10

Autonomous Terrain Classification for Planetary Rover

Koki Fujita
Kyushu University
Japan

1. Introduction

In order to improve autonomous mobility of planetary rover, many works have recently focused on non-geometric features of surrounding terrain such as color, texture, and wheel-soil interaction mechanics (Dima et al., 2004; Halatci et al., 2007; Helmick et al., 2009; Ishigami et al., 2007). To tackle with the issue, most of them propose to utilize on-board sensors such as multi-spectral imagers, CCD cameras, laser range sensor, and accelerometer.

This study aims at classifying textures and physical properties 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 discretly utilize image data, this work proposes to utilize whole of the motion image sequence taking terrain surface from rover on-board camera. Unlike the conventional techniques to classify terrain surfaces based on single or stereo camera images, the proposed method improves discrimination ability for visual salience 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 camera, and homogeneous and isotropic properties of terrain texture, motion image sequence can be reduced to a set of parameters of the Dynamic Texture model (Saisan et al., 2001). The estimated parameters contain unique properties not only with visual salience in terrain surface but also with dynamics in camera (or vehicle) motion and terra-mechanics associated with 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 translational motions of camera 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 work.

2. 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.

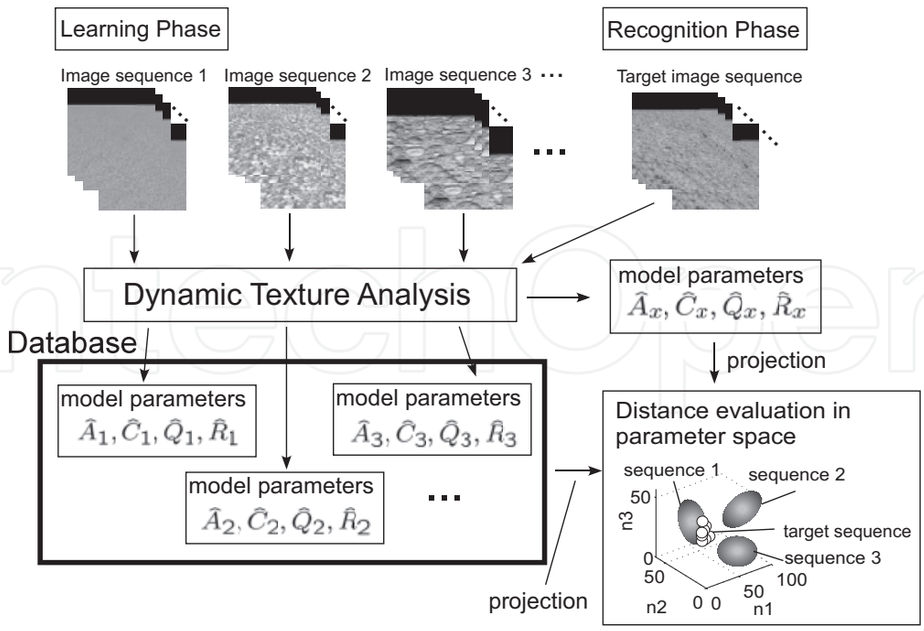


Fig. 1. Overview of terrain classification scheme utilizing Dynamic Texture.

As shown in the figure, the 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 database.
4. Classifying the target image sequence as the one closest to the terrain types in the database.

3. 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:

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

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 white Gaussian random vectors. 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 (Overschee & Moor, 1994), the computational load tends to be large for video sequences which contain substantial data. Previous work on Dynamic Textures (Saisan et al., 2001) proposed to apply a suboptimal estimation algorithm utilizing a 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 estimated dynamical system 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 a 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

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

where the subscript c denotes the vectors or the matrices for the low-dimensional 2-D DCT components.

STEP2: N4SID algorithm (Overschee & Moor, 1994) 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.

4. Recognition of dynamic texture model

Since the linear dynamical system models as shown in Eq. (1) are characterized by the parameter matrices A and C, they can be identified with column space of the extended observability matrix:

$$\mathcal{O}_\infty(M) = \begin{bmatrix} C^T & (CA)^T & (CA^2)^T & \cdots \end{bmatrix}^T. \tag{4}$$

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

$$\mathcal{O}_n(M) = \begin{bmatrix} C^T & (CA)^T & (CA^2)^T & \cdots (CA^{n-1})^T \end{bmatrix}^T. \tag{5}$$

In order to recognize different Dynamic Texture models, the follwoing three typical metrics can be introduced for measuring distances between the dynamical models in parameter space.

1. **Euclidean distance:** For the observability matrix of Eq. (5), a distance metric for models M_1 and M_2 can be defined as a simple but appropriate one to represent a difference in dynamical property as follows:

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

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

2. **Martin's distance:** Martin's distance (Martin, 2000) is a distance metric derived for a linear dynamical system model, ARMA model, which is equivalent to Eq. (1). It can also be applied to distinguish different Dynamic Texture models. 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

$$\begin{aligned} \cos\theta_1 &= \max_{x \in \mathbb{R}^p,\, y \in \mathbb{R}^q} \frac{|x^T A^T B y|}{||Ax||_2 ||By||_2} = \frac{|x_1^T A^T B y_1|}{||Ax_1||_2 ||By_1||_2}, \\ \cos\theta_k &= \max_{x \in \mathbb{R}^p,\, y \in \mathbb{R}^q} \frac{|x^T A^T B y|}{||Ax||_2 ||By||_2} = \frac{|x_k^T A^T B y_k|}{||Ax_k||_2 ||By_k||_2} \text{ for } k = 2, \cdots, q, \end{aligned} \tag{7}$$

   subject to $x_i^T A^T A x = 0$ and $y_i^T B^T B y = 0$ for $i = 1, 2, \cdots, k-1$, the Martin's distance is derived as follows (De Cock & De Moor, 2000):

