Computer-Vision-Based Wheel Sinkage Estimation for Robot Navigation on Lunar Terrain

Guruprasad M. Hegde, *Student Member, IEEE*, Cang Ye, *Senior Member, IEEE*, Christopher A. Robinson, Ashley Stroupe, *Member, IEEE*, and Edward Tunstel, *Fellow, IEEE*

***Abstract*—This paper presents a wheel sinkage detection method that may be used in robotic lunar exploration tasks. The method extracts the boundary line between a robot wheel and lunar soil by segmenting the wheel–soil image captured from a video camera that monitors wheel–soil interaction. The detected boundary is projected onto the soil-free image of the robot wheel to determine the parameters of wheel sinkage. The segmentation method is based on a graph theory. It first clusters a wheel–soil image into homogeneous regions called superpixels and constructs a graph on the superpixels. It then partitions the graph into segments by using normalized cuts. Compared with the existing wheel sinkage detection methods, the proposed algorithm is more robust to illumination condition, shadows, and dust (covering the wheel). The method's efficacy has been validated by experiments under various conditions.**

***Index Terms*—Lunar robotics, normalized cuts, terrain interface angles, wheel sinkage estimation, wheel–soil interaction.**

I. Introduction

MOBILE robots are being used extensively in planetary exploration tasks. An important issue in such tasks is effective robot mobility. Sufficient traction from wheel–terrain/soil interaction is the key to maintain effective robot mobility. In general, a hard terrain produces greater traction than a soft terrain and thus more effective mobility. When moving on a soft terrain, a robot's wheels may slip and/or sink into soil and experience traction loss. Excessive wheel sinkage may result in total traction loss and make the robot immobile. In other words, the efficiency of robot mobility depends on effective wheel–terrain interaction and wheel sinkage information can be used by a robot to assess the effectiveness of wheel–soil interaction and apply corrective action to mitigate the ill-effect of soft soil on the robot mobility.

Almost the entire lunar surface is covered with regolith that contains fine soil grains (<1 cm in diameter) and much finer dust (<100 μm in diameter). The thickness of the regolith varies with location from 2 to 20 m. The properties of lunar regolith are very different from that of the soil of Earth or Mars. To effectively operate a robot on the lunar terrain, the effects of lunar dust must be addressed first. The Apollo missions have revealed that lunar dust has severe degrading effects on extravehicular activities. The Apollo astronauts cited multiple problems caused by lunar dust that adversely affected their activities on the lunar surface. Apollo astronaut John Young regarded dust as "the number one concern in returning to the moon." For robotic surface operation, these effects include vision obscuration of imaging devices, loss of traction, clogging of mechanisms, and abrasion of materials. These problems must be studied and mitigated in order to operate mobile robots (e.g., a sample collecting robot [1]) efficiently and safely in the future robotic lunar surface missions.

This paper is concerned with wheel–lunar–soil interaction. A substantial amount of existing research [2]–[7] on robot–terrain interaction has focused on the soil of Earth or Mars and does not directly address issues germane to robotic lunar exploration. Therefore, it becomes necessary to develop new methods to measure wheel–terrain interaction parameters such as depth of sinkage and terrain interface angles (TIA) under lunar surface conditions. With multiple space agencies turning attention to robotic exploration of the moon in recent years, studies related to lunar wheel–soil interaction have been initiated [8]–[12]. In our earlier work [12], we concentrated on the detection of sinkage depth by a computer vision method. In this paper, we extend the method to the detection of both sinkage depth and TIAs for robotic lunar exploration tasks. As mentioned before, wheel sinkage and TIA information can be used to evaluate the effectiveness of wheel–terrain interaction. In a lunar exploration task, a robot may use wheel sinkage information (perhaps together with wheel slippage information) to avoid further traversal into soft terrain that may potentially lead to entrapment and/or control its speed to restore traction.

A challenging issue in the wheel sinkage detection problem is that the lunar dust may completely cover the robot wheels and make it difficult to reliably identify the wheel–soil boundary in varying illumination conditions. Active 3-D cameras, such as structured light range sensor (e.g., Microsoft Kinect sensor), Flash LIDAR Camera (FLC) (e.g., SwissRanger [13], and

Manuscript received August 29, 2012; revised December 15, 2012 and March 30, 2013; accepted April 23, 2013. Date of publication June 6, 2013; date of current version July 8, 2013. Recommended by Technical Editor E. Richer. This work was supported in part by NASA under the NASA EPSCoR RID Award and the NASA EPSCoR Research Award. An earlier version of this paper was presented at the IEEE International Conference on Mechatronics and Automation, Xi'an, China, August 4–7, 2010.

G. M. Hegde was with the Department of Systems Engineering, University of Arkansas at Little Rock, Little Rock, AR 72204 USA. He is now with Jaguar Land Rover, Ltd., Whitley, CV3 4LF, U.K. (e-mail: gmhegde@ualr.edu).

C. Ye is with the Department of Systems Engineering, University of Arkansas at Little Rock, Little Rock, AR 72204 USA (e-mail: cxye@ualr.edu).

C. A. Robinson was with the Department of Systems Engineering, University of Arkansas at Little Rock, Little Rock, AR 72204 USA. He is now with FIS, Jacksonville, FL 32204 USA (e-mail: Christopher.joe.Robinson@gmail.com).

A. Stroupe is with the Advanced Robotic Control Group, Mobility and Robotic Systems Section, Jet Propulsion Laboratory, Pasadena, CA 91109 USA (e-mail: ashley.w.stroupe@jpl.nasa.gov).

E. Tunstel is with the Space Department, The Johns Hopkins University Applied Physics Laboratory, Laurel, MD 20723 USA (e-mail: edward.tunstel@jhuapl.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2013.2264095

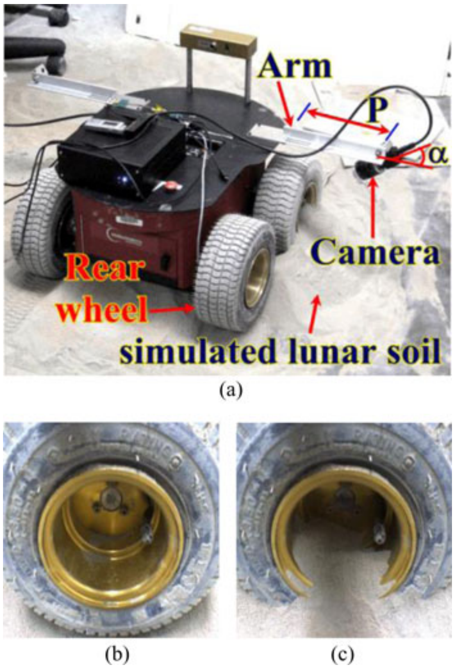


Fig. 1. Wheel sinkage detection by computer vision: a firewire camera is used to monitor each of the wheels. The depth of the wheel sinkage can be determined by measuring the soil-obscured surface of the wheel.

TigerEye FLC [14]), are not susceptive to ambient lighting and the 3-D range data can be used to distinguish soil surface from wheel surface. However, most of the 3-D camera use infrared illumination and, thus, cannot be used under sunlight. The only exception is the TigerEye FLC that uses laser pulse for illumination. However, it has a much higher power dissipation, larger size and weight, and higher cost than a monocular camera. In addition, a flight-ready FLC is currently not available and may cost a lot more than its commercially available counterpart. Therefore, a monocular camera-based wheel sinkage detection method is still an appealing option.

