Auton Robot (2006) 21:55–64
DOI 10.1007/s10514-006-7230-9

Visual wheel sinkage measurement for planetary rover mobility characterization

Christopher A. Brooks · Karl D. Iagnemma · Steven Dubowsky

Published online: 22 April 2006

Abstract Wheel sinkage is an important indicator of mobile robot mobility in natural outdoor terrains. This paper presents a vision-based method to measure the sinkage of a rigid robot wheel in rigid or deformable terrain. The method is based on detecting the difference in intensity between the wheel rim and the terrain. The method uses a single grayscale camera and is computationally efficient, making it suitable for systems with limited computational resources such as planetary rovers. Experimental results under various terrain and lighting conditions demonstrate the effectiveness and robustness of the algorithm.

Keywords Rough terrain · Mobile robots · Machine vision · Soil characterization · Mobility

1. Introduction

Mobile robots are increasingly being employed in natural outdoor terrain for applications such as forestry, mining, and planetary exploration (Le et al., 1997; Gonthier and Papadopoulos, 1998; Cunningham et al., 1999; Volpe, 2003). In natural terrain, wheel-terrain interaction has a strong influence on robot mobility (Bekker, 1956; Wong, 1976). For example, a robot traversing loose sand might experience substantial wheel sinkage and hence poor mobility (see Fig. 1). Conversely, a robot traversing firm clay might experience small wheel sinkage and better mobility. With knowledge of wheel sinkage, a mobile robot could modulate wheel torque to improve traction, or revise its motion plan to avoid potentially hazardous highly-deformable terrain (Iagnemma and Dubowsky, 2004).

Wheel sinkage has been shown to be a key variable in estimating and predicting wheel-terrain interaction phenomena (Iagnemma et al., 2003; Kang, 2003), which can be important for accurately estimating rover dynamics in deformable terrain (Yoshida and Hamano, 2002). Wheel sinkage is also an important input to terrain identification and classification algorithms (Iagnemma and Dubowsky, 2004; Iagnemma et al., 2002). These algorithms are particularly valuable in scientific studies of soil properties during planetary exploration missions (Volpe, 2003).

The recent experiences of the Mars Exploration Rovers, Spirit and Opportunity, have clearly demonstrated the importance of wheel-terrain interaction in rough terrain mobility (Bridges, 2004; Morris, 2004; Petit, 2004). Significant wheel sinkage and slip has lead to loss of traction for Opportunity during traverses to several scientifically significant sites in both Eagle Crater and Endurance Crater. One of these traverses was intended to move the rover away from a treacherous slope. During Spirit's climb up the Columbia hills, its right-front wheel was dragged through the soil, resulting in significant sinkage and motion resistance. This has led to errors in path tracking and increased power consumption. In April–June 2005, substantial wheel sinkage in loose sandy soil caused Opportunity to be nearly immobilized for several weeks. The ability to measure wheel sinkage and terrain conditions could potentially have enabled vehicle control schemes that could have mitigated these problems.

Little research has addressed the problem of wheel sinkage measurement or estimation. Previous research has measured the change in an articulated suspension's configuration

C. A. Brooks (✉) · K. D. Iagnemma · S. Dubowsky
Department of Mechanical Engineering,
77 Massachusetts Ave., Room 3-473M, Cambridge, MA 02139
e-mail: cabrooks@mit.edu

K. D. Iagnemma
e-mail: kdi@mit.edu

S. Dubowsky
e-mail: dubowsky@mit.edu

Fig. 1 Significant wheel sinkage on mockup of Spirit rover at NASA Jet Propulsion Laboratory (NASA/JPL Image)

for the purposes of improving odometry and determining the sinkage of wheels relative to one another (Wilcox, 1994). However, absolute sinkage is necessary for mobility analysis and terrain identification algorithms. Wheel sinkage can potentially be estimated using visual odometry methods (Olson et al., 2001). Practical implementation of these methods for articulated suspension systems are relatively complex, relying on accurate kinematic models of the rover and the use of stereo-based feature identification and tracking algorithms.

This paper presents a vision-based method for online measurement of robot wheel sinkage in deformable terrain. The algorithm is computationally efficient, and thus is suitable for systems with limited resources, such as planetary rovers. The algorithm relies on the difference in image intensity between the wheel rim and surrounding terrain. Experimental results show that the algorithm is accurate and robust to variation in terrain and lighting conditions.

The algorithm presented here measures rigid wheel sinkage based on an image containing a view of a rover wheel, such as the image captured by the belly-mounted hazard avoidance cameras on Spirit and Opportunity. The MER rovers and Mars Pathfinder rover have all used rigid metallic wheels to reduce complexity and increase traction on rocky slopes. Note that this algorithm can also be applied to Earth-based mobile robots with rigid wheels or pneumatic tires, if the tire inflation pressure is high compared to the terrain stiffness (Bekker, 1956).

2. Algorithm overview

The goal of the algorithm is to measure wheel sinkage in rigid or deformable terrain from a visual image of the wheel. It is assumed that a camera is mounted on the robot body, with a field of view containing the wheel. Sinkage is defined as a pair of angles from the vertical termed left and right terrain interface angles (see Fig. 2). This provides a general description of wheel sinkage in uneven terrain. To determine these angles, only an annular region along the wheel rim (between r_{rim} and r_{wheel}) needs to be examined. This reduces computational requirements by eliminating much of the scene.

It is assumed that the location of the wheel relative to the camera is known. This is a reasonable assumption since many robots have rigid suspensions. Robots with articulated suspensions (such as Sojourner, Spirit, and Opportunity) are generally instrumented with sensors that measure the state of suspension articulation. Visual methods for identifying the wheel center could be implemented, however this would add computational complexity. The proposed approach can be extended to steerable wheels by considering the steering angle as part of the suspension configuration.

It is also assumed that the robot wheel rim is visually distinguishable from the surrounding terrain. This is generally true for rigid, metallic wheels or dark pneumatic tires in natural terrains. Visual contrast can be enhanced by coloring the wheel rim a non soil-like color such as yellow or blue. This pixel-level difference in appearance eliminates the need for computationally-intensive texture analysis or stereo-based correlation. The algorithm instead relies on a relatively simple analysis of grayscale intensity along the wheel rim.