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

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 metrics on special manifold such as Stiefel and Grassmann manifolds, on which the parameters of the dynamical system model lie have also been proposed (Turaga et al., 2008). 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 $m \times n$ matrix 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 $U \in \mathbb{R}^{m\times n}$ $(U^T U = I_n)$ is regarded as a point on the Stiefel manifold retaining the column space property of the original observabiliry matrix. Although the Stiefel manifold is endowed with a Riemannian structure and a geodesic computation for distances between points on the manifold is possible, the other distance metric called a "Procrutes distance" is introduced to estimate a class conditional probability density in an ambient Euclidean space.

The Procrustes distance is defined for two matrices X_1 and X_2 on $V_{k,m}$ as follows:

$$\begin{aligned} d_V^2(X_1, X_2) &= \min_{R>0} tr(X_1 - X_2 R)^T (X_1 - X_2 R) \\ &= \min_{R>0} tr(R^T R - 2X_1^T X_2 R + I_k). \end{aligned} \tag{10}$$

If R varies over the space $\mathbb{R}^{k\times k}$, the Procrustes distance is minimized at $R = X_1^T X_2$, so that $d_V^2(X_1, X_2)$ is equal to $tr(I_k - X_2^T X_1 X_1^T X_2)$. Also, the class conditional density for this Procrustes distance metric can be estimated by using the following function (Chikuse, 2003):

$$\hat{f}(X; P_s) = \frac{1}{n} C(P_s) \sum_{i=1}^{n} K(P_s^{-1/2}(I_n - X_{2,i}^T X_1 X_1^T X_{2,i}) P_s^{-1/2}), \tag{11}$$

where $X_{2,i}$ $(i = 1, \cdots, n)$ are the sample matrices on the Stiefel manifold from the same class of the model. $K(A)$ 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 the normalizing factor selected so that the estimated kernel density integrate to unity.

In this paper, X_1 and $X_{2,i}$ are the matrices on the Stiefel manifold constructed by the models M_1 and M_2, respectively, and these matrices correspond to U derived from the singular value decomposition of the observability matrix $\mathcal{O}_n(M)$. As a kernel function to compute $\hat{f}(X; P_s)$, the exponential kernel $K(A) = \exp(-\mathrm{tr}(A))$ is treated. Since the output of $\hat{f}(X; P_s)$ ranges between 0 and 1 and increases inversely with the distance between two models, the following function is defined as an actual distance metric:

$$d_K(M_1, M_2) = 1.0 - \hat{f}(X; P_s) \tag{12}$$

5. Recognition test for real image sequences

In order to validate the effectiveness of the proposed methods, an experiment was 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) were obtained using a CCD camera (SONY XCD-V60CR). On this testbed, translational motions are given to the camera fixed on a wheeled structure. The wheel is driven by constant torque from a brushless DC motor, which gives averagely constant velocity to the CCD camera on flat surface. The experimental environment is shown in Table 1.

Real image sequences as shown in Fig. 3 were applied to the proposed methods. Each terrain sequence depicts different soil particles identically-distributed in the image frames. In order to

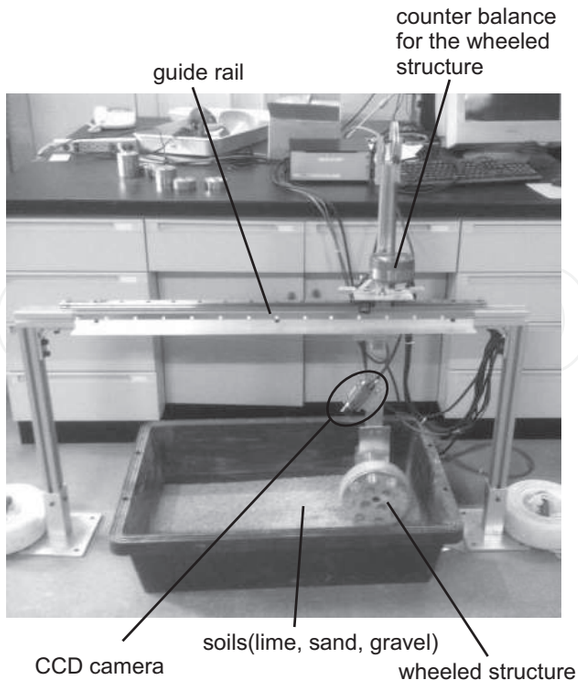


Fig. 2. Testbed for acquiring terrain image sequences.