In this paper, we use a Pioneer 3-AT robot as our robotic platform for demonstrating the wheel sinkage detection method. Fig. 1 depicts the idea of wheel sinkage detection for this robotic platform and our experimental setup for validating the sinkage detection method. A machine vision camera is used to monitor each of the robot's wheels. The camera is mounted on a support arm (arm length $P = 20$ cm) installed on the robot body. It looks downward on the wheel with a tilt angle $\alpha = 55^\circ$. The focal length of the camera's lens is 6 mm. Fig. 1(a) shows the robot on a simulated lunar terrain. The lunar regolith simulant used in this study is a mixture of 1/3 cement and 2/3 play sand. The cement powder contains particles with size ranging from 1 to 100 μm, which is fine enough to simulate the lunar dust. The play sand simulates larger grains of the lunar regolith. Fig. 1(b) shows the rear wheel (without sinkage). The front wheel of the robot sinks into the simulated lunar soil as shown in Fig. 1(c) and a part of the wheel surface is obscured by the soil. The depth of the sinkage can be estimated by measuring the obscured surface of the wheel.

A similar idea has been used in a single-wheel testbed [5] where a pattern of concentric black circles on a white background is attached to the wheel to simplify sinkage detection. However, that approach does not work well for a lunar robot as the lunar dust, which tends to stick to a wide variety of materials, may cover the pattern and make the pattern-based detection scheme fail. Brooks *et al*. [4] introduce a method to identify the wheel–terrain interface points (WTIP) based on the fact that the wheel rim intensity is different from the terrain intensity. There are two WTIPs, at each of which the wheel enters/exits the soil. The method computes the locations of the WTIPs as the points with maximum change in intensity between image rows. A similar method that detects the WTIPs based on intensity difference is presented in [15] where a differential edge detector is used to identify the WTIPs along the lower part of the wheel on the binary wheel–terrain image (obtained by thresholding the wheel–terrain image). The advantages of these intensity-based methods are their simplicities and computational efficiencies. But they work well only if the contrast (intensity difference) between the wheel and terrain is sufficiently large. In our case, a dust-covered wheel might not be distinguishable from the terrain. Therefore, the simple intensity-based method may not work well. It may misidentify the interface points and result in incorrect sinkage measurement.

In this paper, we present a new wheel sinkage measurement method based on image segmentation. Our method identifies the entire wheel–soil boundary by partitioning a wheel–soil image into soil region and nonsoil region. The method may result in more accurate wheel sinkage and TIA measurements. Because it treats the soil region as a single continuous segment in determining the wheel–soil boundary, it is thus less sensitive to the misidentification of a WTIP. However, to ensure the method functions well the segmentation process must be robust to lighting condition, shadows, and dust covering the wheel. Several representative image segmentation methods with low computational cost have been investigated and their performances in locating the soil region are compared. Based on the comparison, we choose to devise our sinkage measurement method using the normalized cuts (NCs) approach.

The first segmentation technique that we studied is Otsu's Threshold (OT) method [16]. The method assumes that an image is composed of two basic classes: foreground and background. By using the image's histogram, the method computes an optimal threshold value that maximizes the difference between image pixels falling into these two classes. This method works well for images with high contrast. It may not work properly in our case where the difference between wheel and soil may be indistinctive. Another image segmentation method is to employ the k-means clustering technique [17]. As an initial step, the method computes an intensity histogram of the image and assigns k centroids with random intensity values. The following grouping process is repeated until it converges (i.e., no image pixel changes its group). First, each pixel is assigned to the closest centroid (closeness is measured in term of intensity difference) resulting in k regions. Second, the centroids of the regions are recomputed. A disadvantage of this method is that the randomly generated initial centroids may result in isolated segments that make it difficult to determine the soil region in its entirety. Recent advancements in image segmentation have led to the development of algorithms [18]–[21] based on a spectral graph theory which are known to produce reliable segmentation results. It has been shown in [19] that the NCs clustering method performs better than other spectral graph partitioning methods, such as the averaged cuts [20] and minimum cuts [21] methods, and produces robust segmentation results with a smaller number of isolated segments. Therefore, we employ the NC method for image segmentation in this paper. It should be noted that our goal is not to demonstrate that the NC is the best segmentation method for sinkage detection, instead we simply want to show that the NC method is better than those methods with low computational cost, such as the OT and k-means methods and therefore is appropriate for this paper.

This paper is organized as follows. In Section II, we briefly introduce the NC method. In Section III, we detail the proposed approach for wheel sinkage measurement. In Section IV, we present the experimental results and compare our proposed method with the OT and k-means methods. The paper is concluded in Section V.

II. Normalized Cuts Method

A. Image Segmentation as a Graph Partitioning Problem

Image segmentation can be modeled as a graph partitioning problem for which an image is represented as a weighted undirected graph $G = (V, E)$ where each pixel is considered as a node V_i. An edge $E_{i,j}$ is then formed between a pair of nodes V_i and V_j. The weight of each edge is calculated as a function of similarity between each pair of nodes. In partitioning an image into a number of segments (disjoint sets of pixels) $V_1, V_2, V_3, \ldots, V_m$, the goal is to maximize the similarity of nodes in a set and minimize the similarity across different sets. With the NC algorithm, the optimal bipartition of a graph into two subgraphs A and B is the one that minimizes the Ncut's value given by

$$\text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)} \tag{1}$$

where $\text{cut}(A, B) = \sum_{u \in A, v \in B} w_{u,v}$ is the dissimilarity between A and B, and $w_{u,v}$ is the weight calculated as a function of the similarity between nodes u and v. $\text{assoc}(A, V)$ is the total connection from nodes in A to all nodes in V, while $\text{assoc}(B, V)$ is the total connection from nodes in B to all nodes in V. From (1), we can see that a high similarity among nodes in A and a low similarity across different sets A and B can be maintained by the minimization process. Given a partition of nodes that separates a graph V into two sets A and B, let x be an $N = |V|$ dimensional indicator vector, $x_i = 1$ if the ith node is in A and –1, otherwise. Let $d_i = \sum_j w_{i,j}$ be the total connection from the ith node to all other nodes. With the above definitions, $\text{Ncut}(A, B)$ in (1) can be calculated. According to [19], if x is relaxed to take on continuous values, the optimal partitions can be obtained by splitting the graph using the eigenvector corresponding to the second smallest eigenvalue of the system

$$(D - W)y = \lambda D y \tag{2}$$

where $D = \text{diag}(d_1, d_2, \ldots, d_n)$, $d_i = \sum_j w_{i,j}$ and $W=[w_{i,j}]$.

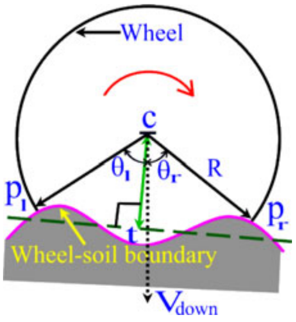


Fig. 2. Detection of wheel sinkage.

B. Grouping Algorithm

The algorithm for grouping of pixels in an image $\boldsymbol{I}$ consists of the following steps.