The algorithm consists of the following three steps: (1) wheel rim identification, (2) pixel intensity computation, and (3) terrain interface identification. The following section describes these steps.

3. Algorithm description

The algorithm uses the following coordinate frames (see Fig. 3):

– *Wheel frame*: A non-rotating frame fixed at the wheel hub, with the *x*-*y* plane in the plane of the wheel and the z-axis parallel to the wheel axle.

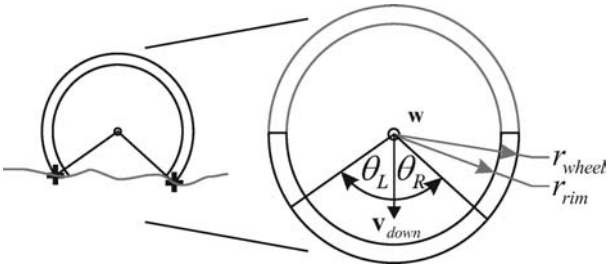

Fig. 2 Rigid wheel sinking into deformable terrain with left (θ_L) and right (θ_R) terrain interface angles shown

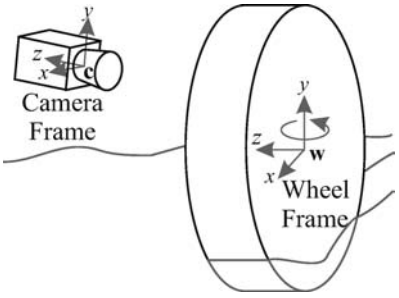

Fig. 3 Illustration of camera and wheel frames

– *Camera frame*: A frame fixed at the camera lens center and with its x and y axes aligned to the image axes.
– *Translated wheel (TW) frame*: The wheel frame translated so its origin coincides with the origin of the camera frame.

The position of a point in a coordinate frame is represented by 4×1 column vectors, using homogeneous coordinate notation. For example, an arbitrary point $\boldsymbol{p}$ can be represented by a 4×1 vector $\mathbf{p}_{\text{wheel}} = [p_{x,\text{wheel}} \quad p_{y,\text{wheel}} \quad p_{z,\text{wheel}} \quad 1]^T$ in the wheel frame. Using this notation, affine transformations may be written as 4×4 matrices. Here three transformation matrices, $\mathbf{T}_{TW}^{\text{wheel}}$, $\mathbf{T}_{camera}^{TW}$, and $\mathbf{T}_{\text{wheel}}^{\text{camera}}$, are defined to relate a position vector expressed in one frame to a position vector expressed in another, such that $\mathbf{p}_{\text{camera}} = \mathbf{T}_{\text{wheel}}^{\text{camera}} \mathbf{p}_{\text{wheel}}$.

3.1. Wheel rim identification and classification

The first step in the algorithm is to identify all pixels that lie in a region of interest of the image. The region of interest is defined as the lower half of the annular region between the inner wheel rim diameter, r_{rim}, and the outer wheel rim diameter, r_{wheel} (see Fig. 2). For rimless wheels or tires, r_{wheel} corresponds to the outer tire diameter and r_{rim} is chosen to be slightly less than r_{wheel}.

For convenience, the camera is modeled as a pinhole camera (Forsyth and Ponce, 2003). This model neglects the distortion caused by lenses, however images may be preprocessed to correct for this distortion. Using this model, the inner and outer wheel rims are projected through the center of the camera lens to form cones $\boldsymbol{W}$ and $\boldsymbol{R}$, respectively. Points on the wheel which lie within the outer wheel rim (and therefore, within $\boldsymbol{W}$) are imaged by pixels on the CCD which also lie within $\boldsymbol{W}$. Points on the wheel which lie within the inner wheel rim are imaged by pixels which lie within $\boldsymbol{R}$.

Cones $\boldsymbol{W}$ and $\boldsymbol{R}$ are represented in the translated wheel frame by 4×4 matrices $\mathbf{W}_{TW}$ and $\mathbf{R}_{TW}$, such that $\mathbf{p}_{TW}^T \mathbf{W}_{TW} \mathbf{p}_{TW} < 0$ for any point $\boldsymbol{p}$ within cone $\boldsymbol{W}$ (and analogously for $\boldsymbol{R}$). Matrices $\boldsymbol{W}_{TW}$ and $\boldsymbol{R}_{TW}$ are calculated using the inner and outer wheel rim diameters and the position (c_x, c_y, c_z) of the camera in the wheel frame:

$$\mathbf{W}_{TW} \equiv \mathbf{A} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -r_{\text{wheel}}^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{A}^{-1} \tag{1}$$

$$\mathbf{R}_{TW} \equiv \mathbf{A} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -r_{\text{rim}}^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{A}^{-1} \tag{2}$$

where

$$\mathbf{A} \equiv \begin{bmatrix} 1 & 0 & c_x & 0 \\ 0 & 1 & c_y & 0 \\ 0 & 0 & c_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Note that conversion to the wheel frame (and analogously to the camera frame) is accomplished as:

$$\mathbf{W}_{\text{wheel}} = \mathbf{T}_{TW}^{\text{wheel}} \mathbf{W}_{TW} \mathbf{T}_{TW}^{\text{wheel}^{-1}}, \tag{4}$$

so that a position vector $\mathbf{p}_{\text{wheel}}$ in the wheel frame lies within $\boldsymbol{W}$ if $\mathbf{p}_{\text{wheel}}^T \mathbf{W}_{\text{wheel}} \mathbf{p}_{\text{wheel}} < 0$.

The region of interest is divided into two sections, corresponding to the left and right halves of the wheel (see Fig. 4). This is done because terrain entry generally occurs in one half of the wheel, and terrain exit occurs in the other. Thus the algorithm will search for one terrain interface in each section. Left and right sections are determined with respect to the vector v_{down}. The vector v_{down} is a unit vector perpendicular to the pitch angle of the vehicle (e.g. on flat terrain, v_{down} is parallel to the gravity vector).