Specification of CCD camera	Focal length: 8 mm Field of view: 34.0 × 25.6 deg CCD image resolution (original): 640 × 480 pixels Frame rate: 30 fps
Height of the camera	330 mm
Wheel diameter	181.7 mm
Velocity of the camera (mean value)	V_1: 17.4 mm/sec, V_2: 35.0 mm/sec, V_3: 53.5 mm/sec

Table 1. Experimental environment

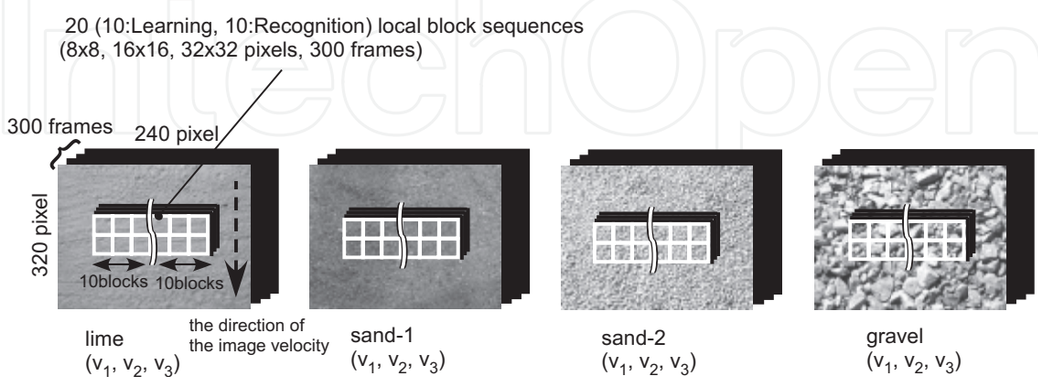


Fig. 3. Real image sequences applied to the proposed methods.

		Terrain type			
		lime (A)	**sand-1 (B)**	**sand-2 (C)**	**gravel (D)**
Image velocity	v_1 **(a)**	Aa	Ba	Ca	Da
	v_2 **(b)**	Ab	Bb	Cb	Db
	v_3 **(c)**	Ac	Bc	Cc	Dc

Table 2. Table of combination

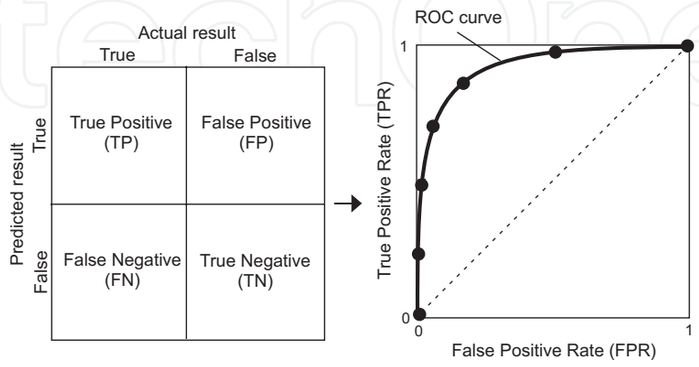


Fig. 4. A receiver operating characteristic (ROC) analysis.

see discriminative ability not only for terrain textures but also for rover translational motion, three constant torques were given to the DC motor for each type of terrain texture, which generated different velocities, V_1 to V_3 (as shown in Table 1). These velocities of the camera resulted in the different image velocity fields named as v_1, v_2, and v_3. The combination for all the experimental parameters with the terrain textures and the image velocities is shown in Table 2. As shown in Fig. 3, 20 local block images of 8×8 (CASE 1), 16×16 (CASE 2), and 32×32 (CASE 3) pixels were cropped from the original images, and the block image sequences consisting of 300 frames (for about 10sec) were applied to the proposed scheme.

In this study, considering sufficient accuracy for estimating the Dynamic Texture model by using N4SID or PCA-ID algorithm, the dimension of the finite observability matrix, n in Eq. (5) was fixed at 10. Also, the number of the 2D-DCT components in Eq. (3) was fixed such that $M_c = N_c = 8$ (i.e. $m_c = M_c \times N_c = 64$). As for the smoothing parameter in the KDF on the Stiefel manifold defined as Eq. (11), it was set such that $P_s = 100$ after trying to apply several values.

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

The terrain image sequences were recognized using threshold values of each distance metric, which were coincident with the maximum distances among the same image sequences in the Learning Phase. Aiming at seeing sensitivity to the threshold values, a receiver operating characteristic (ROC) (Witten et al., 2011) analysis was conducted at the same time.

If the relation between predicted result and actual result for a discrimination threshold is shown as a cross tabulation in Fig.4, two types of the evaluation metrics, true positive rate (TPR) and false positive rate (FPR) are derived as follows:

$$\mathrm{TPR} = N_{\mathrm{TP}}/(N_{\mathrm{TP}} + N_{\mathrm{FN}}), \quad \mathrm{FPR} = N_{\mathrm{FP}}/(N_{\mathrm{FP}} + N_{\mathrm{TN}}), \tag{13}$$

where 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.

For any classification problem, the false positive rate increases with the true positive rate. 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 Fig. 4.

Results of the cross validation test are shown in the following figures and tables. Figs. 5 to 10 show correlation maps in which each grayscale block image implies the correlation between the learned sequences (aligned in columns) and the target sequences (aligned in rows) for the twelve parameter combinations. The grayscale level is computed from the mean values of the distances among twenty block image sequences, such that the darker image shows the shorter distance. To increase visibility of the block images, the grayscale levels are normalized for each map. The results 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 clear correlations for the same terrain textures labeled as A, B, C, and D. They are appeared as the darker block images along the diagonal line from the upper left corner to the lower right corner on each map. While in most of the correlation maps, the differences between the correlation strengths for different terrain textures are not necessarily clear except for the Euclidean distance computed from the proposed 2D-DCT+N4SID algorithm, which only shows the gradual changes of the grayscale levels for different terrain textures.