1) Consider image $\boldsymbol{I}$ as an undirected graph $G = (V, E)$ and construct a similarity matrix W. As stated before, each element of W is the weight of edge $w_{i,j}$ and is calculated by

$$w_{i,j} = \exp\left[-\frac{\|F(i) - F(j)\|_2^2}{\sigma_F}\right] \times \exp\left[-\frac{\|X(i) - X(j)\|_2^2}{\sigma_z}\right] \tag{3}$$

if $\|X(i) - X(j)\|_2 \leq r$ pixels; or $w_{i,j} = 0$ otherwise. Here, $X(p)$ (p stands for i or j) is the spatial location of nodes p, and $F(p)$ is the brightness (or color information) of pixel p. $\|\bullet\|_2$ denotes the L^2-norm of a vector. This means that $w_{i,j} = 0$ for a pair of nodes i and j if they are more than r pixels apart. In other words, (3) computes each weight by taking into account the global information—distance between the two pixels. The heuristic behind this treatment is that two distant pixels are not likely to belong to a segment even if they have similar brightness.
2) Solve (2) for the eigenvectors with the smallest eigenvalues.
3) Use the eigenvector corresponding to the second smallest eigenvalue to bipartition the image by finding the splitting points such that its Ncut value is minimized.
4) Recursively repartition the segments (go to step 1).
5) Exit if the Ncut value for every segment is over some specified threshold.

III. Proposed Method

As mentioned before, our proposed method is to segment the wheel–soil image and then locate the wheel–soil boundary for sinkage and TIA measurements. Fig. 2 illustrates the idea for these measurements. The method applies most effectively to wheels with an outer sidewall that is solid or has an annulus or tire of appreciable radial thickness (as opposed to wheels with thin rims and essentially hollow sidewalls). Here, the robot's wheel sinks into the soil. The center of the wheel is denoted by c. The wheel–soil boundary is labeled in pink color and the wheel–soil interface points are denoted by p_l for the left half of the wheel and p_r for the right half. Points p_r and p_l correspond to the start and end points of the wheel–soil boundary within the wheel, respectively. This means that the wheel enters the soil at p_r and exits the soil at p_l. The left TIA θ_l is defined as the angle between cp_l and V_{down}, while the right TIA θ_r is the angle between cp_r and V_{down}. Here, V_{down} is a unit vector parallel to the gravity vector. To determine the depth of wheel sinkage, we fit a least-squares line (LSL) to the wheel–soil boundary. The wheel sinkage is estimated as $S_e = R - h$, where R is the wheel's radius and h (represented by line *ct* in Fig. 2) is the distance between the wheel center and the LSL. Here, R is assumed to be constant, representing a rigid wheel. However, known amounts of compression for pressurized or deformable wheels could be accounted for without difficulty. The estimated TIAs θ_{le} and θ_{re} can be calculated if the locations of p_l, p_r and the robot's pitch angle are known [4]. Without loss of generality, we place the robot in a terrain with a negligible pitch angle. Therefore, V_{down} always points downward in the wheel–soil image. We calculate the TIAs by

$$\theta_{me} = \cos^{-1}\left(\frac{V_{\text{down}} \cdot cp_m}{||cp_m||}\right) \tag{4}$$

where $m = l, r$ and $|| \cdot ||$ represents the vector norm. It is noted that sinkage detection is performed in the wheel plane as depicted in Fig. 2. According to [3], the tangential forces (a.k.a. drawbar pull) exerting on the wheel is a funtion of the TIAs and the radial stress (between the wheel and soil), which is a function of the sinkage depth S. In other words, the estimated sinkage depth and TIAs are essential parameters for calculating wheel traction.

In this paper, we use the NC method for image segmentation. A direct implementation of the NC method on the wheel–soil image is computationally expensive as the number of pixels (i.e., node number) is large. In [22], this problem is alleviated by down sampling the input gray image to a reasonable size. However, the weight computation is still unaffordable. We resolve this computational bottleneck by 1) partitioning the image into a number of homogeneous groups, called superpixels (SPs); 2) constructing a graph that takes each SP as a node; and 3) applying the NC method to the graph. The spatial location and color of each SP is computed as the centroid and mean color value of its image pixels. In our method, a node of graph G corresponds to a region rather than an image pixel. This substantially reduces the graph's node number and thus the computational cost. The proposed method is described in the following sections.

A. Camera Calibration and Homography Estimation

Each pixel of the soil boundary detected in a wheel–soil image has to be projected back onto the actual wheel plane for calculating the sinkage depth and TIAs. The homography between the wheel plane and image plane needs to be determined. We use the method in [23] to estimate the homography. The same method can also be found in [15]. Here, we omit the details for conciseness. If the camera is calibrated (i.e., the intrinsic parameters are known), the homogeneous transformation matrix that maps a 3-D point in the wheel plane to a pixel in the camera's image plane can then be computed using the camera's homography and intrinsic parameter matrix. We first calibrate the camera by a freely available camera calibration toolbox [24] and using a checkerboard (printed on a thin cardboard). We then attach the checkerboard to the wheel with the wheel center aligned with the center of one black grid. We select six grids within the wheel area, mark one corner point on each of the grid, record their coordinates in the wheel frame and manually found their coordinates in the image planes. Using these six corresponding points, we calculate the homography and determine the homogenous transformation matrix. This matrix is used to determine the correspondence between a pixel in wheel–soil image and a point in the 3-D wheel plane. In this paper, the wheel frame is defined in such a way that the X-axis points left horizontally and Y-axis points up vertically with reference the robot's body frame.

B. Masking Out Unwanted Regions

We mask out the irrelevant image region (the inner part of the wheel rim) to save computational time.

C. Obtaining SPs

The resulting image is converted from RGB to L*u*v color space. In our current implementation, we use L*u*v color space due to its perceptual uniformity. The L*u*v image is then clustered into a number of SPs by the mean-shift (MS) algorithm [25].

D. Graph Construction and Partitioning

We construct graph $G = (V, E)$ by treating each SP as a node and calculate the similarities between nodes i and j as

$$w(i,j) = \begin{cases} \exp\left[-\frac{||F_i - F_j||_2^2}{\sigma_F}\right], & \text{if } V_i, V_j \text{ are neighbors} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where F_p is the L*u*v color vector of node p; for $p = i, j$. SP$_i$ and SP$_j$ are considered as neighbors if they have more than one neighboring image pixel. Graph G is then partitioned into a predetermined number of segments, N, by the NC method. A prespecified segment number is not actually prescribed for the NC method; rather, the segmentation process stops when it finds N segments. The reason for doing so is that it is difficult to find an appropriate threshold for the recursive process of the grouping algorithm in Section II-B.

E. Extraction of Soil Region and Computation of Wheel Sinkage and TIAs

The segments at the bottom of the image are grouped together to obtain the soil region. The edge image of this region gives the wheel–soil boundary. For each pixel on the boundary, the corresponding 3-D point in the wheel plane is computed by using the homogeneous transformation matrix. An LSL is fitted to these points on the wheel plane and h is calculated as the distance from the wheel center (0, 0) to the LSL. As mentioned earlier, the depth of wheel sinkage is estimated as $S_e = R - h$ and the TIAs are calculated by locating p_l and p_r, i.e., the left and right extremes of the wheel–soil boundary.

In summary, our proposed wheel sinkage detection method is executed as follows.

1) Acquire a wheel–soil image from the camera and undistort the image using the intrinsic parameters obtained from the camera calibration process.
2) Mask-out the inner part of the wheel rim.
3) Convert the resulting image to L*u*v space and apply the MS algorithm to obtain a number of SPs, denoted by S_i; for $i = 1,\ldots,m$.
4) Construct a graph G on the SPs and compute the similarity matrix W of order $m \times m$ by using (5).
5) Apply the NC algorithm to $graph$ G with W as the input to obtain N segments, c_i for $i = 1, \ldots, N$, each of which contains a number of SPs.
6) Extract the soil region by grouping the segments at the bottom part of the image and find the edge image of the soil region. The portion of the edge within the wheel region, consisting of a set of pixels p_i for $i = 1, \ldots, k$, gives the wheel–soil boundary. The corresponding 3-D points of p_i on the wheel plane are computed.
7) An LSL is fitted to the 3-D points and the distance from the center of the wheel to the LSL, denoted by h, is computed. The depth of wheel sinkage is then given by $S_e = R - h$.
8) The terrain interface points p_l and p_r are determined as the pixels in p_i with the minimum and maximum column number (two extremes of the wheel–soil boundary), respectively; and the TIAs, θ_{le} and θ_{re}, are computed by (4) using their corresponding points on the wheel plane.