Each pixel in the image can be identified as a member of one of three sets: (1) points belonging to the lower left quadrant of the wheel annulus, S_{left}; (2) points belonging to the lower right quadrant of the wheel annulus, S_{right};

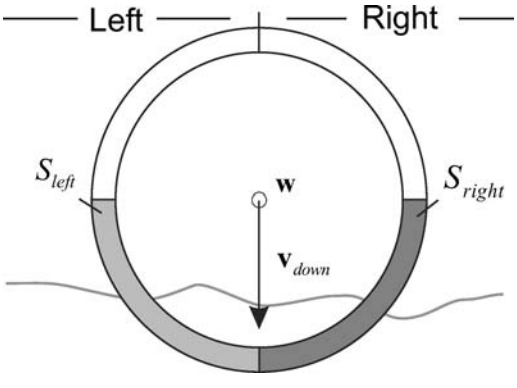

Fig. 4 Annulus sections and v_{down}

and (3) points that are not in the region of interest on the wheel rim.

Because the CCD is a rectangular array of pixels, a 4 × 3 matrix can be used to relate image pixel coordinates to a position vector in the camera frame:

$$\mathbf{p}_{\text{camera}} = \mathbf{T}^{\text{camera}}_{\text{pixel}} \begin{bmatrix} x_{\text{pixel}} \\ y_{\text{pixel}} \\ 1 \end{bmatrix}, \tag{5}$$

where

$$\mathbf{T}^{\text{camera}}_{\text{pixel}} \equiv \begin{bmatrix} scaleX & 0 & f_x \\ 0 & scaleY & f_y \\ 0 & 0 & f_z \\ 0 & 0 & 1 \end{bmatrix}, \tag{6}$$

scaleX and *scaleY* are the number of inches per pixel in the x and y directions, and f_x, f_y, and f_z are the coordinates of pixel (0,0) in the camera frame.

A point p on the camera image plane, corresponding to the pixel $(x_{\text{pixel}}, y_{\text{pixel}})$, lies inside the annular area of interest if it satisfies the following inequality:

$$\mathbf{p}^T_{\text{camera}} \mathbf{W}_{\text{camera}} \mathbf{p}_{\text{camera}} < 0 < \mathbf{p}^T_{\text{camera}} \mathbf{R}_{\text{camera}} \mathbf{p}_{\text{camera}}. \tag{7}$$

If this inequality is satisfied, the point's location on the left or right half can be found by identifying its corresponding projection onto the wheel rim, $\boldsymbol{q}$. Projection is done by finding the pixel's position in the translated wheel reference frame, $\mathbf{p}_{\text{TW}}$,

$$\mathbf{p}_{TW} = \mathbf{T}^{TW}_{\text{camera}} \mathbf{p}_{\text{camera}}, \tag{8}$$

scaling the position vector to lie on the plane of the wheel,

$$\mathbf{q}_{TW} = \begin{bmatrix} -c_z \dfrac{p_{x,TW}}{p_{z,TW}} & -c_z \dfrac{p_{y,TW}}{p_{z,TW}} & -c_z & 1 \end{bmatrix}^T, \tag{9}$$

and writing this position vector in terms of the wheel frame,

$$\mathbf{q}_{\text{wheel}} = \mathbf{T}^{\text{wheel}}_{TW} \mathbf{q}_{TW}, \tag{10}$$

where c_z is the z coordinate of the camera lens in the wheel frame.

The point lies in the right half of the annulus if $(\mathbf{v}_{\text{down}} \times \mathbf{q}_{\text{wheel}}) > 0$, or as the determinant of a matrix:

$$\left| \begin{bmatrix} \mathbf{v}_{\text{down}} & \mathbf{q}_{\text{wheel}} & \begin{matrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{matrix} \end{bmatrix} \right| > 0. \tag{11}$$

If $(\mathbf{v}_{\text{down}} \times \mathbf{q}_{\text{wheel}}) \leq 0$, the point is in the left half.

3.2. Pixel intensity computation

The average grayscale intensity is computed for every row of pixels in S_{left} and S_{right} (see Fig. 5). A row is a set of pixels aligned perpendicular to $\mathbf{v}_{\text{down}}$. n rows are denoted as subsets $r_{\text{left},k} \subset S_{\text{left}}$ and $r_{\text{right},k} \subset S_{\text{right}}$, where $k \in \{1, \ldots, n\}$. Note that n is a function of the spatial resolution Δy and the wheel diameter. A pixel has membership in row k if the following equation is satisfied:

$$\left\lfloor \frac{\mathbf{v}^T_{\text{down}} \mathbf{q}_{\text{wheel}}}{\Delta y} \right\rfloor = k, \tag{12}$$

where Δy is an adjustable parameter corresponding to the smallest change in sinkage the algorithm will detect. Note that decreasing Δy below the imaged resolution of the rim will not increase accuracy.

For each row the summed intensity SI is computed as the sum of each individual pixel's intensity I:

$$SI_{\text{right},k} = \sum_{\mathbf{p} \in r_{\text{right},k}} I(\mathbf{p}) \tag{13}$$

$$SI_{\text{left},k} = \sum_{\mathbf{p} \in r_{\text{left},k}} I(\mathbf{p}) \tag{14}$$

Two $n \times 1$ arrays of summed row intensities are thus formed.

3.3. Terrain interface identification

A one-dimensional spatial filter is employed to smooth the intensity arrays and reduce the effects of noise. Here the summed row intensities are weighted by the number of pixels in a row ($c_{\text{right},k}$ and $c_{\text{left},k}$), to minimize the influence of noise in low pixel-count rows.