Tables 3 to 8 show results of the true positive rates defined in Eq. (13). In this study, two classes of recognized features named “dynamic texture class” and “static texture class” are focused on. While the static texture class is categorized only according to terrain types (i.e. A, B, C and D in Table 2), the dynamic texture class is categorized according to image velocity as well as to terrain type (i.e. Aa, Ab, $\cdots$, Dc in Table 2). In the tables, the results obtained for these two categories are shown. 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% for the 2D-DCT+N4SID algorithm or 89.9% for the PCA-ID algorithm 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 23.3% for the 2D-DCT+N4SID algorithm or 70.8% for the PCA-ID algorithm, and significantly decreasing with the increasing block image sizes. As for the test results, the effect of the block image size is only seen in those for the KDF on the Stiefel manifold. One of the issues of the KDF on the Stiefel manifold is considered 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 be satisfied for relatively large size of the image sequences. Acoording

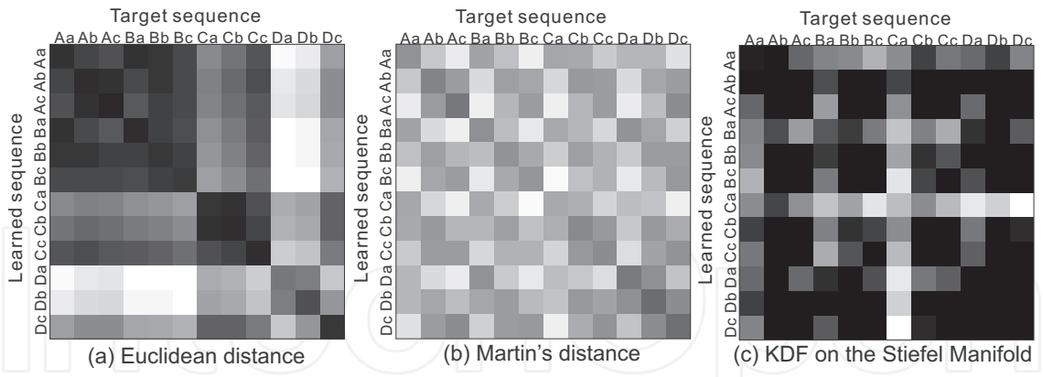


Fig. 5. Correlation map for CASE 1 (The proposed 2D-DCT+N4SID algorithm, 8×8 pixel-block seqeunces, the 1st test).

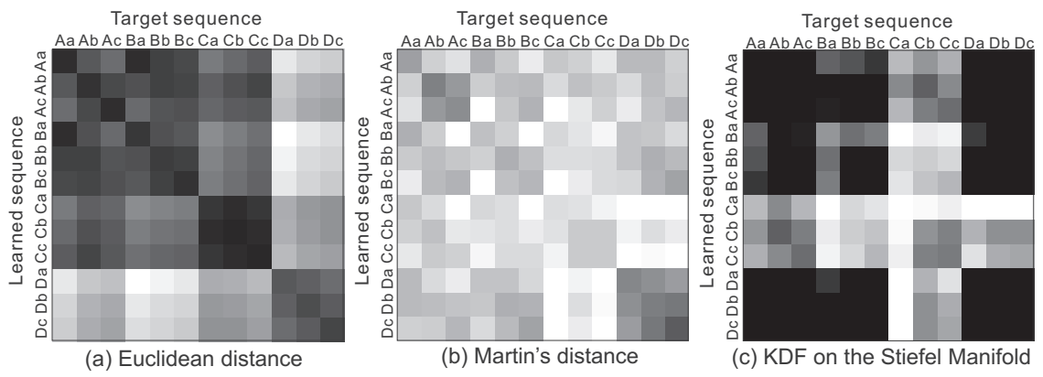


Fig. 6. Correlation map for CASE 2 (The proposed 2D-DCT+N4SID algorithm, 16×16 pixel-block seqeunces, the 1st test).

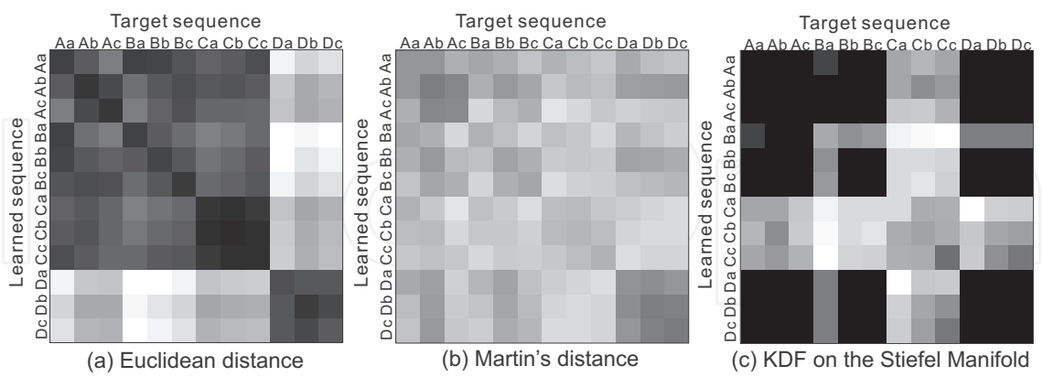


Fig. 7. Correlation map for CASE 3 (The proposed 2D-DCT+N4SID algorithm, 32×32 pixel-block seqeunces, the 1st test).

to the results of the true positive rates, although it seems that the conventional PCA-ID algorithm achieves better performance than the proposed algorithm does, a different view can be obtained from the following results of the ROC analysis.