IV. Experimental Results

It would be ideal if we could present a numerical comparison between the proposed method and the existing sinkage-detection methods to demonstrate the advantage of the proposed method. Unfortunately, this is not possible because each of existing methods was proposed to deal with a particular situation that is not applicable to our case. To be specific, in [4] the region of interest (ROI) (i.e., image to be processed for WTIP detection) is a very thin annulus that allows the method to use the summed intensity of each row for WTIP detection. Each WTIP is identified as the point with maximum change in intensity between two consecutive rows. However, in our case, the ROI is the entire image with the inner rim masked out. The soil shape varies along the tire all the way to the edges of the image. Therefore, we cannot use the summed intensity for WTIP detection. In other words, the method in [3] cannot be used in our case. In [15], the robot's wheel has a full circular shape (all metal, with similar color) and the intensity of the wheel center's image point is used to threshold the wheel–terrain image into a binary image, which is then processed by an edge detector to identify the WTIPs. This approach cannot be used in our case either as the wheel center and tire have completely different color. Finally, the method in [5] attaches a pattern of concentric circles to make WTIP detection easier. As mentioned earlier, this approach cannot be used in our case as the soil may cover the pattern and make it useless. In view of that all of the above methods will fail in our case, we opt to perform experiments to evaluate the proposed method with similar experimental conditions (e.g., illumination) as in [4].

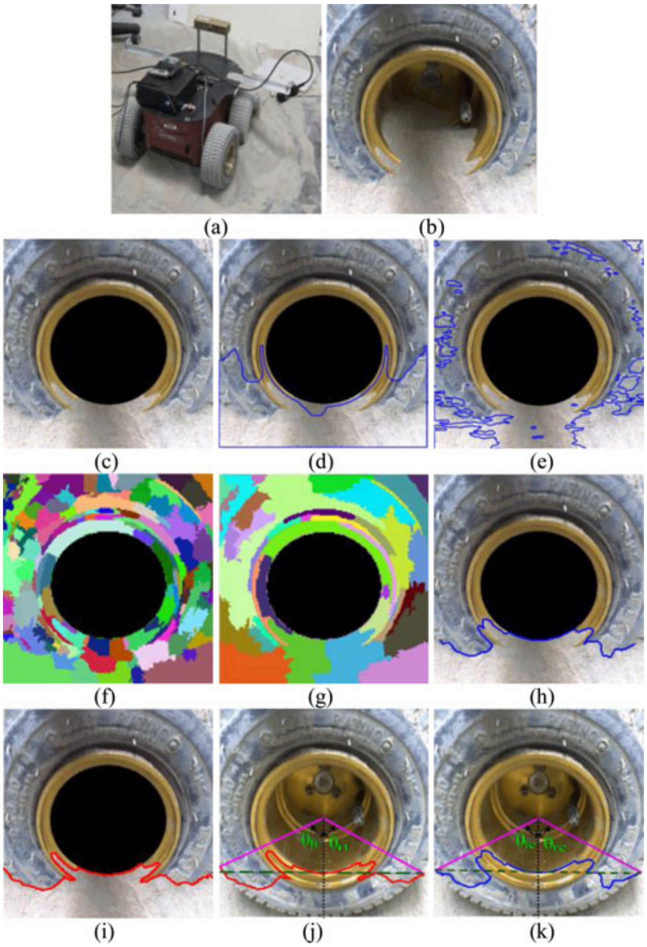


Fig. 3. Extraction of wheel–soil boundary line under normal illumination, SPs = 282, CRF = 7.5 × 10^4: (a) actual scene; (b) wheel image; (c) wheel inner rim masked out from (b); (d) boundary extracted by the OT method; (e) boundary extracted by the k-means method; (f) SPs of (c), (g) results after applying the NC algorithm to (f); (h) boundary extracted by our method; (i) manually labeled boundary; (j) manually labeled boundary, its LSL and TIAs projected on the soil-free wheel image; and (k) extracted boundary of our method, its LSL and TIAs projected on the soil-free wheel image.

In all of our experiments, we down sample the image from 640 × 480 to 320 × 240 pixels after step 2) (see Section III-E) for computational efficiency. For the MS algorithm, the spatial bandwidth h_s, range bandwidth h_r and minimum SP size (in pixel number) M [25] are set to 7, 6.5, and 20, respectively. A smaller h_r value results in misclassification, i.e., an SP contains both wheel pixels and soil pixels. A larger h_r value, however, results in an unnecessarily large number of SPs and thus a higher computational cost for the NC process. The variation in h_s does not have significant impact on segmenting a wheel–soil image into SPs. $M = 20$ is used to prohibit a too small-sized SP. For the NC method, a prespecified number of segments $N = 40$ is used to stop the segmentation process with an appropriate segmentation granularity and thus save computational cost. σ_F [in (5)] is set to 0.045. This value is experimentally determined

TABLE I
SUMMARY OF EXPERIMENTAL RESULTS

Exp	SP	CRF ($\times 10^4$)	Computation Time* (s)		S_t (cm)	S_e (cm)	ΔS (%)	θ_{lt} (Deg)	θ_{le} (Deg)	$\Delta\theta_l$ (%)	θ_{rt} (Deg)	θ_{re} (Deg)	$\Delta\theta_r$ (%)
			SNC	ENC									
1	282	7.4	28.77	9.33	6.4	6.5	1.6	52.0	51.8	-0.4	59.9	58.0	-3.2
2	263	8.5	28.90	7.84	4.5	4.8	6.7	53.0	53.8	1.5	59.5	57.1	-4.0
3	369	4.3	30.41	10.38	7.0	6.7	-4.3	68.2	63.8	-6.4	56.7	56.6	-0.2
4	286	7.2	28.76	8.97	7.0	7.1	1.4	64.4	65.8	2.2	67.9	66.3	-2.4
5	383	4.0	30.60	11.62	0.0	0.0	0.0	NA	NA	NA	NA	NA	NA
6	238	10.4	28.40	8.48	6.3	6.3	0.0	66.8	64.4	-3.6	39.9	41.7	4.5
7	274	7.9	27.74	7.45	5.7	5.7	0.0	60.8	60.5	0.5	51.7	49.9	-3.5
8	284	7.3	30.20	8.13	5.4	5.2	-3.7	60.2	60.1	-0.1	63.2	59.6	-5.7

SP: superpixel, CRF: computation reduction factor, NA: not applicable, SNC: simple normalized cuts, and ENC: extended normalized cuts.
*Computational times of both methods were obtained by using an image with 215 × 215 pixels, which is the maximum image size supported by the SNC MATLAB code [22] that we used.

by applying the NC method (with $N = 40$) on the soil-free image (row e of experiment 5 in Fig. 7) and tuning σ_F such that upon segmentation there is no misclassification in both the wheel and soil regions. In each of our experiments, we assume a constant wheel radius $R = 11$ cm.