A Gaussian distribution with variance $m/2$ is approximated by a binomial distribution w:

$$w_{l,m} = \frac{(2m)!}{2^{2m}(m+l)!(m-l)!} \tag{15}$$

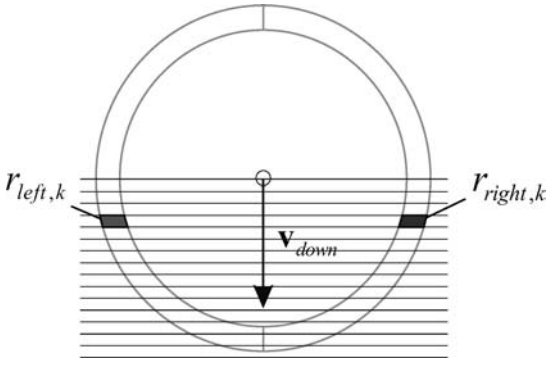

Fig. 5 Assignment of pixels to rows $r_{\text{left},k}$ and $r_{\text{right},k}$.

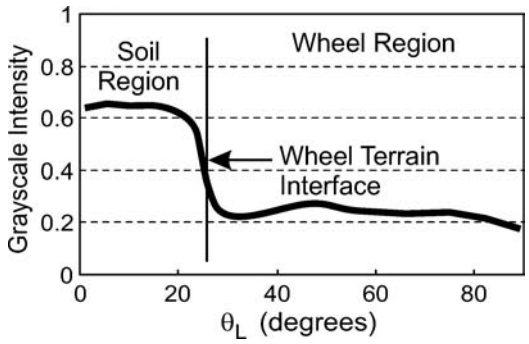

Fig. 6 Sample plot of average pixel intensity vs. angular position

where $l \in \{-m, \ldots, m\}$. This filter is applied to the summed pixel intensities to produce a pair of filtered intensity arrays FI_{right} and FI_{left}:

$$FI_{\text{right},k} = \frac{\sum_{l=-m}^{m} w_{l,m} SI_{\text{right},k+l}}{\sum_{l=-m}^{m} w_{l,m} c_{\text{right},k+l}} \tag{16}$$

$$FI_{\text{left},k} = \frac{\sum_{l=-m}^{m} w_{l,m} SI_{\text{left},k+l}}{\sum_{l=-m}^{m} w_{l,m} c_{\text{left},k+l}} \tag{17}$$

A representative plot of filtered intensity vs. angular position can be seen in Fig. 6. In this example, a dark wheel is partially submerged in light terrain.

The terrain interface location is computed as the point of maximum change in intensity between rows. This exploits the fact that the wheel rim intensity is different from the terrain intensity. The row index with maximum change in intensity is simply:

$$K_{\text{right}} = \arg\max_{k}(FI_{\text{right},k} - FI_{\text{right},k-1}) \tag{18}$$

$$K_{\text{left}} = \arg\max_{k}(FI_{\text{left},k} - FI_{\text{left},k-1}) \tag{19}$$

The interface angles θ_R^* and θ_L^* are then calculated from K_{right} and K_{left} as follows:

$$\theta_R^* = \cos^{-1}\left((K_{\text{right}} - 0.5)\frac{\Delta y}{r_{\text{wheel}}}\right) \tag{20}$$

$$\theta_L^* = \cos^{-1}\left((K_{\text{left}} - 0.5)\frac{\Delta y}{r_{\text{wheel}}}\right) \tag{21}$$

3.4. Sinkage measurement on rigid terrain

The algorithm assumes the presence of a unique maximum change in intensity along the wheel rim. In practice a unique maximum can nearly always be found at the wheel-terrain interface. However, errors can occur when the wheel contacts a rigid patch of terrain, or rigid objects such as rocks. In these situations none of the rim may be occluded by terrain (see Fig. 7). Sensor noise and lighting effects will then lead to false maxima and thus erroneous sinkage values.

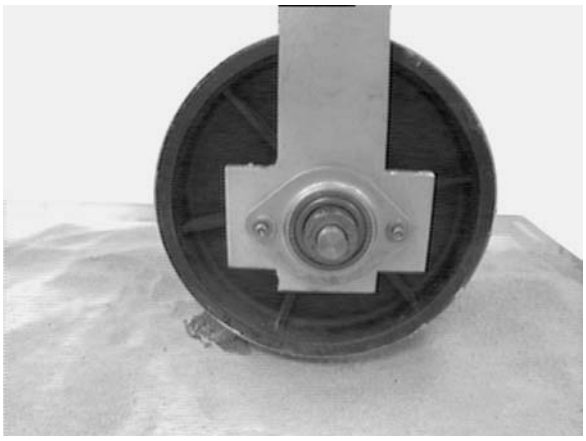

Fig. 7 Wheel rigidly supported by a rock

This problem is addressed by appending a small set of pixels to S_{left} and S_{right} at index $n + 1$. These pixels are taken from an image region below the center of the wheel rim, and are expected to be representative of the local terrain. If the wheel is resting on a rigid terrain surface, the maximum change in intensity will occur between indices n and $n + 1$. This correctly corresponds to a sinkage angle of zero. If the maximum change in intensity occurs in the wheel rim region, the algorithm will operate normally and return the appropriate sinkage angle.

4. Experimental results

4.1. Experiment setup description

Experiments have been performed on the FSRL Wheel-Terrain Interaction Testbed shown in Fig. 8. The testbed consists of a driven wheel mounted on an undriven vertical axis. Weights are added to this axis to replicate different vertical loads on the wheel. Horizontal movement of the wheel is actively controlled. A camera is mounted to the testbed so that it moves horizontally with the wheel, but not vertically. The vertical position of the wheel relative to the camera is sensed with a potentiometer. This position is used to determine the transformations between the reference frames. Suspension configuration sensors would be used to perform this function on a rover.