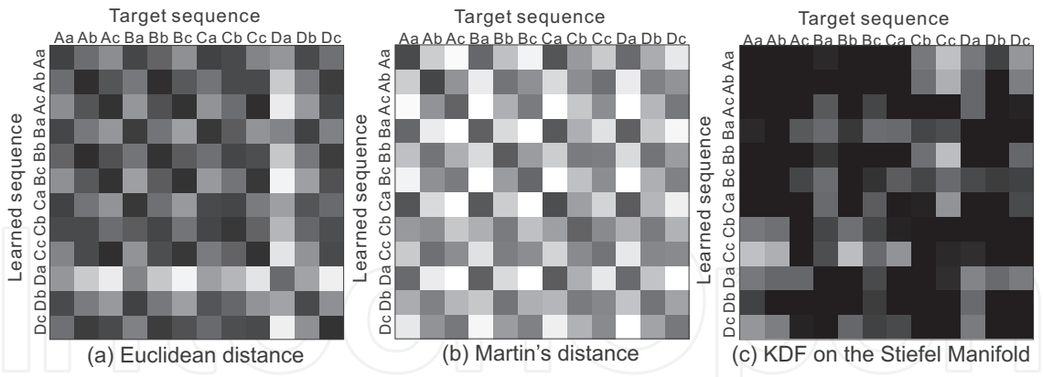


Fig. 8. Correlation map for CASE 1 (The conventional PCA-ID algorithm, 8×8 pixel-block seqeunces, the 1st test).

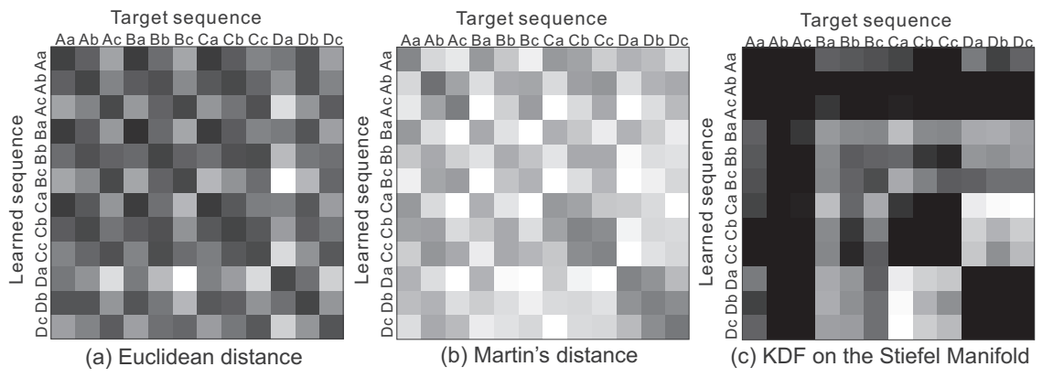


Fig. 9. Correlation map for CASE 2 (The conventional PCA-ID algorithm, 16×16 pixel-block seqeunces, the 1st test).

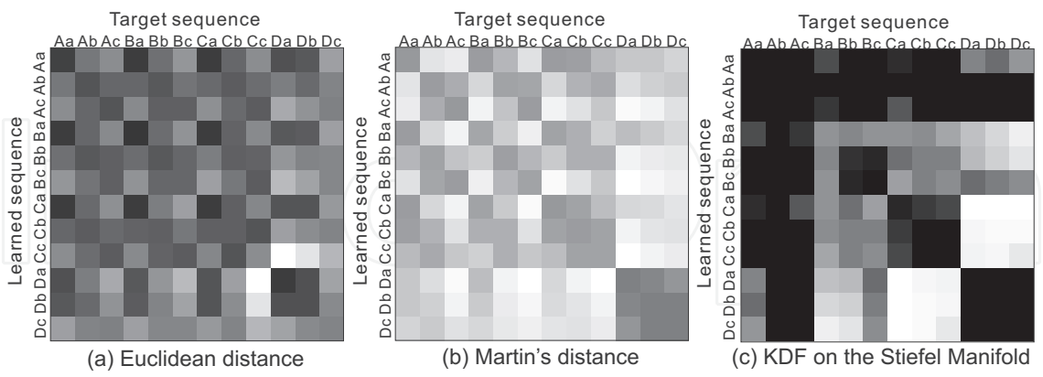


Fig. 10. Correlation map for CASE 3 (The conventional PCA-ID algorithm, 32×32 pixel-block seqeunces, the 1st test).

Some ROC curves for the same experimental results are shown in Figs. 11 to 16. 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

Dynamic texture class				Static texture class			
	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 1 (The proposed 2D-DCT+N4SID algorithm, 8×8 pixel-block sequences).

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%

Table 4. True positive rates for CASE 2 (The proposed 2D-DCT+N4SID algorithm, 16×16 pixel-block sequences).