To evaluate the accuracy of the wheel sinkage measurement, we need to define the true wheel sinkage and TIAs. In this paper, we manually mark the true wheel–soil boundary and use its LSL to determine the true sinkage S_t. The percentage error of sinkage measurement ΔS is then calculated using S_t and S_e. We then obtain the terrain interface points from the true boundary and compute the true TIAs, θ_{lt} and θ_{rt}, by using (4). The percentage error $\Delta\theta_m$; for $m = l, r$ is computed using θ_{mt} and θ_{me}.

For the results shown in this section, we represent an unlabeled segment in black and a labeled segment with a random nonblack color. We represent the true wheel–soil boundary (manually labeled) in red and extracted wheel–soil boundary in blue and their respective LSLs by a dotted green line.

In the first experiment, we test our method under normal illumination. Fig. 3(a) and (b) shows the experimental setup and the wheel–soil image captured by the firewire camera, respectively. The image after masking out the inner part of the wheel rim is depicted in Fig. 3(c). The boundaries extracted by the OT and k-means methods are shown in Fig. 3(d) and (e), respectively. We can observe that the OT method misclassifies some parts of the wheel as soil. This is because the dust-covered wheel has similar color as the soil and it is difficult to determine a good threshold to accurately partition the wheel and soil regions. The k-means clustering method results in many isolated clusters that make it difficult to extract the soil as a single contiguous region for determining its boundary.

Fig. 3(f) shows the 282 SPs obtained after applying the MS algorithm to Fig. 3(c). The use of the SPs reduces node number from 76,800 to 282. As the edge number of an n-node graph is $n \times (n–1)/2$, the number of edge weight computations is reduced from 2,949,081,600 to 39,621, about 7.5 × 10^4 times smaller. The amount of reduction in the number of edge weight computations will be referred to as computation reduction factor (CRF) further on. The initial grouping of the SPs using the NC method is shown in Fig. 3(g).

As the NC method takes into account pixels' intensity values and their spatial locations in computing their similarity, it results in spatially coherent clusters. Since the bottom segments are in the soil region, we group them into one segment to obtain the soil region. The edge image of this region gives the wheel–soil boundary, which is shown in Fig. 3(h). Fig. 3(i) shows the manually marked true wheel–soil boundary. Fig. 3(j) and 3(k) depict the true and extracted wheel–soil boundaries and their LSLs and TIAs projected onto the soil-free wheel images. For simplicity, we will omit the TIA labeling in subsequent figures associated with the remaining experiments.

We compute h, θ_{le}, and θ_{re} as described in steps 7) and 8) (see Section III-E) and compute the true TIAs θ_{lt} and θ_{rt} using the true interface points. In this case, $h = 4.5$ cm, giving $S_e =$ 6.5 cm. The true wheel sinkage S_t is found to be 6.4 cm. Hence, the sinkage measurement error is $\Delta S = 1.6\%$. The estimated TIAs θ_{le} and θ_{re} are found to be 51.8° and 58.0° while the true TIAs θ_{lt} and θ_{rt} are 52.0° and 59.9°, respectively. Therefore, the TIA measurement errors are $\Delta\theta_l =$ –0.4% and $\Delta\theta_r =$ –3.2%.

It should be noted that one may directly apply the NC method on the wheel–soil image (i.e., without using SPs) to obtain the wheel–soil boundary and then estimate wheel sinkage. Our experiments show that this method yields similar segmentation results and thus similar sinkage estimate as our method does. However, it incurs a much larger memory requirement and computational cost. We call this sinkage-detection method the simple normalized cuts (SNC)-based method, and our method the extended normalized cuts (ENC)-based method. Table I compares the computational times (in seconds) of both methods.

The second experiment is conducted to test our method in a very low illumination condition. We use a single source light with very low output to illuminate the environment [see Fig. 4(a)]. Fig. 4(b) shows the captured wheel–soil image. The results of the OT and k-means methods are shown in Fig. 4(c) and 4(d), where we observe the same problems of the two methods as mentioned in the first experiment. The wheel–soil boundary extracted by our method is shown in Fig. 4(e). As we can see, our method extracts the entire soil region as a single contiguous region and consequently the boundary line. The estimated wheel sinkage is $S_e = 4.8$ cm; the true wheel sinkage is $S_t = 4.5$ cm; and the sinkage measurement error is $\Delta S = 6.7\%$. The estimated TIAs θ_{le} and θ_{re} are 53.8° and 57.1.0° while the true TIAs θ_{lt} and θ_{rt} are 53° and 59.5°, respectively. The TIA measurement errors are $\Delta\theta_l = 1.5\%$ and $\Delta\theta_r =$ –4.0%.

The third experiment is carried out to test our method's performance in the case of self-shadowing. Such conditions occur when a part of the robot's body comes in between the light source and the wheel. We create this condition by placing the

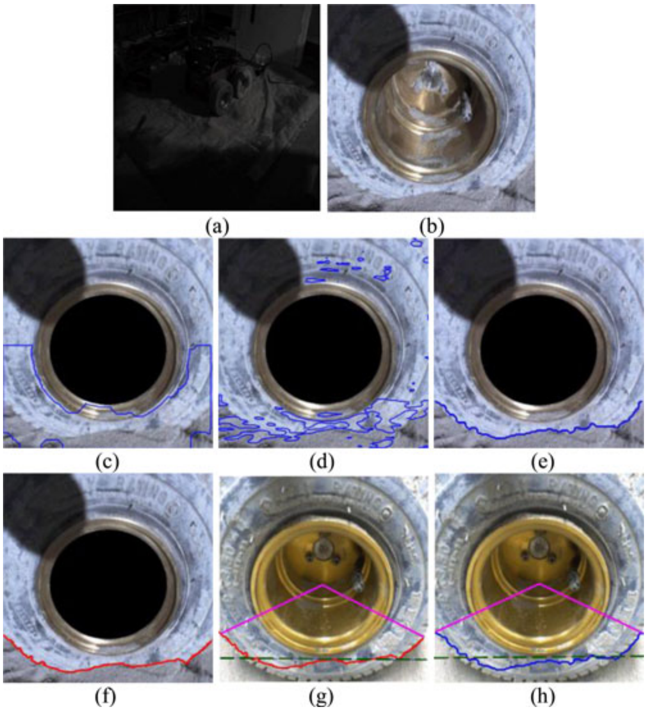


Fig. 4. Extraction of wheel–soil boundary line under very low illumination, (a) actual scene; (b) wheel image; (c) boundary line extracted by the OT method; (d) boundary extracted by k-means method; (e) boundary extracted by our method; (f) manually marked boundary; (g) manually marked boundary, its LSL and TIAs projected on the soil-free wheel image; and (h) extracted boundary of our method, its LSL and TIAs projected on the soil-free wheel image.