In this study, images from the testbed camera were collected under different terrain and lighting conditions, including variable wheel slip and terrain unevenness conditions, variable soil colors, and with and without rocks. Lighting was varied from uniform, diffuse illumination to a point-source

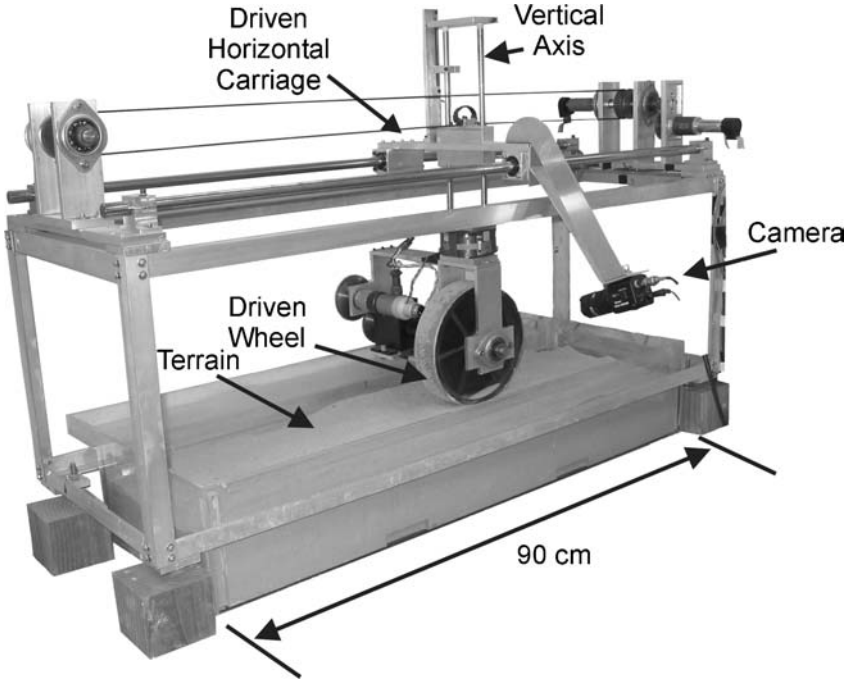

Fig. 8 FSRL Wheel-Terrain Interaction Testbed

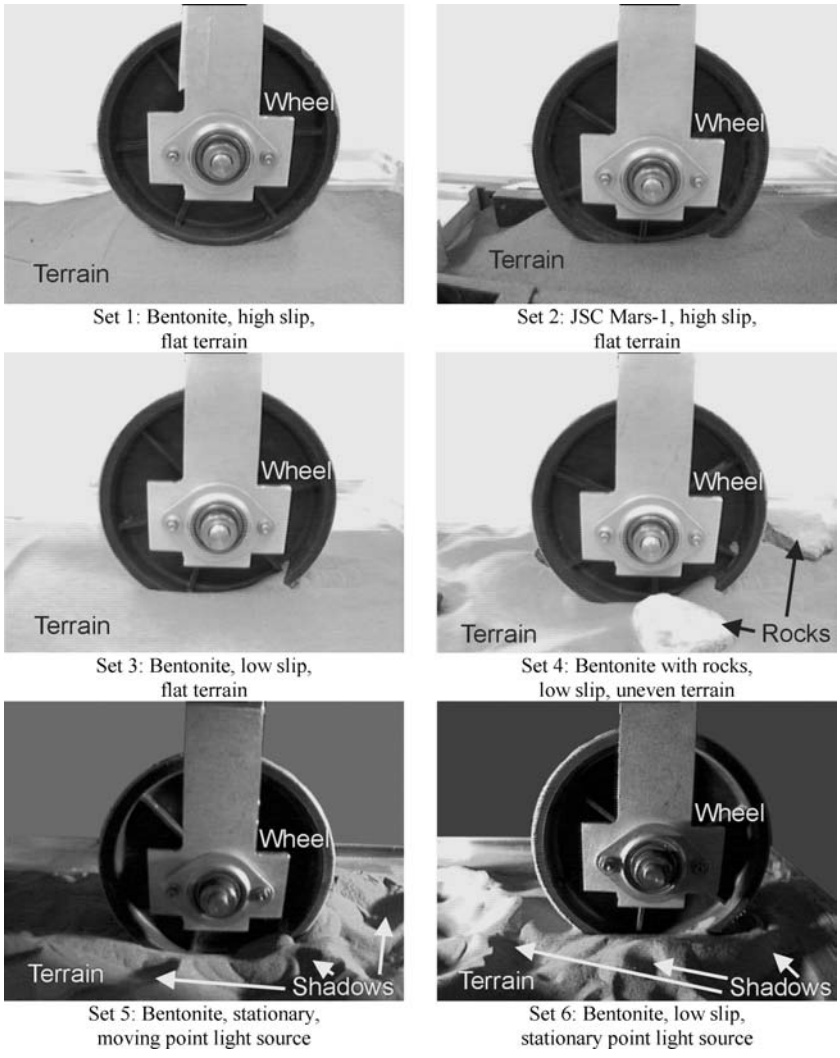

Fig. 9 Sample wheel sinkage measurement images from image sets 1–6

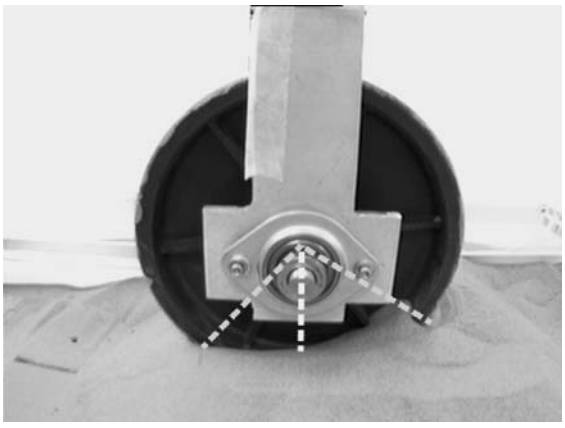

Fig. 10 Sample image of sinkage angles identified by the algorithm

that cast sharp shadows. In the data presented here, nineteen images were collected at two second intervals for each set of conditions. Actual values for the wheel sinkage were manually identified by a human analyst, based on these images.

Figure 9 shows representative images from different conditions. Image set 1 shows a wheel moving through flat bentonite clay under uniform lighting with a high slip ratio. The bentonite clay is dry and granular with a light tan color. The high slip ratio causes the wheel to sink into the clay, yielding a wide range of sinkages.