Dynamic texture class				Static texture class			
	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 3 (The proposed 2D-DCT+N4SID algorithm, 32×32 pixel-block sequences).

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.

Although all the ROC plots start from the lower left corners, they don't necessarily reach to the upper right corners. Most of the plots for the KDF on the Stiefel manifold end in the middle of the ROC spaces, and especially for the dynamic texture class, they end up very low recognition rates, which is similarly seen in the previous tables of the true positive rate. On the other hand, the proposed 2D-DCT+N4SID algorithm depicts more desirable curves for the Euclidean distance, while the highest positive rates don't necessarily exceed those for the conventional PCA-ID algorithm.

Comparing between the two feature classes, the dynamic texture class and the static texture class, the recognition rates for the latter class show higher rates for a certain threshold values of each distance metric as shown in Tables 3 to 8. However, their recognition performances

Dynamic texture class				Static texture class			
	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 1 (The conventional PCA-ID algorithm, 8×8 pixel-block sequences).

Dynamic texture class				Static texture class			
	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 2 (The conventional PCA-ID algorithm, 16×16 pixel-block sequences).

Dynamic texture class				Static texture class			
	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%

Table 8. True positive rates for CASE 3 (The conventional PCA-ID algorithm, 32×32 pixel-block sequences).

in the ROC space are not necessarily better than for the dynamic texture class as depicted in Figs. 11 to 16. From these results, the three distance metrics introduced in this study are perceived as the ones intimately involved with the dynamical properties not only with the static visual salience.

The merit of the proposed algorithm is also seen in the computational time for the Dynamic Texture model learning. Table 9 shows the computational time for each process in the recognition test using a PC (CPU: Intel Core i7-640LM, RAM: 8MB). This result clearly shows that the conventional PCA-based algorithm becomes ineffective with the increasing size of the block image, and the computational time for the proposed algorithm does not depend on the size of the block images so much as the conventional algorithm does.

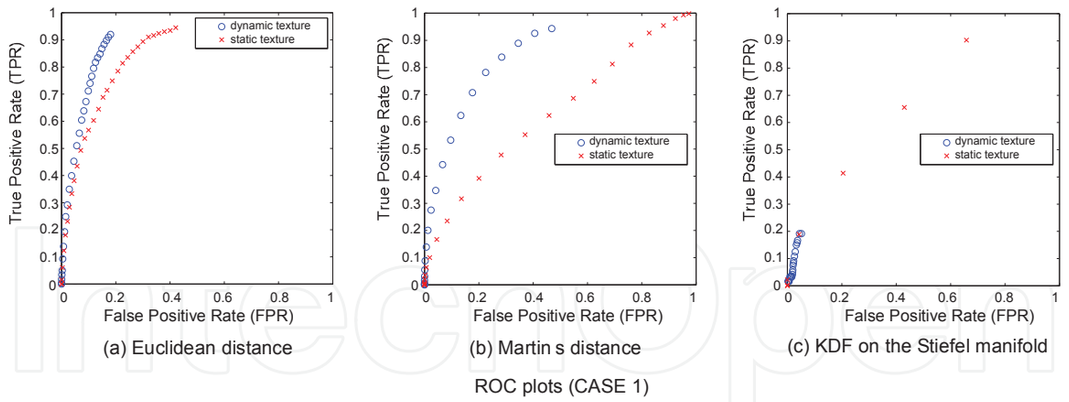


Fig. 11. ROC plots for CASE 1 (The proposed 2D-DCT+N4SID algorithm, 8×8 pixel-block sequences, the 1st test).

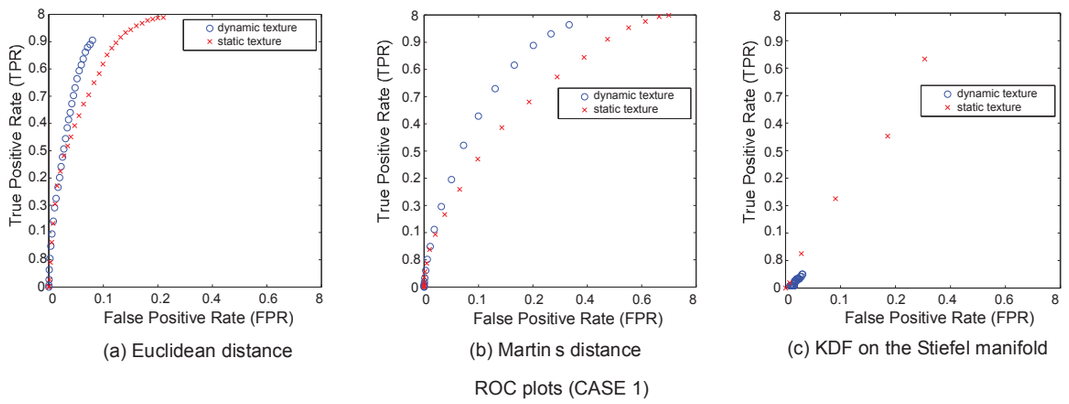


Fig. 12. ROC plots for CASE 2 (The proposed 2D-DCT+N4SID algorithm, 16×16 pixel-block sequences, the 1st test).

