering sufficient accuracy for estimating the Dynamic Texture model by using N4SID or PCA-ID algorithm, the dimension of the finite obserbavility matrix, n in Eq. (5) is fixed at 10. Also, the number of the 2D-DCT components in Eq. (3) is fixed such that $M_c = N_c = 8$.

Recognition rate is evaluated through 2-fold cross validation test, that is, while half of the block image sequences are applied for the learning phase, the rest of the target sequences are for the recognition phase. The same process is repeated after exchanging the block image sequences for each phase. Note here that the block image sequences for the both phases are selected so that they never overlap with each other in the spatiotemporal domain as shown in Fig. 3.

In this study, each terrain image sequence is recognized using a threshold value of the distance metrics, which is determined by the maximum distance among the same image sequences in the learning phase. Aiming at seeing sensitivity to the threshould values, a receiver operating characteristic (ROC)[10] analysis is conducted at the same time. In signal detection problem, if the relation between predicted result and actual

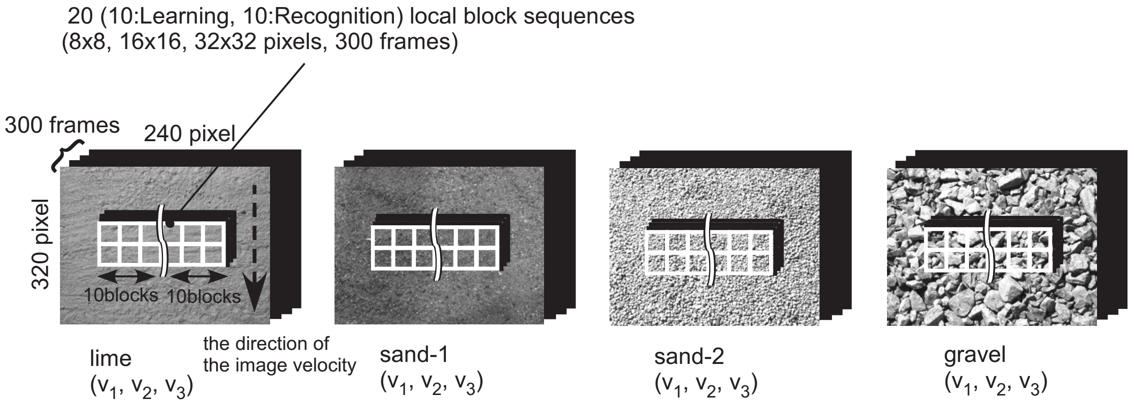

20 (10:Learning, 10:Recognition) local block sequences
(8x8, 16x16, 32x32 pixels, 300 frames)
300 frames
240 pixel
320 pixel
10blocks
10blocks
the direction of the image velocity
lime (v_1, v_2, v_3)
sand-1 (v_1, v_2, v_3)
sand-2 (v_1, v_2, v_3)
gravel (v_1, v_2, v_3)

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

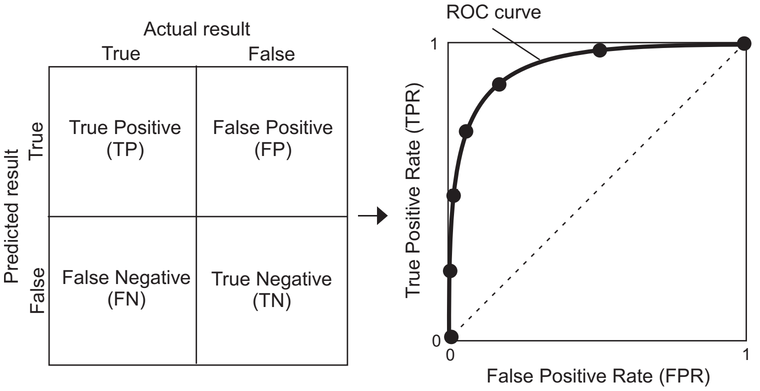


Figure 4. Receiver Operating Characteristic (ROC).

result for a discrimination threshold is shown as a cross tabulation in Fig.4, two types of the evaluation metrics, true positive rate and false positive rate are derived as follows:

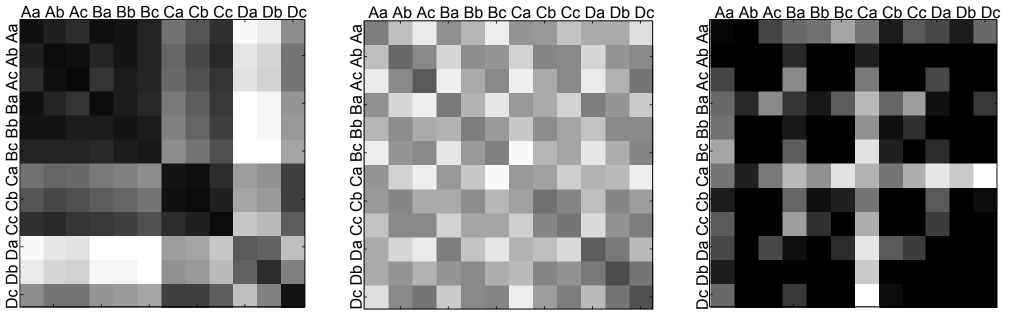
$$\mathrm{TPR} = N_{\mathrm{TP}}/(N_{\mathrm{TP}} + N_{\mathrm{FN}}), \quad \mathrm{FPR} = N_{\mathrm{FP}}/(N_{\mathrm{FP}} + N_{\mathrm{TN}}), \tag{12}$$

where TPR and FPR mean the true positive rate and the false positive rate, respectively. Also, N_{TP}, N_{FN}, N_{FP}, and N_{TN} mean the numbers of the true positive, the false negative, the false positive, and the true negative, respectively. For these two operating characteristics (TPR and FPR) computed from various threshold values, ROC curve is plotted as shown in the right-hand side of Fig. 4.

Some results of the cross validation test are shown in the following figures and tables. Figures 5 and 6 show correlation maps in which each grayscale block image implies the correlation between learned sequences (aligned in column) and target sequences (aligned in row). The grayscale level is computed from the mean values of the distances among twenty block image sequences, such that darker image shows shorter distance. To increase visibility of the block images, the grayscale levels are normalized for each map. The results

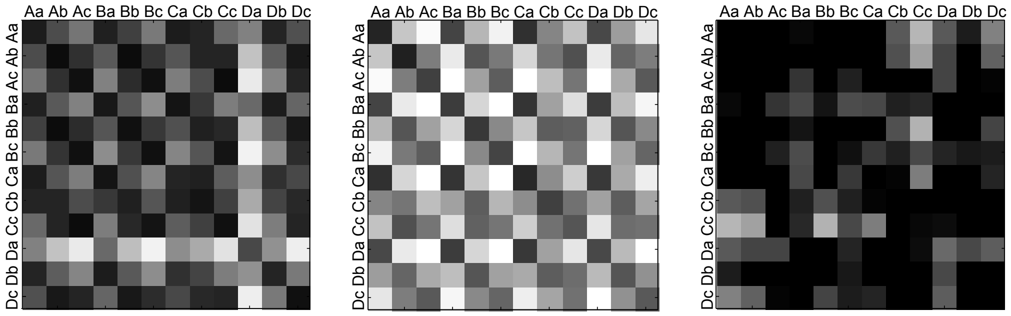
Table 1. Table of combination

		Terrain type			
		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



(a) Euclidean distance (b) Martin's distance (c) KDF on the Stiefel Manifold

Figure 5. Correlation map for the 2-fold cross validation (2D-DCT+N4SID, 8×8pixel block seqeunces, the 1st test).



(a) Euclidean distance (b) Martin's distance (c) KDF on the Stiefel Manifold

Figure 6. Correlation map for the 2-fold cross validation (PCA-ID, 8×8pixel block seqeunces, the 1st test).

of the PCA-ID algorithm are also shown to compare the proposed 2D-DCT+N4SID algorithm with the conventional one.

As shown in these figures, all the distance metrics show stronger correlations for the same terrain textures labeled as A, B, C, and D, appearing as the darker diagonal lines from the left upper corner to the right lower corner on the maps.

Tables 2 to 7 show results of recognition rates, which are computed as the true positive rate in Eq. (12). In this study, two classes of recognizing features, dynamic texture and static texture are defined. While the static texture is categorized only according to terrain type (A, B, C and D, which correspond to each column in Table 1), the dynamic texture is categorized according to image velocity as well as to terrain type (Aa, Ab, $\cdots$, Dc, which correspond to each matrix element in Table 1). The results of the recognition rates for the conventional PCA-ID algorithm are also shown for comparison.

As shown in the tables, for the Euclidiean and the Martin's distances, the true positive rates are relatively high over 86.5% on average for the both feature classes. On the other hand, the KDF on the Stiefel manifold shows lower rates especially for the dynamic texture class, which results in at most 70.8% on average. According to these experimental results, the effect of the block image size is not clearly seen except for the KDF on the Stiefel manifold, which shows lower rates with respect to larger block image sizes. One of the issues of the KDF on the Stiefel manifold is consdidered that this metric needs enough sample matrices on the Stiefel manifold from the same image sequences to compute the kernel density function, which may not satisfied for relatively large size of the image sequences.

Some ROC curves plotted for the experimental results are shown as Figs. 7 to 12. They are plotted for 30 threshold values equally sampled between the maximum and the minimum values for the learned sequences. For all the results, while the plots close to the lower left corner ($(\mathrm{FPR}, \mathrm{TPR}) = (0,0)$) show the ones for the minimum threshold values, the plots on upper right portions show the ones for the maximum threshold values.

In ROC space, since false positive rate is usually inclease with respect to true positive rate, ROC curve is depicted as the one from the lower left corner ($(\mathrm{FPR}, \mathrm{TPR}) = (0,0)$) to the upper right corner ($(\mathrm{FPR}, \mathrm{TPR}) =$

$(1,1)$). As for the desirable classifier (distance metric), true positive rate should be enough high even if false positive rate is low, which implies a curve line skewed to upper left corner of the frame as shown in Fig. 4.

From the viewpoint of the above characteristic of the ROC space, for the proposed 2D-DCT+N4SID algorithm, the Euclidean distance is more appropriate as a distance metric.

It should be discussed in future work which metric is more appropriate to discriminate more various types of the terrain texures and dynamical properties caused by rover motion, considering the validity of the model estimation algorithms.

Table 2. True positive rates for CASE 1 (2D-DCT+N4SID, 8×8 pixel-block sequences).

dynamic texture				**static texture**			
	1st	**2nd**	**mean**		**1st**	**2nd**	**mean**
Euclidean dist.	92.0%	83.5%	87.8%	**Euclidean dist.**	94.5%	98.7%	96.6%
Martin's dist.	94.3%	98.5%	96.4%	**Martin's dist.**	99.8%	99.6%	99.7%
KDF on the Stiefel manifold	19.2%	27.5%	23.3%	**KDF on the Stiefel manifold**	90.3%	93.3%	91.8%

Table 3. True positive rates for CASE 2 (2D-DCT+N4SID, 16×16 pixel-block sequences).

dynamic texture				**static texture**			
	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%

Table 4. True positive rates for CASE 3 (2D-DCT+N4SID, 32×32 pixel-block sequences).

dynamic texture				**static texture**			
	1st	**2nd**	**mean**		**1st**	**2nd**	**mean**
Euclidean dist.	85.8%	88.8%	87.3%	**Euclidean dist.**	97.8%	100.0%	98.9%
Martin's dist.	96.9%	98.8%	97.8%	**Martin's dist.**	99.6%	99.9%	99.7%
KDF on the Stiefel manifold	5.8%	10.0%	7.9%	**KDF on the Stiefel manifold**	85.8%	84.7%	85.3%

Table 5. True positive rates for CASE 1 (PCA-ID, 8×8 pixel-block sequences).

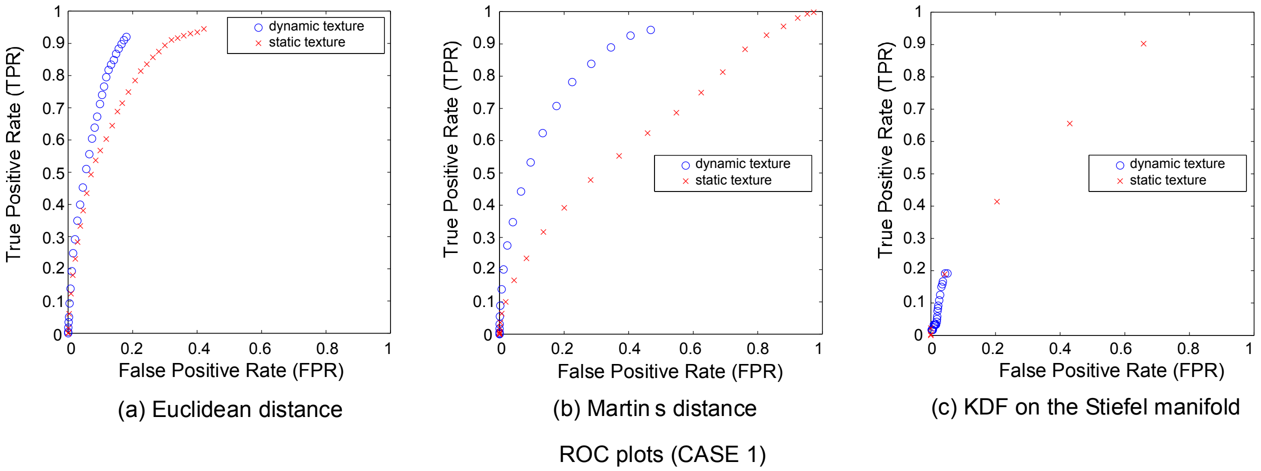
dynamic texture				static texture			
	1st	2nd	mean		1st	2nd	mean
Euclidean dist.	88.3%	91.5%	89.9%	Euclidean dist.	95.9%	99.7%	97.8%
Martin's dist.	95.9%	96.8%	96.3%	Martin's dist.	99.7%	99.8%	99.8%
KDF on the Stiefel manifold	60.8%	80.8%	70.8%	KDF on the Stiefel manifold	100.0%	95.3%	97.6%

Table 6. True positive rates for CASE 2 (PCA-ID, 16×16 pixel-block sequences).

dynamic texture				static texture			
	1st	2nd	mean		1st	2nd	mean
Euclidean dist.	93.0%	96.1%	94.5%	Euclidean dist.	99.3%	99.7%	99.5%
Martin's dist.	94.0%	93.1%	93.5%	Martin's dist.	99.9%	99.8%	99.9%
KDF on the Stiefel manifold	8.3%	4.2%	6.3%	KDF on the Stiefel manifold	93.6%	91.1%	92.4%

Table 7. True positive rates for CASE 3 (PCA-ID, 32×32 pixel-block sequences).

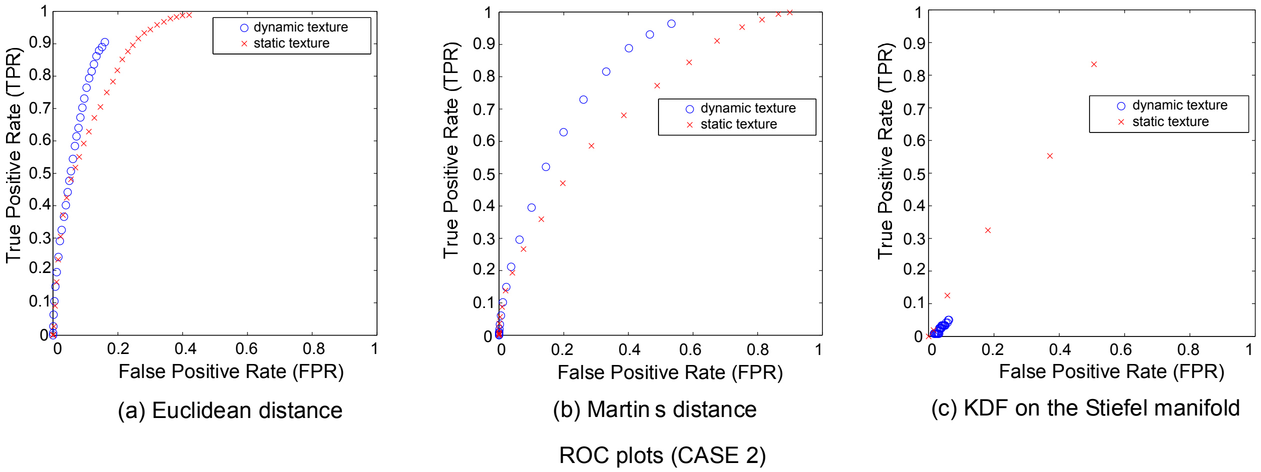
dynamic texture				static texture			
	1st	2nd	mean		1st	2nd	mean
Euclidean dist.	91.1%	89.8%	90.4%	Euclidean dist.	99.5%	99.6%	99.5%
Martin's dist.	87.3%	94.9%	91.1%	Martin's dist.	99.1%	99.4%	99.3%
KDF on the Stiefel manifold	0.0%	0.8%	0.4%	KDF on the Stiefel manifold	80.0%	83.6%	81.8%



(a) Euclidean distance (b) Martin s distance (c) KDF on the Stiefel manifold

ROC plots (CASE 1)

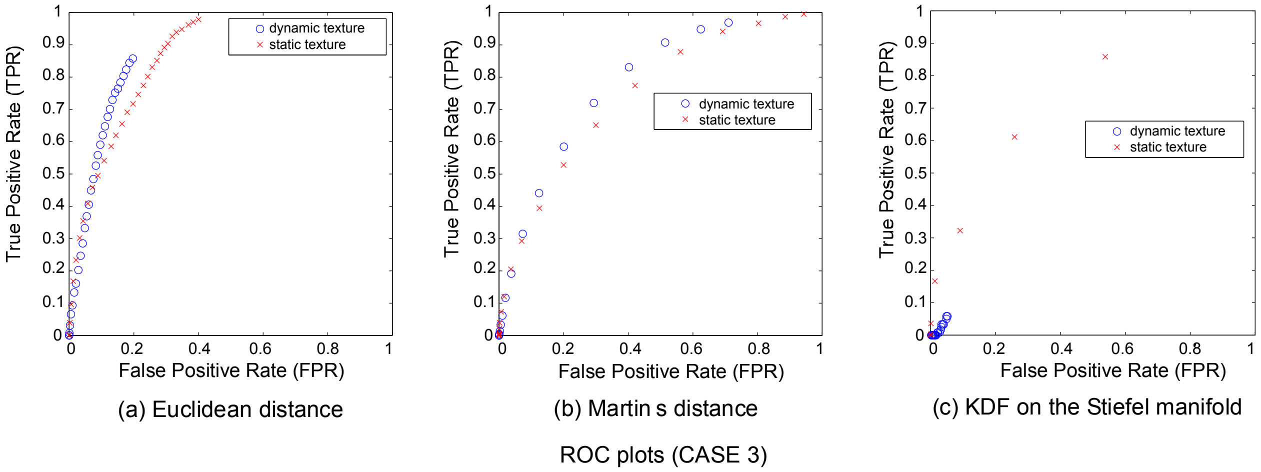
Figure 7. ROC plots for CASE 1 (2D-DCT+N4SID, 8×8 pixel-block sequences).



(a) Euclidean distance (b) Martin s distance (c) KDF on the Stiefel manifold

ROC plots (CASE 2)

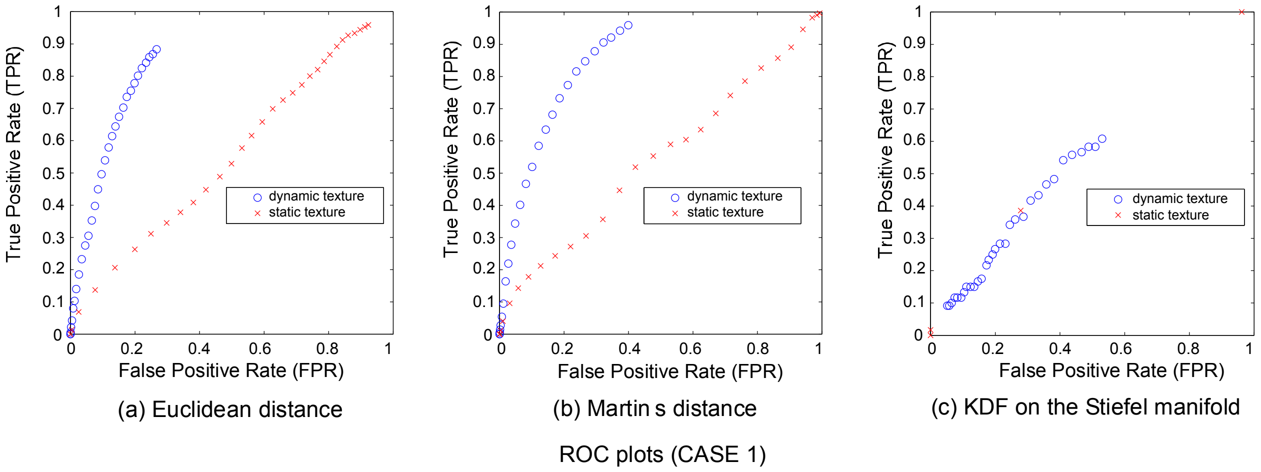
Figure 8. ROC plots for CASE 2 (2D-DCT+N4SID, 16×16 pixel-block sequences).



(a) Euclidean distance (b) Martin s distance (c) KDF on the Stiefel manifold

ROC plots (CASE 3)

Figure 9. ROC plots for CASE 3 (2D-DCT+N4SID, 32×32 pixel-block sequences).



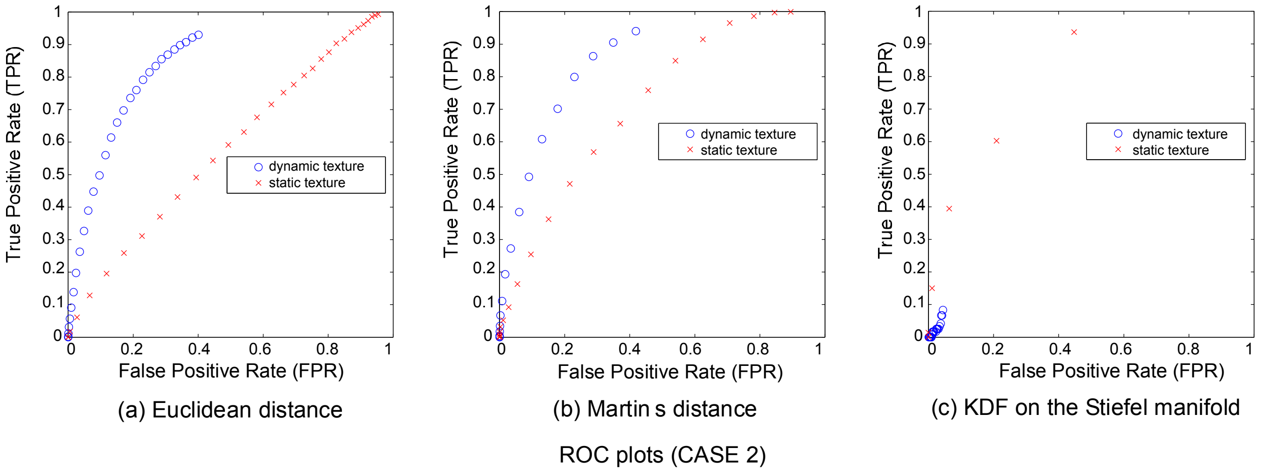
(a) Euclidean distance

(b) Martin s distance

(c) KDF on the Stiefel manifold

ROC plots (CASE 1)

Figure 10. ROC plots for CASE 1 (PCA-ID, 8×8 pixel-block sequences).



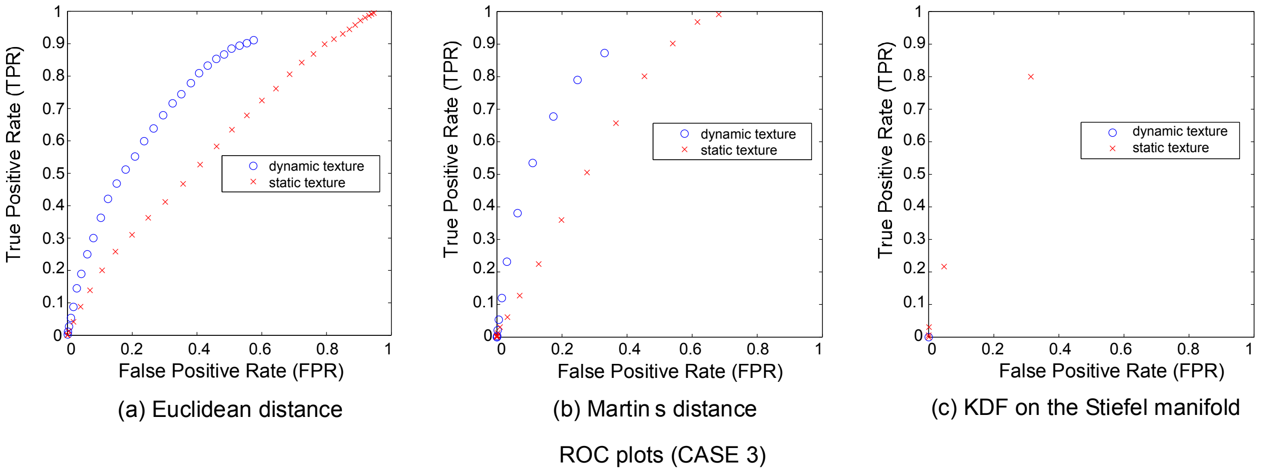
(a) Euclidean distance

(b) Martin s distance

(c) KDF on the Stiefel manifold

ROC plots (CASE 2)

Figure 11. ROC plots for CASE 2 (PCA-ID, 16×16 pixel-block sequences).



(a) Euclidean distance

(b) Martin s distance

(c) KDF on the Stiefel manifold

ROC plots (CASE 3)

Figure 12. ROC plots for CASE 3 (PCA-ID, 32×32 pixel-block sequences).

VI. Conclusion

This paper proposed a novel terrain classification method for planetary rover utilizing Dynamic Texture. The recognition rates computed from several distance measures for the estimated Dynamic Texture models were evaluated through the experiments using a testbed. According to the experimental results, some distance metrics show relatively high true positive rates to discriminate not only terrain textures but also rover translational motion. The proposed model estimation algorithm also shows desirable characteristic in the ROC space for a distance metric.

In future works, ditance metric suitable to distinguish various types of terrain textures as well as rovers dynamical properties such as translational velocity, slippage, and sinkage is going to be discussed in detail. At the same time, the validity of the model estimation algorithms based on a linear dynamical system model is further evaluated.

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, pp.391-410, 2009.

[2] Halatci, I., Brooks, C.A., and Iagnemma, K., “Terrain Classification and Classifier Fusion for Planetary Exploration Rovers,” Proc. of the IEEE Aerospace Conference, pp.1-11, 2007.

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

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

[5] 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, pp.58-63 ,2001.

[6] Turaga, P., Veeraraghavan, A., and Chellappa, R., “Statistical Analysis on Stiefel and Grassmann Manifolds with Applications in Computer Vision,” Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’08), pp.1-8, 2008.

[7] Overschee, P.V. and Moor, B.D., “N4SID: Subspace Algorithms for the Identification of Combined Deterministic-Stochastic Systems,” Automatica, Vol.30, Issue 1, pp.75-93, 1994.

[8] Martin, R.J., “A Metric for ARMA Processes,” IEEE Trans. on Signal Processing, Vol.48, No.4, pp.1164-1170, 2000.

[9] De Cock, K. and De Moor B., “Subspace Angles Between Linear Stochastic Models,” Proc. of the 39th IEEE Conference on Decision and Control (CDC2000), Vol.2, pp.1561-1566, 2000.

[10] Witten, I.H., Frank, E., and Hall, M.A., Data Mining: Practical Machine Learning Tools and Techniques 3rd ed., Morgan Kaufmann, Burlington, MA, 2011.