Set 2 shows a wheel moving with high slip ratio through flat JSC Mars-1 soil simulant under uniform lighting (Allen et al., 1997). JSC Mars-1 soil simulant is a brown mixture of weathered volcanic ash particles developed to simulate the color and consistency of Martian soil. The dark color tests the algorithm effectiveness in situations with low contrast between the wheel and terrain.

Set 3 shows a wheel moving through flat bentonite clay under uniform lighting with low slip and nearly constant sinkage. This simulates the conditions a rover might see when driving over homogeneous terrain.

Set 4 shows a wheel moving through uneven bentonite clay under uniform lighting in the presence of rocks. Rocks occlude the wheel-terrain interface or appear as additional potential interfaces. They also support it rigidly and thus cause conditions with zero sinkage.

Set 5 shows a stationary, sunken wheel in uneven bentonite clay illuminated by a moving point source. The moving light source simulates the effect of a rover moving with respect to the sun or vice versa. For the first image in this set, the light source is far to the right of the wheel. The light source is gradually moved from the right side to a point above and behind the camera. It is then gradually moved down and to the left of the wheel, so that in the last image the light source is just above the surface of the terrain, far to the left of the wheel.

Set 6 shows a wheel moving through uneven bentonite clay illuminated by a stationary point source casting sharp shadows. This simulates the most difficult conditions for the algorithm, which would occur when the sun is low in the sky.

4.2. Passive lighting results

Representative results are shown in Figs. 10 through 12. Figure 10 shows a sample image from image set 1 with the algorithm-identified sinkage angles superimposed. The center line shows the direction of $\mathbf{v}_{\text{down}}$. The lines to the left and right run from the center of the wheel to the left and right terrain interfaces. It can be seen that the actual sinkage angles closely match those identified using the algorithm.

Figure 11 plots the actual and visually-measured sinkage as a percentage of the wheel radius for all images in image set 1. The x-axis is the index of the image being analyzed, corresponding to one of the 19 images collected in each image set. The y-axis shows the sinkage as a percentage of the wheel radius. The left and right plots show the sinkage for the left and right sides of the wheel, respectively, for the same images. The sinkage algorithm results match the actual sinkage very accurately.

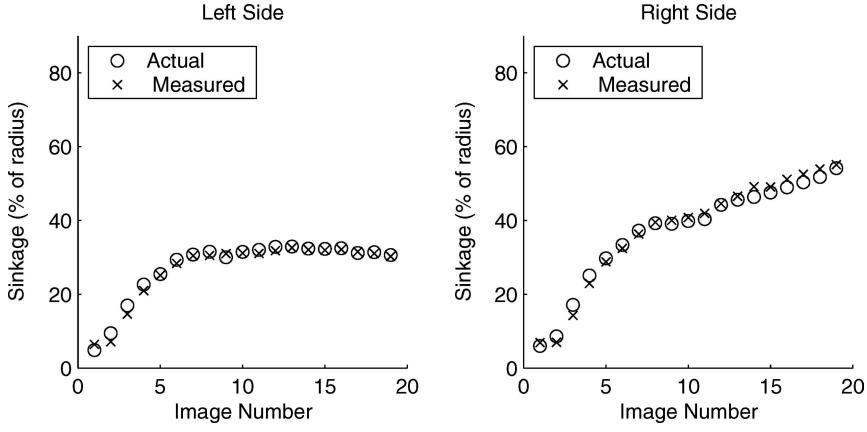

Fig. 11 Actual and visually-measured sinkages for image set 1 (Bentonite, high-slip, flat terrain, fully-lit)

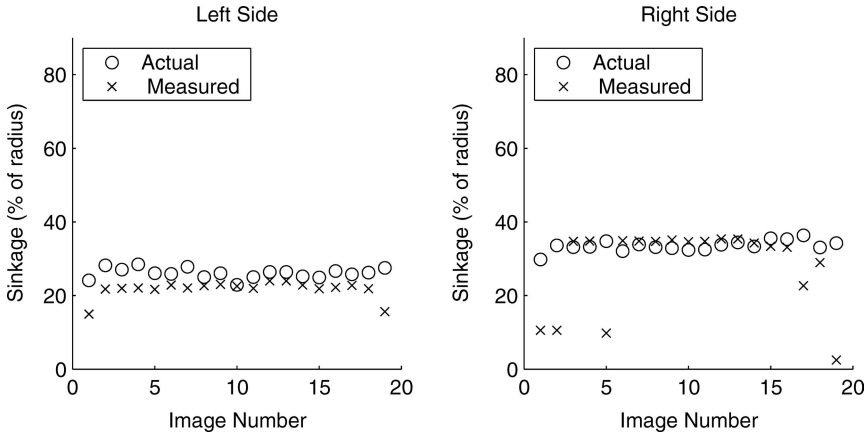

Fig. 12 Actual and visually-measured sinkages for image set 5 (Bentonite, stationary wheel, moving light source)

Figure 12 shows similar results for image set 5. Since the wheel is stationary and not digging into the terrain, a constant sinkage should result even though the image changes due to the moving light source. Note that actual sinkage values vary slightly due to inaccuracies in manually identifying the true sinkage. The visually-measured sinkage is close to the actual sinkage for most of the images. Sources of errors are discussed later in this section.

Table 1 summarizes results of sinkage angle measurement for all six image sets. Error is computed as the difference between the visually-measured sinkage and the actual sinkage, as a percentage of the wheel radius, for the left- and right-side interface angles. Note that the actual sinkage in these data sets varies from 0 to 60% of the wheel radius.

The algorithm detected wheel sinkage under a wide range of conditions with good accuracy. Errors in set 4 were caused by rocks occluding the wheel-terrain interface. While these small errors could be mitigated by a texture- or geometry-based rock detection algorithm, adding such an algorithm would drastically increase the computational requirements.

A more significant error source was uneven lighting. Sets 5 and 6 show substantially higher RMS error than sets 1–4. Reflections on the wheel rim occasionally caused misidentifications. Other problems were the result of shadows falling on uneven terrain itself, as seen in Fig. 11. However, these errors tended to appear as easily-identifiable "outliers" (i.e. as large anomalous changes in the visually-measured angle) that could be mitigated by intelligent filtering.