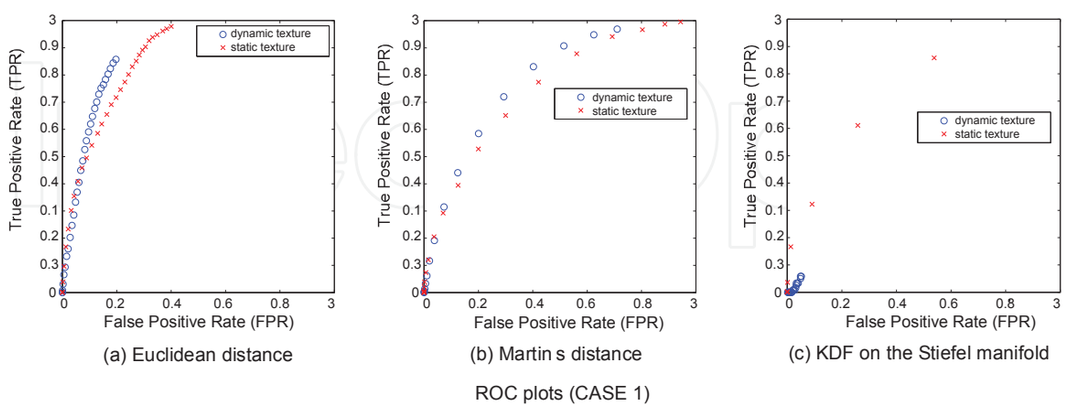


Fig. 13. ROC plots for CASE 3 (The proposed 2D-DCT+N4SID algorithm, 32×32 pixel-block sequences, the 1st test).

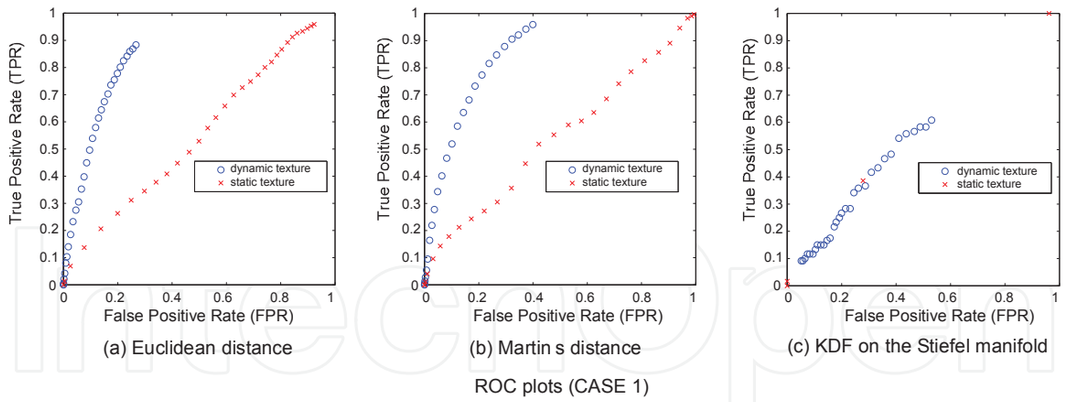


Fig. 14. ROC plots for CASE 1 (The conventional PCA-ID algorithm, 8×8 pixel-block sequences, the 1st test).

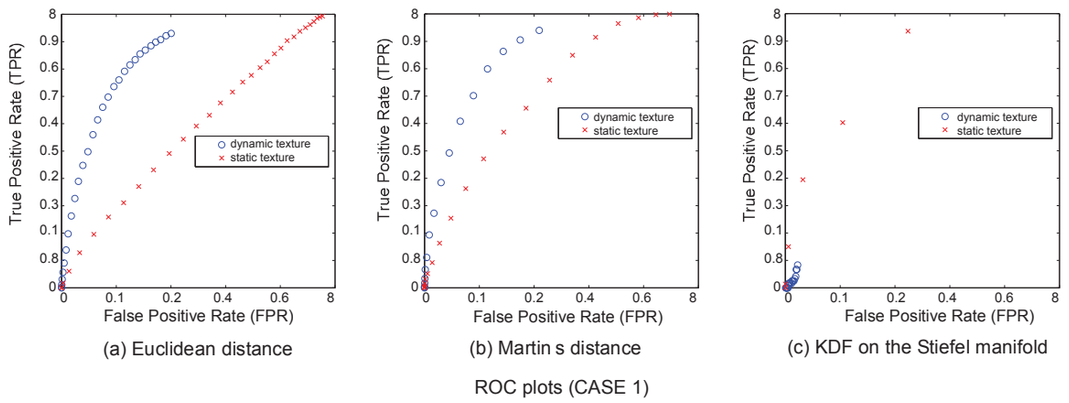


Fig. 15. ROC plots for CASE 2 (The conventional PCA-ID algorithm, 16×16 pixel-block sequences, the 1st test).