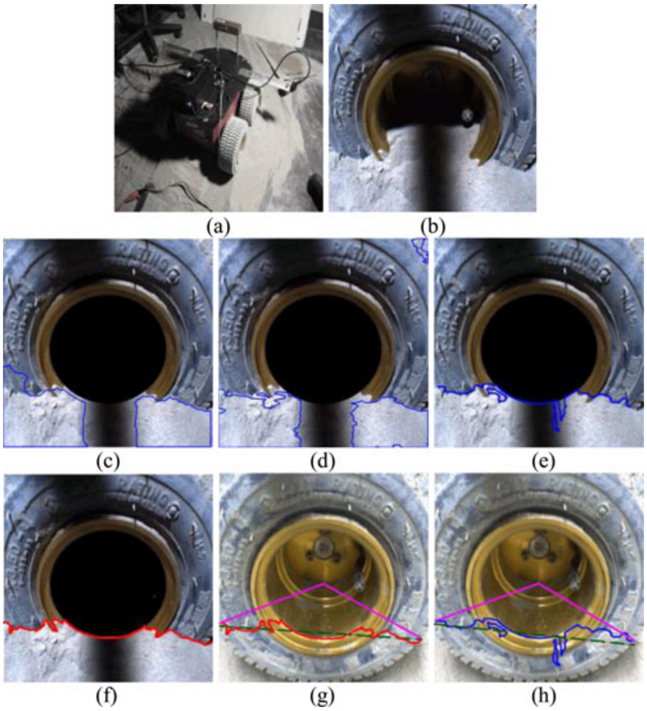


Fig. 5. Extraction of wheel–soil boundary line under self-shadow: (a) actual scene; (b) wheel image; (c) boundary line extracted by the OT method; (d) boundary extracted by k-means method; (e) boundary extracted by our method; (f) manually marked boundary; (g) manually marked boundary its LSL and TIAs projected on the soil-free wheel image; and (h) extracted boundary of our method, its LSL and TIAs projected on the soil-free wheel image.

light source above the robot such that the shadow of the camera and its supporting arm is casted over the wheel. We show the scene in Fig. 5(a). The casted shadow on the robot's wheel can be seen in Fig. 5(b). We can observe that both the OT and k-means methods miss the shadow region as shown in Fig. 5(c) and 5(d), respectively, whereas our method is quite robust to self-shadows and the result is shown in Fig. 5(e). The estimated wheel sinkage is $S_e = 6.7$ cm; the true wheel sinkage is $S_t =$ 7.0 cm; and the sinkage measurement error is $\Delta S =$ –4.3%. The estimated TIAs, θ_{le} and θ_{re}, are 63.8° and 56.6° while the true TIAs, θ_{lt} and θ_{rt}, are found to be 68.2° and 56.7°, respectively. Hence, the TIA measurement errors are $\Delta\theta_l =$ –6.4% and $\Delta\theta_r =$ –0.2%.

In the fourth experiment, we validate our method under nonuniform illumination, i.e., a part of the wheel is brighter than the other. Such conditions occur when a large object (e.g., an astronaut or another robot) casts a shadow on the wheel. To create this condition, we place a piece of cardboard such that it casts a shadow only on a part of the wheel [see Fig. 6(a)]. The captured wheel–soil image is shown in Fig. 6(b). The result of the OT method is shown in Fig. 6(c). Here, we can see that the darker area of the soil is not detected as a soil region because their pixels' intensity values are below the threshold value. Fig. 6(d) shows the results of the k-means method where we can observe that the k-means method fails to detect the darker region and results in isolated clusters. The extracted boundary of our method is shown in Fig. 6(e). For this experiment, the estimated

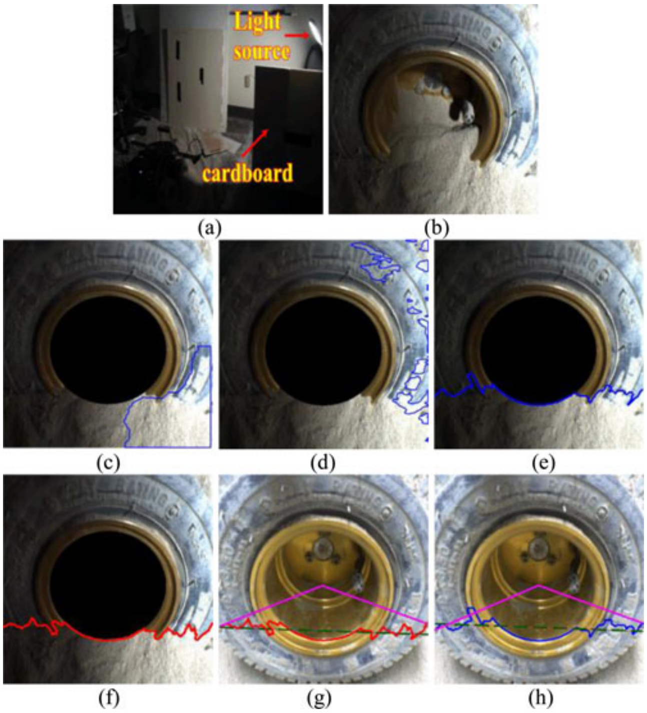


Fig. 6. Extraction of wheel–soil boundary line under nonuniform illumination: (a) actual scene; (b) wheel image; (c) boundary line extracted by the OT method; (d) boundary extracted by k-means method; (e) boundary extracted by our method; (f) manually marked boundary; (g) manually marked boundary its LSL and TIAs projected on the soil-free wheel image; and (h) extracted boundary of our method , its LSL and TIAs projected on the soil-free wheel image.

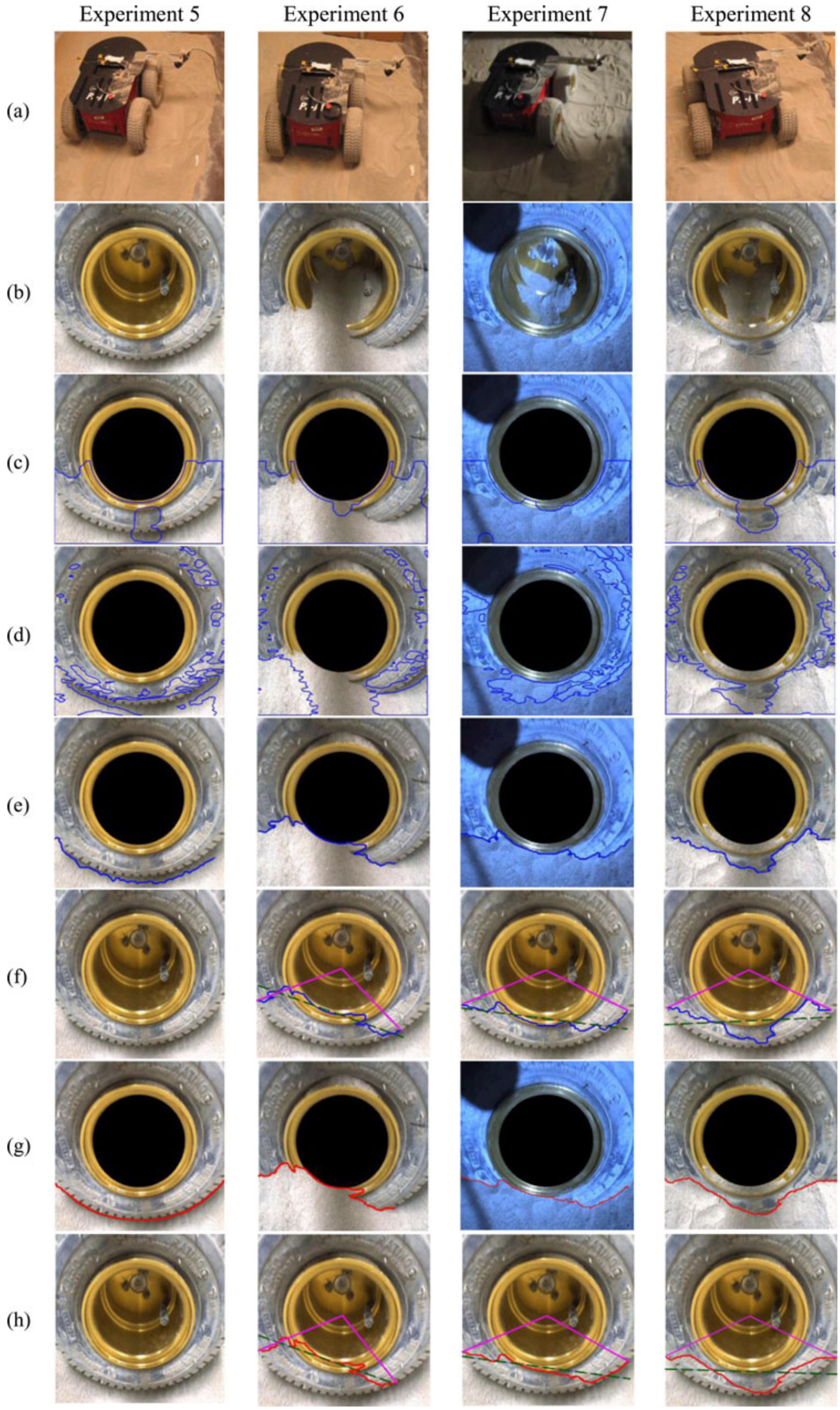


Fig. 7. Results of experiments 5–8: (a) actual scene; (b) wheel image; (c) boundary extracted by the OT method; (d) boundary extracted by k-means method; (e) boundary extracted by our method; (f) extracted boundary its LSL and TIAs projected on the soil-free wheel image; (g) manually labeled wheel–soil boundary; and (h) manually labeled boundary, its LSL and TIAs projected on the soil-free wheel image.

wheel sinkage is $S_e = 7.1$ cm; the true wheel sinkage is $S_t = 7.0$ cm; and the sinkage measurement error is $\Delta S = 1.4\%$. The estimated TIAs, θ_{le} and θ_{re}, are 65.8° and 66.3° and the true TIAs, θ_{lt} and θ_{rt}, are 64.4° and 67.9°, respectively. Hence, the TIA measurement errors are $\Delta\theta_l = 2.2\%$ and $\Delta\theta_r = –2.4\%$.

We have considered other scenarios in addition to those reported above. For brevity, only four of the experiments are shown in Fig. 7 where each column represents the experimental settings and results of one experiment. Experiment 5 is a scenario with zero sinkage (i.e., the wheel is not covered by any soil) and the other three experiments are additional situations with various wheel sinkage. For experiment 5, the boundary extracted by our method is shown in row e. We can see that none of the pixels lies on the wheel. This means that our method detects a sinkage of 0 cm. Therefore, no wheel–soil boundary is projected onto the soil-free wheel image in row f.

Table I summarizes the numerical results of the eight experiments that indicates: 1) The use of SPs in the proposed sinkage detection method results in a CRF in the order of 10^4 that significantly reduces the method's computational time ($\sim$30%); and 2) The method exhibits a consistently good accuracy in measuring wheel sinkage and TIAs.

V. Discussions

In our experiments, a constant wheel radius is assumed. In real scenarios, however, the robot motion induced wheel deformation has to be considered. If the vertical force on the wheel can be measured, the wheel deformation may be estimated with reasonable accuracy and taken into account in computing wheel sinkage. We will also need to consider scenarios with deeper sinkage where the wheel–soil boundary goes over the lower inner part of the rim. Such a scenario can be detected by using the fitting error of the LSL. Then only the parts of the wheel–soil boundary that are on the tire will be used for wheel sinkage computation. We will address these two issues in our future work.

Currently, the method is implemented in MATLAB 7.7 on a computer with an AMD 2.60-GHz Phenom 9950 Quad-core processor and 4-GB RAM. No parallel computation is performed in our current MATLAB code. The operating system used is Microsoft Windows Vista Professional 64-bit. It takes about 10 s to segment a 215 × 215-pixel image. We expect that it may achieve real-time performance if implemented in C++.

It is noted that the camera's suspension arm in a robot that is moving on rough terrain may cause certain vibration to the camera and deteriorate the image. This may result in sinkage measurement error. A solution to this problem is to use a miniature camera with a wider view angle. Since a 320 × 240-pixel image is sufficient for sinkage measurement, we may use a small camera and reduce the mass on the arm. Also, we can reduce the arm's length by using a smaller focal length (i.e., wider view angle) lens. This way, we will alleviate the motion-induced camera vibration. Two other issues that are worthy of further investigation are as follows: First, the camera may have dual use—sinkage detection and hazard avoidance. For instance, images for sinkage detection may be captured and processed every 1 m of traverse and the other images can be used for hazard avoidance. Second, a method to keep the camera lens dust free is essential for the application of the proposed method to lunar exploration tasks.

Finally, the proposed method may be used for wheel sinkage detection for robot navigation on other deformable terrains on Mars or Earth. Sinkage detection in Mars regolith or on a sandy Mars or Earth terrain appears less challenging because in this case the intensity difference between the wheel and terrain is relatively large and the soil particles are not likely to cover the wheel surface. It is expected that the proposed method will work equally well or better for navigation on a sandy terrain. For navigation on off-road muddy terrain on Earth, mud may cover the entire wheel and pose greater challenge to the proposed method. We will need to carry out additional experiments to characterize the proposed method's efficacy under this particular terrain condition.

VI. Conclusion

We have presented a new method that may estimate the parameters of wheel–terrain interaction of a robot in lunar exploration missions. The method relies on extracting the soil region of a wheel–soil image and identifying the wheel–soil boundary line. It employs the NC method to segment the wheel–soil image and obtain the soil region in its entirety. To reduce the computational cost, the mean-shift method is used to partition the image into a number of SPs that form a graph, from which the NC method extracts the soil region. The nature of the NC method—utilizing both spatial distance and the color similarity between pixels in segmentation—leads to a better segmentation performance of the proposed method. To determine the depth of wheel sinkage, we fit a LSL to the pixels of the soil boundary. The wheel surface below this line represents the wheel sinkage and the TIAs are computed using the two extreme points of the wheel–soil boundary.

We have tested our method in different illumination scenarios, such as normal illumination, low lighting, self-shadowing, and nonuniform illumination conditions. The experimental results demonstrate that the proposed method is able to estimate the wheel sinkage parameters with good accuracy in all of these conditions.

Although it is devised for wheel sinkage estimation for robotic vehicles navigating on lunar regolith, the proposed method can also be used in robotic navigation on other soft terrains on Mars or Earth.

With respect to the application of the proposed method on a planetary rover with a flexible suspension scheme, a possible solution is to attach the camera to the suspension mechanism such that there is no relative movement between the camera and wheel. This way, the proposed method may be directly used with the least algorithmic change. Certainly, one may still attach the camera to the robot's frame and use a computer vision method to detect the wheel center. However, this may incur issue of reliable detection of the wheel center and introduce error to wheel sinkage detection.

References

[1] H. Omori, T. Murakami, H. Nagai, T. Nakamura, and T. Kubota, "Development of a novel bio-inspired planetary subsurface explorer: Initial experimental study by prototype excavator with propulsion and excavation units," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 2, pp. 459–470, Apr. 2013.

[2] M. G. Bekker, *Off-Road Locomotion*. Ann Arbor, MI, USA: Univ. Michigan Press, 1960.

[3] K. Iagnemma, S. Kang, H. Shibly, and S. Dubowsky, "Online terrain parameter estimation for wheeled mobile robots with application to planetary rovers," *IEEE Trans. Robot.*, vol. 20, no. 2, pp. 921–927, Oct. 2004.

[4] C. A. Brooks, K. D. Iagnemma, and S. Dubowsky, "Visual wheel sinkage measurement for planetary rover mobility characterization," *Auton. Robot.*, vol. 21, no. 1, pp. 55–64, 2006.

[5] G. Reina, L. Ojeda, A. Milella, and J. Borenstein, "Wheel slippage and sinkage detection for planetary rovers," *IEEE/ASME Trans. Mechatronics*, vol. 11, no. 2, pp. 185–195, Apr. 2006.

[6] L. Ojeda, D. Cruz, G. Reina, and J. Borenstein, "Current-based slippage detection and odometry correction for mobile robots and planetary rovers," *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 366–378, Apr. 2006.

[7] D. Lhomme-Desages, C. Grand, J.-C. Guinot, and F. Ben Amar, "Doppler-based ground speed sensor fusion and slip control for a wheeled rover," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 4, pp. 484–492, Aug. 2009.

[8] K. Yoshida, T. Watanabe, N. Mizuno, and G. Ishigami, "Slip, traction control, and navigation of a lunar rover," presented at the 7th Int. Symp. Artif. Intell., Robot. Autom. Space, Nara, Japan, May 2003.

[9] K. Iizuka, Y. Kunii, Y. Kuroda, and T. Kubota, "Design scheme on wheeled forms of exploration rover in consideration of mechanism between wheels and soil," in *Proc. 9th Int. Symp. Artif. Intell., Robot., and Autom. in Space*, Los Angeles, CA, USA, Feb. 2008.

[10] D. Wettergreen, D. Jonak, D. Kohanbash, S. J. Moreland, S. Spiker, J. Teza, and W. L. Whittaker, "Design and experimentation of a rover concept for lunar crater resource survey," in *Proc. 47th AIAA Aerosp. Sci. Meeting Incl. New Horizons Forum Aerosp. Expo.*, Orlando, FL, USA, Jan. 2009.

[11] L. Ding, H.-B. Gao, Z.-Q. Deng, and J.-G. Tao, "Wheel slip-sinkage and its prediction model of lunar rover," *J. Central South Univ. Technol.*, vol. 17, pp. 129–135, 2010.

[12] G. M. Hegde, C. J. Robinson, C. Ye, A. Stroupe, and E. Tunstel, "Computer vision based wheel sinkage detection for robotic lunar exploration tasks," in *Proc. IEEE Int. Conf. Mechatronics Autom.*, Xi'an, China, 2010, pp. 1777–1782.

[13] [Online]. Available: http: //www.mesa-imaging.ch/prodview4k.php

[14] [Online]. Available: http: //www.advancedscientificconcepts.com/products/tigereye.html

[15] L. Wang, X. Dai, and H. Ju, "Homography-based visual measurement of wheel sinkage for a mobile robot," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2010, pp. 3543–3548.

[16] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[17] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probability*, Berkeley, CA, USA, 1967, vol. 1, pp. 281–297.

[18] J. Malik, S. Belongie, J. Shi, and T. Leung, "Textons, contours and regions: Cue combination in image segmentation," in *Proc. Int. Conf. Comput. Vis.*, 1999, pp. 918–925.

[19] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[20] S. Sarkar and P. Soundararajan, "Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 5, pp. 504–525, May 2000.

[21] Z.-Y. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1101–1113, Nov. 1993.

[22] J. Shi, A MATLAB library for normalized cuts based image segmentation. [Online]. Available: http://www.cis.upenn.edu/~jshi/software/

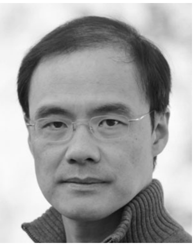
[23] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[24] J.-Y. Bouguet, Camera calibration toolbox for MATLAB, [Online]. Available: http: //www.vision.caltech.edu/bouguetj/calib_doc/

[25] W. Tao, H. Jin, and Y. Zhang, "Color image segmentation based on mean shift and normalized cuts," *IEEE Trans. Syst., Man, Cybern. B, Cyber.*, vol. 37, no. 5, pp. 1382–1389, Oct. 2007.

Guruprasad M. Hegde (S'11) received the B.E. degree from the Sardar Vallabhbhai National Institute of Technology, Surat, India, in June 2005, and the Ph.D. degree from the University of Arkansas at Little Rock, Little Rock, AR, USA, in August 2011.

Since December 2011, he has been a Computer Vision Specialist at TATA ELXSI and Jaguar Landrover, Ltd., Whitley, U.K. His job role includes investigating and developing various intelligent features based on computer vision in the automotive and transportation fields. His research interests include robotics, range data processing, image segmentation, sutonomous navigation, and machine learning.



Cang Ye (S'97–M'00–SM'05) received the B.E. and M.E. degrees from the University of Science and Technology of China, Hefei, Anhui, China, in 1988 and 1991, respectively, and the Ph.D. degree from the University of Hong Kong, Hong Kong, in 1999.

He was a Research Fellow at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from 1999 to 2001, and Research Fellow from 2001 to 2003 and Research Faculty from 2003 to 2005 at the Advanced Technologies Laboratory, University of Michigan, Ann Arbor, MI, USA. He was an Assistant Professor in the Department of Applied Science, from 2005 to 2010, and has been an Associate Professor in the Department of Systems Engineering, University of Arkansas at Little Rock, Little Rock, AR, USA, since July 2010. His research interests include mobile robotics, intelligent systems, and computer vision.

Dr. Ye is a member of the Technical Committee on Robotics and Intelligent Sensing and the IEEE Systems, Man, and Cybernetics Society.

Christopher A. Robinson received the B.S. degree in systems engineering with a minor in mathematics from the University of Arkansas at Little Rock, Little Rock, AR, USA, in 2010.

He participated in summer research opportunities in Germany with Fern Universität in Hagen, INSA in Rouen, France, and a semester exchange with the City University of Hong Kong. He is currently a mainframe COBOL programmer at FIS, Jacksonville, FL, USA.

Ashley Stroupe (M'99) received the B.S. degree in physics from Harvey Mudd College, Claremont, CA, USA, in 1986, the M.S. degree in electrical engineering from George Mason University, Fairfax, VA, USA, in 1998, and the Ph.D. degree in robotics from Carnegie Mellon University, Pittsburgh, PA, USA, in 2003.

She is currently a Robotics Software Engineer at NASA's Jet Propulsion Laboratory, Pasadena, Ca, USA, where she has been since 2003. Her primary role is surface operations for Mars rovers (Spirit, Opportunity, and Curiosity). Her research interests include rover autonomy, autonomous science, autonomy for complex tasks, and rover operations with degraded capabilities.

Edward Tunstel (S'92–M'96–SM'99–F'12) received the B.S. and M.Eng. degrees in mechanical engineering from Howard University, Washington, DC, USA, in 1986 and 1989, respectively, and the Ph.D degree in Electrical Engineering from the University of New Mexico, Albuquerque, NM, USA, in 1996.

He is a Senior Roboticist at The Johns Hopkins University Applied Physics Laboratory, Laurel, MD, USA and served as its Space Robotics and Autonomous Control Lead since 2007. He previously spent 18 years at the NASA Jet Propulsion Laboratory where he was a Senior Robotics Engineer and Advanced Robotic Controls Group Lead, most recently serving as Flight Systems Engineer for autonomous surface navigation and as Lead Analyst for mobility and robotic arm operations for the NASA Mars Exploration Rovers. His research interests include mobile robot navigation, autonomous control, cooperative robotics, robotic systems engineering, and soft computing for autonomous systems, yielding more than 130 publications including several edited books.

Dr. Tunstel is a member of the American Institute of Aeronautics and Astronautics (AIAA) and National Society of Black Engineers. He is an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS and the Vice President for Systems Science and Engineering with the IEEE Systems, Man, and Cybernetics (SMC) Society. He is also a member of the IEEE SMC TC on Robotics and Intelligent Sensing, IEEE RAS TC on Space Robotics, and the AIAA Space Automation and Robotics TC.