Table 1 Visual wheel sinkage measurement results

Image set (see Fig. 9)	Left-side angle RMS error (%)	Right-side angle RMS error (%)
1	1.08	1.61
2	2.40	2.46
3	2.33	2.48
4	5.21	2.06
5	5.10	12.10
6	8.85	14.01

4.3. Active lighting results

An alternative method for minimizing errors caused by uneven lighting is to employ active lighting. Figure 13 shows a sample image from a series where a strobe was used to illuminate the wheel-terrain interface (set 7). Here the wheel was driving through topsoil, and the wheel rim was yellow, providing good contrast. The results are plotted in Fig. 14. In shadowy conditions similar to sets 5 and 6, tests using a strobe to illuminate the wheel resulted in RMS errors of less than 2% of the wheel radius.

4.4. Computational requirements

Computational requirements for this algorithm are relatively low. The use of suspension configuration sensors to identify the position of the wheel rim eliminates the need

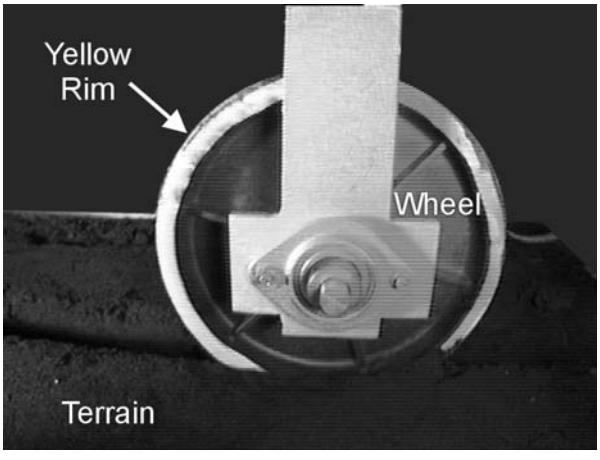

Fig. 13 Sample image from set 7 (active lighting)

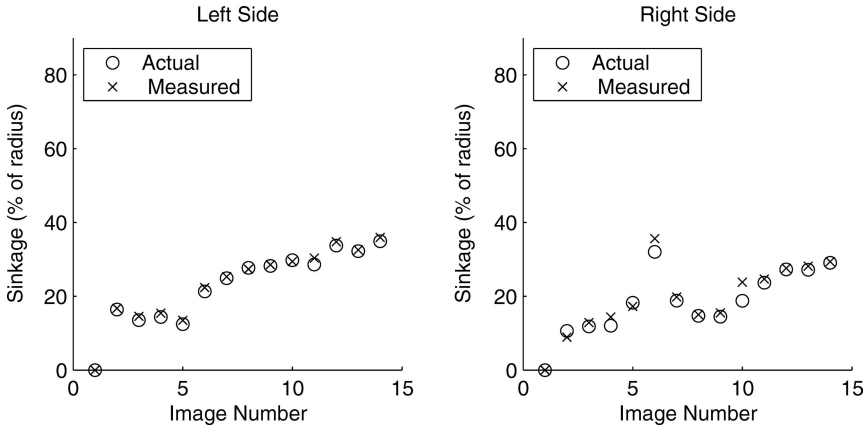

Fig. 14 Actual and visually-measured sinkages for image set 7 (active lighting)

for vision-based detection of the wheel position, which would likely increase computation significantly. A Matlab version of the algorithm processed images at 3 Hz on a Pentium III 933 MHz PC. An optimized compiled implementation would be expected to run significantly faster. For image sets and settings discussed here, approximately 90,000 floating point operations were required per image. A standalone executable version of the code required 80 KB of memory for the program, with an additional 60 KB of memory for execution. These low requirements suggest that the algorithm is suitable for on-board implementation on a planetary rover with limited computational resources.

4.5. Mars exploration rover image analysis

The algorithm was also tested on several images from the Mars Exploration Rovers, manually estimating the coordinate transformation parameters. Figure 15 shows a sample image from the Spirit rover "Hazcam" of the left front wheel.

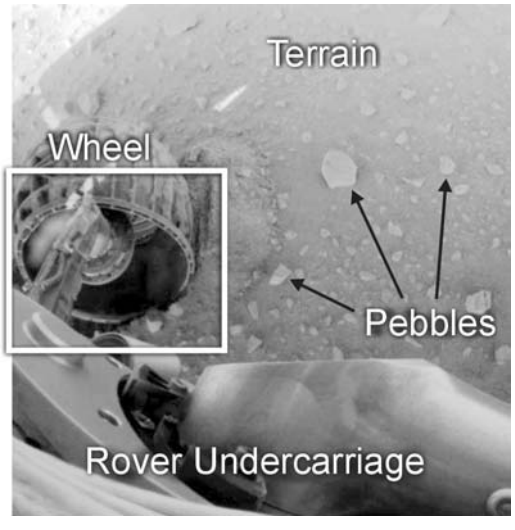

Fig. 15 Actual (cropped) image from Spirit left front Hazcam (NASA/JPL Image)

Fig. 16 Computer identification of sinkage angles in Spirit image (NASA/JPL Image)

Figure 16 shows the computer-identified sinkage angles for this same image. Despite the narrow rim and low contrast between the rim and the terrain, the algorithm appears to give believable results. Actual values for the wheel sinkage were unavailable for this data set. A modified rover wheel with a wider rim painted black or white would make the algorithm more robust to lighting variation.

5. Conclusion

A vision-based method for determining the sinkage of a robot wheel in rigid or deformable terrain has been presented. The method detects the location of the wheel-terrain interface by finding the maximum change in intensity along the wheel rim. The method is computationally efficient and uses a single grayscale vision sensor, making it potentially suitable for systems with limited computational resources such as planetary rovers. Experimental results have shown the method to be accurate and relatively robust to lighting variations. It has also been shown that active lighting can be implemented to further improve measurement accuracy.

Acknowledgment This work was supported by the NASA Jet Propulsion Laboratory (JPL) Mars Technology Program. The authors would like to acknowledge the support of Drs. Richard Volpe and Samad Hayati at JPL.

References

Allen, C., Morris, R., Jager, K., Golden, D., Lindstrom, D., Lindstrom, M., and Lockwood, J. 1997. Martian Regolith Simulant JSC Mars-1. In *Proc. 28th Lunar and Planetary Science Conf.*, Houston, TX.

Bekker, G. 1956. *Theory of Land Locomotion*. University of Michigan Press: Ann Arbor, MI.

Bridges, A. 2004. Opportunity conquers Mars Crater. *The Columbian*, 23 March 2004, A6.

Brooks, C. 2004. Terrain Identification Methods for Planetary Exploration Rovers. Master's Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA.

Cunningham, J., Corke, P., Durrant-Whyte, H., and Dalziel, M. 1999. Automated LHDs and underground haulage trucks. *Australian Journal of Mining*, 51–53.

Forsyth, D. and Ponce, J. 2003. *Computer Vision: A Modern Approach* Prentice Hall: Upper Saddle River, NJ.

Gonthier, Y. and Papadopoulos, E. 1998. On the development of a real-time simulator for an electro-hydraulic forestry machine. In *Proc. IEEE Int. Conf. Robotics and Automation*. Leuven, Belgium, pp. 127–132.

Iagnemma, K. and Dubowsky, S. 2004. *Mobile Robots in Rough Terrain: Estimation, Motion Planning, and Control with Application to Planetary Rovers*. Springer Tracts in Advanced Robotics (STAR) Series, Vol. 12, Springer.

Iagnemma, K., Kang, S., Brooks, C., and Dubowsky, S. 2003. Multi-sensor terrain estimation for planetary rovers. In *Proc. Seventh Int. Symposium on Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS*, Nara, Japan.

Iagnemma, K., Shibly, H., and Dubowsky, S. 2002. On-line terrain parameter estimation for planetary rovers. In *Proc. of the 2002 IEEE Int. Conf. Robotics and Automation*, Washington, DC, pp. 3142–3147.

Kang, S. 2003. Terrain parameter estimation and traversability assessment for mobile robots. Master's Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA.

Le, A., Rye, D., and Durrant-Whyte, H. 1997. Estimation of track-soil interactions for autonomous tracked vehicles. In *Proc. of the IEEE International Conf. on Robotics and Automation,* pp. 1388–1393.

Morris, J. 2004. NASA to attempt to fix opportunity abrasion tool. *Aerospace Daily & Defense Report*, 211(35).

Olson, C., Matthies, L., Schoppers, M., and Maimone, M., 2001. Stereo ego-motion improvements for robust rover navigation. In *Proc. of IEEE International Conference on Robotics and Automation*.

Petit, C. 2004. Roving about the Red Planet. *U.S. News & World Report* 137(8), 13 Sept. 2004, p. 52.

Volpe, R. 2003. Rover functional autonomy development for the mars mobile science laboratory. In *Proc. 2003 IEEE Aerospace Conf.*, Big Sky, MT, Vol. 2, pp. 643–652.

Wilcox, B. 1994. Non-geometric hazard detection for a mars microrover. In *Proc. AIAA Conf. Intelligent Robotics in Field, Factory, Service, and Space*, Houston, TX, Vol. 2, pp. 675–684.

Wong, J. 1976. *Theory of Ground Vehicles*. John Wiley and Sons., New York.

Yoshida, K. and Hamano, H. 2002. Motion dynamics and control of a planetary rover with slip-based traction model. In *Proc. of SPIE—The Int'l Soc. for Optical Engineering*. Vol. 4715, pp. 275–286.

Christopher Brooks is a graduate student in the Mechanical Engineering department of the Massachusetts Institute of Technology. He received his B.S. degree with honor in engineering and applied science from the California Institute of Technology in 2000, and his M.S. degree from the Massachusetts Institute of Technology in 2004. He is a student collaborator on the Mars Exploration Rover science mission. His research interests include mobile robot control, terrain sensing, and their application to improving autonomous robot mobility. He is a member of Tau Beta Pi.

Karl Iagnemma is a research scientist in the Mechanical Engineering department of the Massachusetts Institute of Technology. He received his B.S. degree *summa cum laude* in mechanical engineering from the University of Michigan in 1994, and his M.S. and Ph.D. from the Massachusetts Institute of Technology, where he was a National Science Foundation graduate fellow, in 1997 and 2001, respectively. He has been a visiting researcher at the Jet Propulsion Laboratory. His research interests include rough-terrain mobile robot control and motion planning, robot-terrain interaction, and robotic mobility analysis. He is author of the monograph *Mobile Robots in Rough Terrain: Estimation, Motion Planning, and Control with Application to Planetary Rovers* (Springer, 2004). He is a member of IEEE and Sigma Xi.

Steven Dubowsky received his Bachelor's degree from Rensselaer Polytechnic Institute of Troy, New York in 1963, and his M.S. and Sc.D. degrees from Columbia University in 1964 and 1971. He is currently a Professor of Mechanical Engineering at M.I.T and Director of the Mechanical Engineering Field and Space Robotics Laboratory. He has been a Professor of Engineering and Applied Science at the University of California, Los Angeles, a Visiting Professor at Cambridge University, Cambridge, England, and Visiting Professor at the California Institute of Technology. During the period from 1963 to 1971, he was employed by the Perkin-Elmer Corporation, the General Dynamics Corporation, and the American Electric Power Service Corporation. Dr. Dubowsky's research has included the development of modeling techniques for manipulator flexibility and the development of optimal and self-learning adaptive control procedures for rigid and flexible robotic manipulators. He has authored or co-authored nearly 300 papers in the area of the dynamics, control and design of high performance mechanical and electromechanical systems. Professor Dubowsky is a registered Professional Engineer in the State of California and has served as an advisor to the National Science Foundation, the National Academy of Science/Engineering, the Department of Energy, and the US Army. He is a fellow of the ASME and IEEE and is a member of Sigma Xi and Tau Beta Pi.