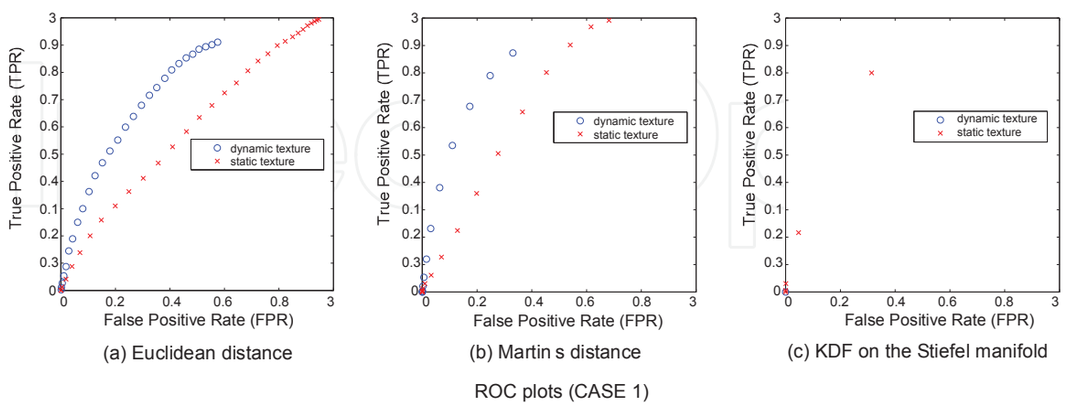


Fig. 16. ROC plots for CASE 3 (The conventional PCA-ID algorithm, 32×32 pixel-block sequences, the 1st test).

Computational time (2D-DCT+N4SID)	Computational time (PCA-ID)
CASE 1 (8 × 8 pixels): 35.2 sec	CASE 1 (8 × 8 pixels): 30.0 sec
CASE 2 (16 × 16 pixels): 36.2 sec	CASE 2 (16 × 16 pixels): 131.8 sec
CASE 3 (32 × 32 pixels): 38.1 sec	CASE 3 (32 × 32 pixels): 1184.2 sec

Table 9. The computational times for the recognition test.

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.

6. Conclusion

This paper proposes 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. Also, one of the metrics computed from the proposed model estimation algorithm shows more desirable characteristic in the ROC space.

7. Future work

In future works, ditance metric suitable to distinguish various types of terrain textures as well as dynamical properties of rover 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.

8. Acknowledgement

The author would like to thank Dr. N. Ichimura in National Institute of Advanced Industrial Science and Technology (AIST) in Japan for his kind interest and valuable comments. The author also thanks 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).

9. References

Chikuse, Y. (2003). *Statistics on special manifolds, Lecture Notes in Statistics*, Springer, New York, NY.

De Cock, K. & De Moor, B. (2000). Subspace angles between linear stochastic models, *Proceedings of the 39th IEEE Conference on Decision and Control (CDC2000), Vol.2*, pp. 1561–1566.

Dima, C., Vandapel, D. & Hebert, M. (2004). Classifier fusion for outdoor obstacle detection, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 665–671.

Halatci, I., Brooks, C. & Iagnemma, K. (2007). Terrain classification and classifier fusion for planetary exploration rovers, *Proceedings of the IEEE Aerospace Conference*, pp. 1–11.

Helmick, D., Angelova, A. & Matthies, L. (2009). Terrain adaptive navigation for planetary rovers, *Journal of Field Robotics* Vol. 26(No. 4): 391–410.

Ishigami, G., Nagatani, K. & Yoshida, K. (2007). 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)*, pp. 2361–2366.

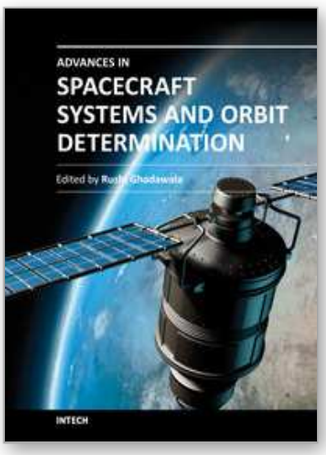
Martin, R. (2000). A metric for ARMA processes, *IEEE Transactions on Signal Processing* Vol. 48(No. 4): 1164–1170.

Overschee, P. & Moor, B. (1994). N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems, *Automatica*, Vol.30(Issue 1): 75–93.

Saisan, P., Doretto, G., Wu, Y. & Soatto, S. (2001). Dynamic texture recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01), Vol.2*, pp. 58–63.

Turaga, P., Veeraraghavan, A. & Chellappa, R. (2008). Statistical analysis on Stiefel and Grassmann Manifolds with applications in Computer Vision, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'08)*, pp. 1–8.

Witten, I., Frank, E. & Hall, M. (2011). *Data Mining: Practical Machine Learning Tools and Techniques 3rd ed.*, Morgan Kaufmann, Burlington, MA.



Advances in Spacecraft Systems and Orbit Determination

Edited by Dr. Rushi Ghadawala

ISBN 978-953-51-0380-6
Hard cover, 264 pages
Publisher InTech
Published online 23, March, 2012
Published in print edition March, 2012

"Advances in Spacecraft Systems and Orbit Determinations", discusses the development of new technologies and the limitations of the present technology, used for interplanetary missions. Various experts have contributed to develop the bridge between present limitations and technology growth to overcome the limitations. Key features of this book inform us about the orbit determination techniques based on a smooth research based on astrophysics. The book also provides a detailed overview on Spacecraft Systems including reliability of low-cost AOCS, sliding mode controlling and a new view on attitude controller design based on sliding mode, with thrusters. It also provides a technological roadmap for HVAC optimization. The book also gives an excellent overview of resolving the difficulties for interplanetary missions with the comparison of present technologies and new advancements. Overall, this will be very much interesting book to explore the roadmap of technological growth in spacecraft systems.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen