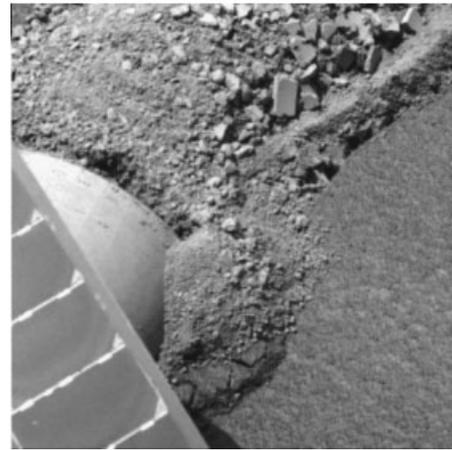
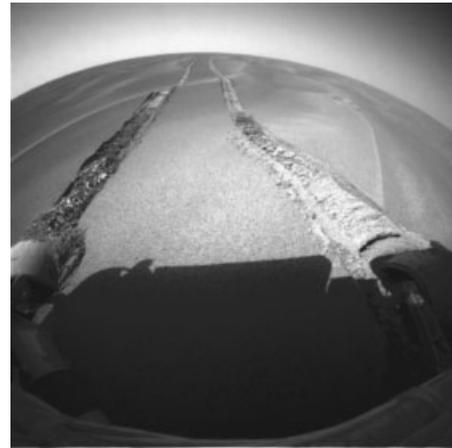




## 1. INTRODUCTION

Slip is a measure of the lack of progress of a wheeled ground robot while driving. High levels of slip can be observed on certain terrains, which can lead to significant slow down of the vehicle, inability to reach its predefined goals, or, in the worst case, getting stuck without the possibility of recovery. Similar problems were experienced in the Mars Exploration Rover (MER) mission in which one of its rovers got trapped in a sand dune, experiencing 100% slip (Figure 1). The science goals of future Mars rover missions will require the rover to explore areas of the planet which feature very steep and rocky terrain, where a lot of slippage is possible. It will be important to be able to predict slip from a distance, so that adequate planning is performed and areas of high slip are avoided.

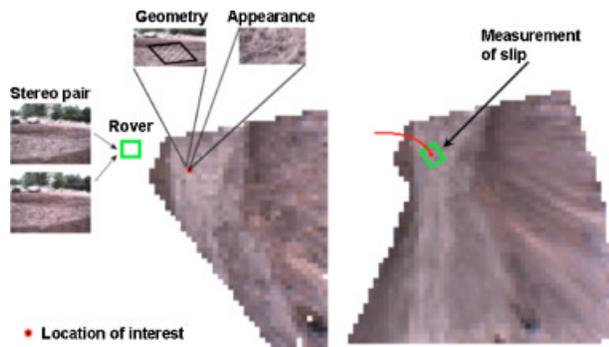
The mobility of a vehicle on off-road terrain is known to be strongly influenced by the interaction between the vehicle and the terrain (Bekker, 1969). Slip is the result of this complex interaction and, second to tip-over hazards, it is the most important factor in traversing slopes (Leger et al., 2005; Biesiadecki et al., 2005). However, with a few exceptions (Ojeda, Borenstein, Witus & Karlsen, 2006; Brooks, Iagnemma & Dubowsky, 2005), slip has not been considered as an aspect of terrain traversability in state-of-the-art autonomous navigation systems so far, mainly because of the highly nonlinear nature of the rover-terrain interactions and the complexity of modeling of these interactions (Andrade, Ben Amar, Bidaud & Chatila, 1998; Iagnemma, Shibly & Dubowsky, 2002). The most commonly used approach is to represent the surrounding terrain as a geometric elevation map, using range data from sensors, such as stereo cameras, radar, or lidar, in which a binary perception of the terrain, i.e., obstacle versus nonobstacle, is done (Daily et al., 1988; Kelly & Stentz, 1998). This idea has been extended to detecting compressible grass and foliage, which would otherwise be perceived as an obstacle (Macedo, Manduchi & Matthies, 2000; Lacaze, Murphy & Delgiorno, 2002; Matthies, Bergh, Castano, Macedo & Manduchi, 2003), but this again uses more or less geometric concepts of penetrability of terrain. Regarding slip, a sandy slope might be nontraversable because of large slip, whereas the same slope covered with different material, e.g., compacted soil, could be perfectly traversable. Such areas of large slip are called *nongeometric obstacles*, as they cannot be detected by software which uses geometrical information only (Leger et al., 2005), and more ad-



**Figure 1.** The Mars Exploration Rover “Opportunity” trapped in the “Purgatory” dune on sol 447. A similar 100% slip condition can lead to mission failure.

vanced perception of the physical terrain properties is needed to detect them.

Visual characteristics of the terrain, in addition to geometry, can give more clues to its mechanical properties and the eventual rover-terrain interaction. Thus, we propose to use stereo pair imagery as the input for slip prediction (Angelova, Matthies, Helmick & Perona, 2006; Angelova, Matthies, Helmick, Sibley & Perona, 2006). The rationale behind this approach is that, from a mechanical point of view, slip depends on physical and geometrical properties of the terrain (Bekker, 1969), and stereo imagery provides information about both the geometry from the range data and the visual appearance of the terrain. So, stereo imagery contains much information which



**Figure 2.** Main idea: learning of mechanical (slip) output from visual information. The rover collects visual information (appearance and geometry) about a future location of interest in the forward looking map from its stereo pair images (left). When this location is reached by the rover, a slip measurement is taken using onboard sensors (right). Correlating vision information to the corresponding slip measurement and learning this mapping allows for prediction of slip from a distance using visual information only.

can help predict slip on the forthcoming terrain. The main challenge is how to interpret the vision data to infer properties about the terrain or predict slip.

Our approach to this problem is to correlate the visual information and the corresponding measured slip while the rover is traversing the terrain. In particular, we extract information about the terrain observed from a distance by using information from a stereo pair only, measure the slip of the rover when it traverses this particular region, and create a mapping between visual information and the resultant slip (Figure 2). We propose to *learn* this functional relationship using the experience from previous traversals (Angelova, Matthies, Helmick & Perona, 2006; Angelova, Matthies, Helmick, Sibley & Perona, 2006). Thus, after learning, the expected slip can be predicted from a distance using only stereo imagery as input. More importantly, the rover's own sensors are used as feedback to the learning, which can remove the human-in-the-loop factor for data labeling. A learning approach is chosen, because: (1) creating a physical slip model is extremely complicated due to the large number of variables involved, (2) the mapping from visual input to a mechanical terrain property, such as slip, is a complex function, which does not have a known analytical form or a physical model and one possible way to observe it and learn about it

is via training examples, and (3) a learning approach promotes adaptability of the vehicle's behavior. A similar idea of learning, using visual and other sensory input, has been previously applied by Wellington & Stentz (2004) for retrieving the load-bearing surface in vegetation and stems from ideas for learning permeability of vegetation (Macedo et al., 2000; Matthies et al., 2003).

To address the problem of slip learning and prediction we propose a general framework in which the task is subdivided into: (1) learning the terrain type from visual appearance and then, after the terrain type is known (2) learning slip from the terrain geometry using nonlinear approximation (Angelova, Matthies, Helmick & Perona, 2006; Angelova, Matthies, Helmick, Sibley & Perona, 2006). We term the latter dependence of slip on terrain geometry, when the terrain type is known, *slip behavior*. The proposed decomposition of the problem is adequate because from a mechanical point of view it is known that different terrains exhibit different slip behavior characteristics (Bekker, 1969; Terzaghi, 1948), and because terrain appearance can be considered approximately independent of terrain geometry. This decomposition also introduces some structure in the problem, so that we can solve it with a reasonable amount of training data.

This work has been the first to attempt predicting slip from a distance. We have proposed an overall solution framework in which the slip is learned and predicted from visual information. This paper extends the results of (Angelova, Matthies, Helmick & Perona, 2006; Angelova, Matthies, Helmick, Sibley & Perona, 2006).

### 1.1. Testbed

This research is targeted for planetary rovers, such as the Mars Exploration Rover (Figure 3, left). A Mars research rover testbed, called Rocky8 (Figure 3, middle), is used for experimental purposes instead. We also used a LAGR robot<sup>1</sup> (Figure 3, right), as it is a more convenient data collection platform.

Rocky8 is a prototype research rover with six wheels in a rocker-bogie configuration (Tarokh, MacDermott, Hayati & Hung, 1999) which allows for improved mobility on rough terrain. We have used its hazard stereo cameras with 80° field of view (FOV),

<sup>1</sup>LAGR stands for Learning Applied to Ground Robots and is an experimental all-terrain vehicle program funded by DARPA.



**Figure 3.** The Mars Exploration Rover “Spirit” in the Jet Propulsion Laboratory (JPL) Spacecraft Assembly Facility (left). Rocky8 rover in the Mojave desert (middle). LAGR robot on off-road terrain (right).

wheel encoders, rocker and bogie angle sensors, and Inertial Measurement Unit (IMU). The rover’s nominal speed of operation is 8 cm/s. In the dataset used in the paper, stereo pair imagery is acquired after each stop of the robot.

The LAGR robot has two front differential drive wheels and two rear caster wheels. It is about 1 m tall. It is equipped with stereo cameras with 70° FOV, wheel encoders, IMU, and Global Positioning System (GPS). The IMU and GPS are processed into a *global pose*. The robot can run in autonomous mode or be manually joysticked using a radio controller. It can achieve speeds of up to 1.2 m/s, although for some of our experiments it was set to drive at 30 cm/s. Stereo imagery is acquired continuously at 5 Hz.

## 1.2. Outline

The rest of the paper is organized as follows. We start by providing a definition of slip (Section 2) and by reviewing related previous work (Section 3). Section 4 introduces the datasets we are using. Section 5 describes the proposed framework in which learning and prediction of slip is decomposed into terrain class recognition (Section 6) and nonlinear regression modeling for learning of slip behavior for each terrain type (Section 7). Both Sections 6 and 7 provide experimental evaluations of the individual components. Section 8 provides experimental results of the final slip prediction, when using remote sensors as the only input. Section 9 concludes the paper and gives directions for future work.

## 2. DEFINITION OF SLIP

Slip  $s$  is defined as the difference between the velocity measured by the wheel ( $wr$ ) and the actual velocity  $v$ :

$s = wr - v$ , where  $w$  is angular wheel velocity and  $r$  is the wheel radius (Wong, 1993). It can also be normalized by the commanded wheel velocity:  $s = (wr - v) / wr$  (Bauer, Leung & Barfoot, 2005; Iagnemma et al., 2002; Wong, 1993). Similarly, the slip for the whole rover is defined as the difference between the actual vehicle velocity and the velocity estimated from the kinematic model for each degree of freedom (DOF) of the rover per step (i.e., between two consecutive stereo pairs) (Helmick, Cheng, Roumeliotis, Clouse & Matthies, 2004). It can also be normalized, to receive a unitless slip value or express it in percentage of the step size. In this paper we use the normalized version of slip for the whole rover.

For the kinematic estimate, we use a differential drive model for the LAGR vehicle and a full rocker-bogie kinematic model for Rocky8 (Tarokh et al., 1999; Helmick et al., 2004). The actual position (ground truth) can be estimated by visually tracking features (Matthies & Schafer, 1987; Matthies, 1989), a method now called visual odometry (VO), or measured with some global position estimation device. VO is the preferred method for ground truth estimation because it is a convenient, self-contained sensor on the vehicle. By using VO, data collection and training can be done automatically, onboard the rover, which coincides with the goals of planetary exploration missions. Furthermore, global positioning devices are not always available, especially in planetary missions.

Validation of VO position estimation has been performed by several groups (Olson, Matthies, Shoppers & Maimone, 2001; Helmick et al., 2004; Nister, Naroditsky & Bergen, 2006). VO position estimation error has been measured to be less than 2.5% of the distance traveled, compared to ground truth surveyed with a Total Station that has 2 mm precision,

for runs of 20–30 m in outdoors testing with the Rocky8 rover (Helmick et al., 2004). Similar results of 1.2% position error for a 20 m traverse have been achieved by Olson et al. (2001) while testing in different circumstances, i.e., using a smaller robot, wide field of view cameras, different image resolution, etc. VO path length errors of about 1%–1.6% for 180–380 m traverses in outdoor environments have been reported by Nister et al. (2006) with a different VO algorithm. The results of these tests indicate that VO is a precise position estimation technique and is adequate to be used as ground truth both for computing slip per step and for precise localization within short to midsize (20 m) traverses, i.e., to be able to map correctly the position of the location seen from a distance to the location traversed later on.

We measure slip with respect to the previous rover frame (corresponding to the beginning of the step) which is defined as follows: the  $X$  coordinate is along the direction of forward motion,  $Y$  is along the wheel axis, and  $Z$  is pointing down. We define slip in  $X$  and slip in  $Y$  as the components of slip along the  $X$  and the  $Y$  axes, respectively. Slip in yaw is the rotation angle around the  $Z$  axis. Note that the LAGR vehicle has only three kinematically observable DOFs, while Rocky8 has five (Tarokh et al., 1999; Helmick et al., 2004). Slip is normalized by the commanded velocity in  $X$  and will be expressed in percent. There will be cases in which the commanded forward velocity is 0, e.g., a purely crabbing motion for the Rocky8 rover, which will make the slip value undefined. As those cases are rare, we remove those steps from our dataset.

We have adopted a macrolevel (of the whole rover) modeling of slip, in the spirit of Helmick et al. (2004) and Lindemann & Voorhees (2005). More specifically, our assumptions are that, between two consecutive steps, the rover will be traversing approximately locally planar and homogeneous regions, and the weight distribution on all its wheels will be the same. These assumptions mean that we consider slip (i.e., predict the terrain type, estimate terrain slopes, etc.) in regions comparable in size to the size of the robot or its wheel and not at the pixel level, for example. Naturally, those assumptions are violated in our field test data, which is taken on real-life terrains with all complications, such as uneven and nonhomogeneous terrain, clumps on the ground, or rocks in front of the wheels. For example, when one of the wheels traverses a rock, an unexpected slip in yaw might occur, because the rock creates different trac-

tion compared to the soil or can serve as an additional external force to the vehicle. As similar events are not modeled by our system, there will be some sources of sometimes significant noise in the slip measurements in our data. Nevertheless, this macrolevel modeling is justified, as the slip prediction is intended to be used in a first, quick evaluation of terrain traversability to be handed down to a planner. More complex mechanical slip modeling can be applied (Kraus, Fredriksson & Kumar, 1997; Jain et al., 2003; Ishigami, Miwa, Nagatani & Yoshida, 2006), but to predict slip, information about soil mechanical properties of the forthcoming terrain is still required and will have to be learned. These approaches may deal better with uneven terrain, e.g., if dynamic simulation of the traverse over detailed terrain elevation models is performed (Jain et al., 2003), but they will be considerably more computationally expensive.

Slip also depends on the commanded velocity, although for the relatively small speeds of the Mars rovers, velocity is not a significant factor. We have factored it out by averaging consecutive steps, by driving at approximately constant velocity, or by normalizing slip stepwise by the commanded velocity.

### 3. PREVIOUS WORK

Although early work in autonomous navigation and traversability analysis based on forward looking sensors did not use learning (Daily et al., 1988; Kelly & Stentz, 1998; Goldberg, Maimone & Matthies, 2002), learning-based approaches have started to become more and more preferred (Pomerleau, 1989; Bellutta, Manduchi, Matthies, Owens & Rankin, 2000; Matthies et al., 2003; Vandapel, Huber, Kapuria & Hebert, 2004; Wellington, Courville & Stentz, 2005). The reason for that is that intelligent autonomous behavior needs to be adaptive to the environment and the more complex the environment is, the less likely it is that predefined rules or heuristics will work well. This is particularly true for outdoor, off-road, unstructured environments which offer a lot of challenges, e.g., variability in terrains and lighting conditions, lack of structure, lack of prior information, etc., and in which learning approaches have proved to be more appropriate (Seraji, 2000; Howard, Tunstel, Edwards & Carlson, 2001; Dima, Vandapel & Hebert, 2004; Vandapel et al., 2004; Wellington & Stentz, 2004; LeCun, Muller, Ben, Cosatto & Flepp, 2005; Lieb, Lookingbill & Thrun, 2005; Wellington et al., 2005).

Related work on vision-based perception of the forthcoming terrain has been considered for the purposes of determining the mobility of Mars rovers (Howard et al., 2001) or the traversability in tall grass for agricultural vehicles (Wellington and Stentz, 2004), for detecting the drivable rural road in the context of off-road autonomous navigation (Rasmussen, 2001), or for detecting obstacles in indoor (Ulrich & Nourbakhsh, 2000) and outdoor environments (Batavia & Singh, 2001; Kim, Sun, Oh, Rehg & Bobick, 2006). In most of the abovementioned cases the task at hand is to determine a binary output value, “traversable” or “not traversable,” or at most, several traversability levels (Seraji, 2000). In the proposed work we predict a nonlinear behavior rather than a single binary value, using visual information, similar to Wellington & Stentz (2004). Moreover, we have proposed to learn this behavior, instead of adopting a known physical model, because such a model might be hard or impractical to obtain, as is the case with slip for which significant experimentation is required to adjust the parameters related to soil behavior and vehicle-terrain interaction (Bekker, 1969; Wong, 1993).

From a mechanical point of view, modeling and estimation of slip has been done at various levels of complexity and for various vehicle architectures (Bekker, 1969; Wong, 1993; Kraus et al., 1997; Le, Rye & Durrant-Whyte, 1997; Andrade et al., 1998; Farritor et al., 1998). These methods are rather complicated and need to be performed at the traversed location, as they require local sensor measurements and detailed knowledge of terrain geometry. They are computationally intensive and impractical in the present setup. As slip depends also on the mechanical soil characteristics (Terzaghi, 1948; Bekker, 1969), additional estimation of soil parameters, such as cohesion and friction angle (Le et al., 1997; Iagnemma et al., 2002), or modeling of the soil behavior (Andrade et al., 1998) needs to be done. Methods for online terrain parameter estimation (Iagnemma et al., 2002), for recognizing terrain types (Brooks et al., 2005), and for characterizing terrain trafficability (Ojeda et al., 2006) from onboard mechanical sensors have been proposed, but these estimates apply to the present vehicle location. No method, to our best knowledge, is available for predicting terrain parameters from a distance. One way to address this problem is by using forward looking sensors, e.g., vision, as proposed in this paper.

Although slip has been acknowledged as an om-

nipresent problem in localization, especially in rough-terrain mobility (Hoffman, Baumgartner, Huntsberger & Schenker, 1999), very few authors have considered counteracting slip for improving vehicle mobility. Among them are the slip compensation algorithm of Helmick et al. (2004, 2005), in which the slip, measured at a particular step, is taken into account to adjust the next step, compensating for the distance which was not traversed; or the algorithm for improving traction control, proposed by Iagnemma & Dubowsky (2004). However, those methods, again, work at the traversed rover location and do not allow for planning at a distance, which our method enables.

Previous approaches have used manually created functions of slip as dependent on slopes (Lindemann & Voorhees, 2005). Slip measurements were performed on short traverses of the rover on a tilt-table platform set to varying slope angles. These results showed that slip is a very nonlinear function of terrain slopes. For example, in deep sand, slip of about 20% on a 10° slope and of about 91% on a 20° slope was measured, when the rover was driving straight upslope. The results of these experiments have been used successfully to teleoperate the Opportunity rover out of Eagle Crater, but the approach is very labor intensive, as it requires manual measurements. It also needs careful selection of the soil type on which the tests are performed to match the target Mars soil. Another limitation is that no slip models were available for angles of attack different from 0°, 45°, or 90° from the gradient of the terrain slope (Cheng, Maimone & Matthies, 2005). The results are also specific to the vehicle. For example, a small design modification in the pattern of the wheels can change the slip behavior (Bauer et al., 2005), affecting a potential physical model. We believe that learning slip is a more general approach, namely, the same learning algorithm can be applied to another vehicle to learn its particular behavior on different terrains. Moreover, the proposed method enables the vehicle to apply the learned models dependent on what it has sensed from the environment.

The work described above concerns estimating slip from mechanical measurements, or, in our case, visual information. Conversely, slip measurements have been used to infer mechanical terrain parameters on the Mars Pathfinder Mission in a controlled one-wheel soil-mechanics experiment (Moore et al., 1999). Similar experiments have been done by Arvidson et al. (2004) for MER. This gives us the assertion

that slip characteristics are directly correlated to terrain mechanical properties and the intuition that if the terrain soil type could be correctly recognized (which would entail its mechanical properties) then slip behavior is predictable.

#### 4. DATASET

For our slip prediction experiments we have collected several datasets on off-road terrains with the LAGR vehicle. There are five major terrain types which the rover has traversed: soil, sand, gravel, asphalt, and woodchips (Figure 4). We focused mainly on terrains which are significant for planetary exploration and provided the latter two for comparison only. In addition to that, there are several other terrain types which appear in the sequences, such as green or dry grass, which we considered as a single “grass” class in the terrain classification in Section 6. The terrains contain irregularities, undulations of the surface, small rocks, and grass clumps for off-road terrains or discolorations for asphalt. Although we have good variability in the terrain relief in our dataset (level, upslope, and downslope areas on soil, asphalt and woodchip terrains, transverse slope on gravelly terrain, flat sandy terrain, etc.), not all possible slip behaviors could be observed in the area of data collection. For example: there was no sloped terrain covered with sand, besides, the LAGR robot showed poor mobility on flat sand, i.e., about 80% slip (Angelova, Matthies, Helmick, Sibley & Perona, 2006); the gravelly terrain available was only possible to be traversed sideways for safety reasons; there was no transverse slope for the soil or asphalt datasets. We have collected a total of ~5000 frames which are split approximately into 3000 for training and 2000 for testing. The distance covered by the rover during the data collection is roughly about 1 km. These data have been used extensively for testing in Sections 6–8.

A second smaller dataset was collected with the Rocky8 rover in the Mojave desert (Figure 13, later in this paper). It covers a distance of about 30 m. As there was a single terrain type in the course of the traverse (desert sand) we will be using it for evaluation of the slip prediction without the terrain recognition module (Section 7.3.5).

### 5. GENERAL FRAMEWORK FOR SLIP LEARNING AND PREDICTION

In this section we propose a general framework to learn the functional relationship between visual information and the measured slip using training examples.

The amount of slippage for a given vehicle depends on the soil type and the terrain’s geometry (Bekker, 1969), so both geometry  $G$ , captured by the terrain’s slopes, and appearance  $A$ , e.g., texture and color, must be considered. At training time, the information about appearance and geometry coming from the stereo imagery is correlated to the measured slip (in  $X$ ,  $Y$ , or yaw) as the robot traverses the cell. At query time, geometry and appearance alone are used to predict slip.

#### 5.1. General Framework

The dependence of slip on terrain slopes, called earlier slip behavior, is known to be highly nonlinear (Lindemann & Voorhees, 2005), but the precise relationship varies with the terrain type (Bekker, 1969). So, we cast the problem into a framework similar to the mixture of experts (MOE) framework (Jacobs, Jordan, Nowlan & Hinton, 1991), in which the input space is partitioned into subregions, corresponding to different terrain types, and then several functions, corresponding to different slip behaviors, are learned for each subregion. That is, in each region one model of slip behavior would be active, i.e., when the terrain type is known, slip will be a function of terrain geometry only.

More formally, let  $I$  be all the information available from stereo pair images,  $I=(A,G)$ . Let  $f(S|I)$  be the regression function of slip  $S$  ( $S$  can be any of the slip in  $X$ ,  $Y$ , or yaw) on the input variables  $A,G$  (used interchangeably with the image information  $I$ ). Now, considering that we have several options for a terrain type  $T$ , each one occurring with probability  $P(T|A,G)$ , given the information from the image in question  $A,G$ , we can write  $f(S|I)$  as follows:

$$f(S|I) = f(S|A,G) = \sum_T P(T|A,G) f(S|T,A,G),$$



**Figure 4.** Example images from some of the terrains collected by the LAGR vehicle: sand, soil, gravel, woodchips, and asphalt.

where  $\sum_T P(T|A, G) = 1$ . This modeling admits one exclusive terrain type to be selected per image, or a soft partitioning of the space, which allows for uncertainty in the terrain classification. We assume that the terrain type is independent of terrain geometry  $P(T|A, G) = P(T|A)$  and that, given the terrain type,

slip is independent of appearance  $f(S|T, A, G) = f(S|T, G)$ . Assuming independence of appearance and geometry is quite reasonable because, for example, a sandy terrain in front of the rover will appear approximately the same, no matter if the rover is traversing a level or tilted surface. So we get

$$f(S|I) = \sum_T P(T|A) f(S|T, G).$$

In summary, we divide the slip learning problem into a terrain recognition part [ $P(T|A)$ , i.e., the probability of a terrain type, given some appearance information] and a slip prediction part [ $f(S|T, G)$ , i.e., the dependence of slip on terrain geometry, given a fixed terrain type  $T$ ]. For simplicity, in this paper, instead of the mixing coefficients  $P(T|A)$ , we use a single winner-take-all terrain classification output  $T(A) = \operatorname{argmax}_T P(T|A)$ , which will be learned and predicted by a terrain classifier (Section 6). The regression functions  $f_T(S|G) = f(S|T, G)$  for different terrain types  $T$  will be learned and predicted by a nonlinear regression method (Section 7). More precisely, suppose we are given training data  $D = \{(\mathbf{x}_i, \mathbf{y}_i), S_i\}_{i=1}^N$ , where  $\mathbf{x}_i$  is the  $i$ th appearance input vector,  $\mathbf{y}_i$  is the  $i$ th geometry input vector,  $S_i$  is the corresponding slip measurement, and  $N$  is the number of training examples ( $\mathbf{x}$ ,  $\mathbf{y}$  are particular representations of the appearance  $A$  and geometry  $G$  information in the image, respectively). We will train, independently, a texture classifier  $T(\mathbf{x})$  to determine the terrain type, using the appearance information  $\mathbf{x}$  in Section 6 and a nonlinear function approximation  $S_T(\mathbf{y}) = f_T(S|G = \mathbf{y})$  for a particular terrain type  $T$  in Section 7. When doing testing, we will use the full input vector  $(\mathbf{x}, \mathbf{y})$ , recognize the terrain type  $T_0 = T(\mathbf{x})$ , and then predict slip, as a function of slopes, from the slip behavior function  $S_{T_0}(\mathbf{y})$ , learned for the terrain  $T_0$ .

We believe this approach is adequate for our slip prediction problem because terrain types do not represent a continuum in appearance space and, in general, would form separate regions in the input space (experts). However, several experts might need to be active at the same time, to make smooth transitions in borderline terrain cases. Both cases are naturally incorporated in the MOE framework. The alternative to introducing structure in the problem is pooling appearance and geometry features, which will not only make the problem more complex, because of increased dimensionality, but will also require a formidable amount of training data. Moreover, this framework is quite general and, in principle, allows for different ways of addressing the problems of learning to recognize terrain types from appearance and different algorithms for learning of slip behavior from terrain geometry.

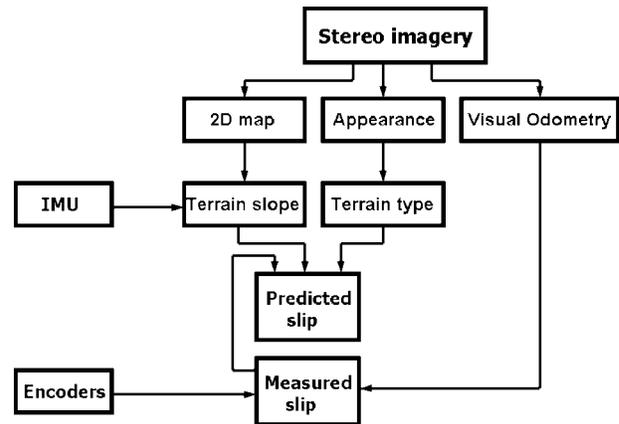


Figure 5. Slip learning and prediction algorithm framework.

## 5.2. Architecture

In this section we briefly describe the architecture of our system, summarized in Figure 5. We will be using the stereo imagery as input, as well as the IMU of the vehicle and its wheel encoders (the latter is needed only for training). Stereo imagery is used to create a two-dimensional (2D) cell map of the environment from its range data. It also provides appearance information for each cell in the map. The 2D map contains geometry information about the terrain ( $G$ ) and, as we are interested in terrain slopes with respect to gravity, we use the vehicle's IMU to retrieve an initial gravity leveled pose. In fact, both LAGR and Rocky8 rovers use a filtered IMU signal, taking into consideration other sensors as well, e.g. the LAGR robot uses GPS and Rocky8 uses VO based pose estimation. The appearance information from color imagery ( $A$ ) will be used to decide which terrain type corresponds to a cell or a neighborhood of cells. This is all the information necessary to perform slip prediction with our algorithm.

In order to learn slip we have added slip feedback. The mechanism to measure slip is as follows. The actual motion between two frames is estimated by VO, which only needs two consecutive stereo pairs as input (Matthies & Schafer, 1987). The motion which the vehicle thinks it has performed is given by the vehicle's forward kinematics. As the LAGR vehicle has a differential drive model the wheel encoders are sufficient to compute its full kinematics. A more complex kinematic model, which needs addi-

tional angle sensors, is needed for a rocker-bogie type of vehicle, such as Rocky8 or MER, but it is well understood how to compute it (Tarokh et al., 1999; Helmick et al., 2004). Differencing the actual motion and the motion estimated by the kinematic model gives a measurement of slip for a particular step. This feedback is used for collecting training examples to learn slip from stereo imagery.

Note that the slip prediction coming from appearance and geometry information is based on frames which observe terrain at a distance, i.e., those stereo frames will come earlier in time than the frames which measure the slip feedback (using VO). The stereo input for both is denoted by a single box in Figure 5, as both types of information come from a single stereo imagery sensor. The advantage of such a system is that it can sense the terrain remotely and that it needs only passive, cheap and self-contained sensors on the vehicle, such as stereo vision.

## 6. TERRAIN CLASSIFICATION

This section describes the terrain classification [ $T(A)$ ] using vision information, which is the first step of our algorithm. For the purposes of slip prediction, we consider only the part of the image plane which corresponds to the robot's 2D map of the environment. That is, for now, we are not interested in regions beyond the distance where range data is available, because we simply cannot retrieve any reliable slope information and therefore cannot predict slip. A reasonable map for the LAGR vehicle is of size  $12 \times 12$  m or  $15 \times 15$  m, centered on the vehicle. Note that the MER panoramic camera has considerably higher resolution and look-ahead (Bell et al., 2003). The map is subdivided into cells, each one of size  $0.4 \times 0.4$  m. Our goal is to determine the terrain type in each cell of the map. In fact, we will be classifying the patches corresponding to the projections of map cells to the image plane.

Note that the patches at close range and at far range have considerably different appearances, so a single texture based classifier could not be used for both. This is due to the fact that the spatial resolution decreases rapidly with range. This could also be clarified by looking at the amount of information in the image plane which corresponds to different areas in the 2D map. For the LAGR vehicle the estimates are: about 70% of the image plane is mapped to ranges

below 10 m, about 7%—to ranges between 10 and 50 m, and about 2%—to ranges between 50 m and the horizon (Matthies et al., 2005). So, for our experiments we build five independent classifiers which are active at different ranges (ranges up to 2, 2–3, 3–4, 4–5, and 5 m and above).

### 6.1. Terrain Classification Algorithm

As we are interested in classifying patches, corresponding to map cells, the approach we use considers the common occurrence of texture elements, called "textons," in a patch. This representation is appropriate, because a texture is defined not by a single pixel neighborhood, but rather by the co-occurrence of visual patterns in larger regions. The idea follows the texton-based texture recognition methods proposed by Leung & Malik (2001) and Varma & Zisserman (2003, 2005). The approach is summarized in Figure 6.

Five different texture classifiers are trained, each one specialized at different range. For each classifier and for each terrain type class (we have six terrain classes), a set of patches in the image plane, corresponding to the map cells at the appropriate ranges, are collected. All the training patches belonging to some range are processed by extracting a set of  $5 \times 5$  RGB regions forming a 75-dimensional vector representation of a local pixel neighborhood. Those vectors are clustered with the  $k$ -means algorithm and the cluster centers are defined to be the textons for this class. We extracted  $k=30$  textons per class. As a result, a total of 180 textons, called "texton dictionary," are collected for the whole training set. Working in a feature space composed of local neighborhoods allows for building statistics of dependencies among neighboring pixels, which is a very viable approach, as shown by Varma & Zisserman (2005).

Now that the dictionary for the dataset has been defined, each texture patch is represented as the frequencies of occurrences of each texton within it, i.e., a histogram.<sup>2</sup> In other words, the patches from the training set are transformed into 180-dimensional vectors, each dimension giving the frequency of occurrence of the corresponding texton in this patch. All vectors are stored in a database to be used later for classification. Similarly, during classification, a

<sup>2</sup>Instead of searching for each texton within a patch individually, each pixel location of the patch is assigned to the texton closest in Euclidean distance.

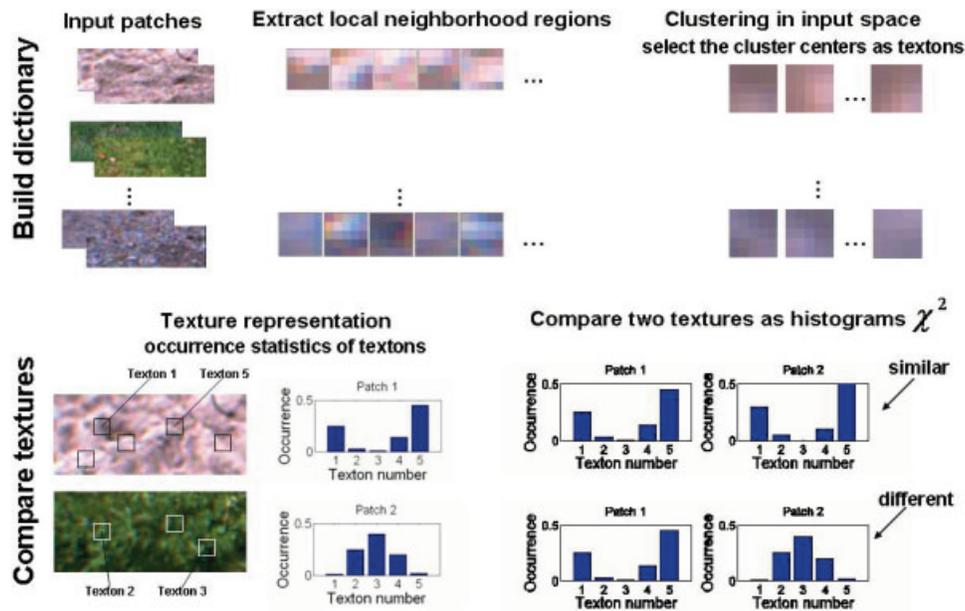


Figure 6. Schematic of the terrain classification algorithm (Leung & Malik, 2001; Varma & Zisserman, 2005).

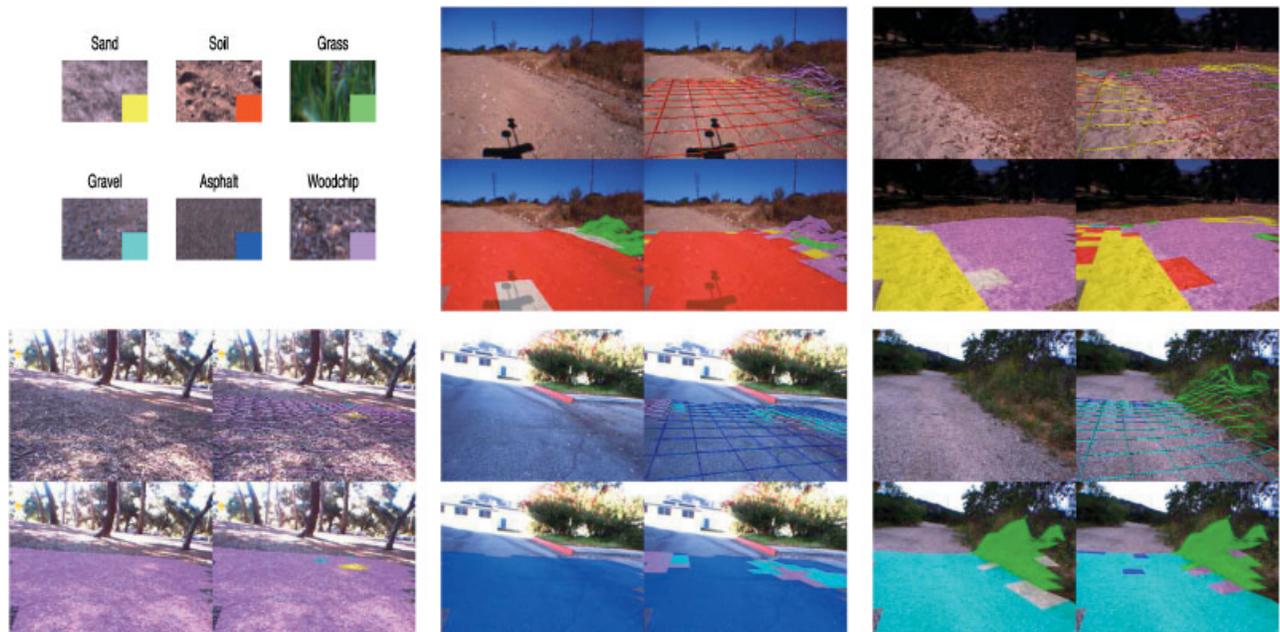
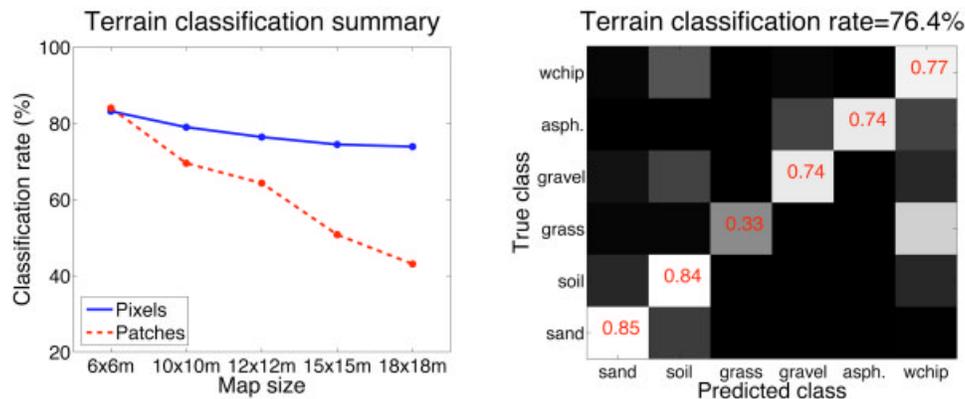


Figure 7. Example texture classification results from each of the datasets. Patches from the six terrain types considered in the texture classification and the corresponding color coding assigned are shown at top left. Each composite image contains the original image (top left), the ground truth terrain classification (bottom left), and the results of the terrain classification algorithm represented in two different ways (top right and bottom right). Ambiguous terrain type in the ground truth is marked with white. Those regions are not required to be classified correctly.



**Figure 8.** Terrain classification results for different map sizes (left). Different ways of representing the classification rate by counting correctly classified patches or pixels are shown. Confusion matrix for the  $12 \times 12$  m map (right). The classification rate for each class is displayed on the diagonal.

query image is transformed into a 180-dimensional vector, i.e., a texon occurrence histogram, and compared to the histogram representations of the examples in the database, using a nearest neighbor method and a  $\chi^2$ -based distance measure (Varma & Zisserman, 2005). The majority vote of  $N=7$  neighbors is taken as the predicted terrain class of the query patch. The result of the classifier will be one single class. To determine the terrain type in the region the robot will traverse (Section 8) we select the winner-take-all patch class label in the  $4 \times 4$  cell neighborhood region. In both decisions, a probabilistic response, rather than choosing a single class, would be more robust. Addressing more advanced probabilistic inference within a patch and among neighboring patches is a subject of our future work.

## 6.2. Terrain Classification Results

In this section we report results of the terrain classification algorithm. Our dataset is composed of five different image sequences which are called soil, sand, gravel, asphalt, and woodchip after the prevailing terrain type in them (Figure 4), but an additional “grass” class can appear in those sequences. As mentioned earlier, we consider patches in the original color image that correspond to cells of the map. Each patch is classified into a particular terrain type and all the pixels which belong to this patch are labeled with the label of the patch (Figure 7). To measure the test performance we take  $\sim 30$  frames in

each sequence, which are separated by at least ten frames within the sequence, so as not to consider images similar to one another. The test set contains a total of  $\sim 150$  frames which span  $\sim 1500$  frames. The ground truth terrain type in the test set is given by a human operator. Example classification results are shown in Figure 7.

Summary results of the terrain classifier for the five sequences for different look-ahead distances are given in Figure 8. Classification performance is measured as the percent of correctly classified area (i.e., number of pixels) in the image plane and the correctly classified patches corresponding to cells in the map. The drop-off in performance, especially in terms of patches, is due to a large number of classification errors at far range. This is expected, as the patches at far range correspond to very small image area (with little information content) and therefore are much more likely to be misclassified. Naturally, regarding slip prediction, a larger map is preferred, as it allows the robot to see farther, but the terrain classification errors at far ranges can make slip prediction unreliable at large distances. Therefore, a tradeoff between accuracy of classification and being able to see farther must be made. To be concrete, in our further experiments we fix the map size at  $12 \times 12$  m. The confusion matrix<sup>3</sup> for the terrain classification for the  $12 \times 12$  m map is shown in Figure 8.

<sup>3</sup>The confusion matrix shows what percentage of the test examples belonging to a class have been classified as belonging to any of the available classes. Its diagonal shows the correct classi-

From it we can see that grass is often misclassified as woodchips [this happens for the areas of dry grass (Figure 7, top middle)], soil is sometimes misclassified as sand and vice versa, asphalt is misclassified as gravel, etc.

### 6.3. Discussion

The texton-based algorithm has been previously applied to artificial images (Varma & Zisserman, 2005), but not to the autonomous navigation domain. Our main motivation for using it here is that slip prediction requires fine discrimination between visually similar terrains, such as soil, sand, and gravel. Previous approaches for terrain classification in the context of autonomous navigation focus on recognizing terrain types, such as road, grass, and sky, which are much easier to discriminate using only color (Manduchi, 1999; Bellutta et al., 2000; Rassmussen, 2001). The texton-based approach is also robust to intraclass variability, often observed in natural terrains.

Although the algorithm is relatively fast ( $\sim 4.7$  s per frame for a  $512 \times 384$  pixels image and a  $12 \times 12$  m map on a 3.6 GHz P4 processor), some additional improvements to decrease the computational time are needed, to be able to work onboard the vehicle in real time. Another limitation of the algorithm, which could be addressed in future work, is that currently the training of the texton approach does not work in an online, incremental fashion. This is mainly due to the clustering procedure during texton selection and to the fixed length histogram representation.

## 7. LEARNING SLIP BEHAVIOR ON A FIXED TERRAIN

In this section we describe the method for learning to predict slip as a function of terrain geometry, when the terrain type is known, i.e., the slip behavior.

The input for slip prediction, i.e., the terrain geometry  $G$ , will be represented by the longitudinal and lateral slopes which are the terrain slopes decomposed along the  $X$  and  $Y$  axes of the current position of the robot, respectively. They are named pitch and roll angles, as they correspond to the vehicle's pitch

and roll, but they are retrieved from stereo imagery. The terrain slopes are estimated as described in Section 7.2, see also Gennery (1999).

and roll, but they are retrieved from stereo imagery. The terrain slopes are estimated as described in Section 7.2, see also Gennery (1999).

### 7.1. Learning Algorithm

We consider the problem of learning of slip behavior as a nonlinear function approximation. That is, the slip  $S_T(\mathbf{y})$ , i.e.,  $f_T(S|G=\mathbf{y})$ , is approximated by a nonlinear function of terrain geometry  $G$ . Previous experimental evidence shows that slip behavior is a highly nonlinear function of terrain slopes (Lindemann & Voorhees, 2005). So to model this highly nonlinear dependence, we use a type of receptive field regression algorithm (Schaal & Atkeson, 1998; Vijayakumar, D'Souza & Schaal, 2005). The main idea is to split the input domains into subregions, called *receptive fields*, and apply locally linear fits to the data to approximate a globally nonlinear function. While there are many algorithms which can be applied to this learning task, such as neural networks, support vector regression, etc., our choice is mainly motivated by the fact that the algorithm needs to be eventually running onboard the rover, so it has to allow fast online updates. The receptive field regression approach gives a good tradeoff between memory-based nonlinear regression methods (Hastie, Tibshirani & Friedman, 2001) and global function approximation methods, such as neural networks.

Slip  $S$  (we have dropped the subindex  $T$  for simplicity) can be written in the following form:

$$\hat{S}(\mathbf{y}) = \sum_{c=1}^C K(\mathbf{y}, \mathbf{y}_c) \left( b_0^c + \sum_{r=1}^R b_r^c \langle \mathbf{d}_r^c, \mathbf{y} \rangle \right),$$

where  $\mathbf{y}$  are the input slopes,  $\mathbf{y} = (y_{\text{pitch}}, y_{\text{roll}})$ ,  $K(\mathbf{y}, \mathbf{y}_c)$  is a weighting function, or kernel,  $K(\mathbf{u}, \mathbf{v}) = \exp(-\|\mathbf{u} - \mathbf{v}\|^2 / \lambda)$ ,  $\mathbf{y}_c$  is a training example which serves as a receptive field center,  $\mathbf{d}_r^c$  are several local projections in each receptive field  $c$ ,  $b_i^c$  are the corresponding regression coefficients,  $R$  is the number of linear projections (here  $R \leq 2$ ), and  $\lambda$  is a parameter which controls the receptive field size ( $\lambda > 0$ ). In other words, the slip  $S$ , corresponding to a query point  $\mathbf{y}$ , is computed as a linear combination of  $C$  linear functions (one per each receptive field), where the weights are computed according to the distance from  $\mathbf{y}$  to the centers of the receptive fields. As the weighting functions  $K(\mathbf{y}, \mathbf{y}_c)$  depend on the distance from the

query example  $\mathbf{y}$  to the receptive field centers  $\mathbf{y}_c$ , the final functional approximation will be nonlinear.

Now, given the training data  $\mathbf{D}_y = \{\mathbf{y}_i, S_i\}_{i=1}^N$ , where the vectors  $\mathbf{y}_i$  contain the estimated slopes from range imagery,  $S_i$  are the corresponding measurements of slip at this particular location, and  $N$  is the number of training examples, the learning procedure's task is to estimate the unknown parameters so that they fit the training data  $\mathbf{D}_y$  well. The parameters to be learned are the receptive field centers  $\mathbf{y}_c$ ,  $1 \leq c \leq C$ , the linear regression parameters  $b_0^c$ ,  $b_r^c$ ,  $\mathbf{d}_r^c$ ,  $1 \leq r \leq R$ ,  $1 \leq c \leq C$ , and the parameter  $\lambda$  which determines the receptive fields' size.

For a given  $\lambda$ , the receptive fields are distributed to cover the input space so that all training data belong to at least one receptive field. This is done by allocating a new receptive field in the input space whenever an incoming training example is not covered by other receptive fields, setting the center  $\mathbf{y}_c$  to be the new example (Schaal & Atkeson, 1998). To estimate the parameters  $b_r^c$ ,  $\mathbf{d}_r^c$  in each receptive field, a partial least squares (PLS) linear fit (Wold, 1966; Hastie et al., 2001) is performed, in which the training points are weighted according to their distance to the receptive field center (Vijayakumar et al., 2005). In our case of only two-dimensional inputs, one can also use the weighted linear regression (Schaal & Atkeson, 1998) or some other locally linear projection. However, by using PLS, the algorithm can be easily extended to working with higher dimensional inputs, because of the dimensionality reduction capabilities of PLS (Vijayakumar et al., 2005). As our method uses the PLS regression, it is closer to the locally weighted projection regression (LWPR) method of Vijayakumar et al. (2005). We parameterize the receptive field size by only one parameter  $\lambda$  (which implies symmetric kernels). More advanced structured kernels could be applied as in Vijayakumar et al. (2005), but they introduce additional parameters to be learned, which would require a larger sample size. We select the parameter  $\lambda$  using a validation set, in order to avoid overfitting.<sup>4</sup>

<sup>4</sup>The purpose of the validation set is to test the generalization of a learned model independently of its training and select the best model possible for the data. For example, in this particular case, selecting an infinitely small receptive field size would allow for one receptive field per each example and therefore perfect approximation of the function on the training set. However, this will result in a very poor performance on examples outside of the training data.

For example, the best selected  $\lambda$  for our soil dataset renders a kernel of local activity within about  $4^\circ$  in pitch and roll angles.

An important aspect of this algorithm is that, when a new example arrives during training, only the parameters of the receptive fields in the vicinity of this example are to be reevaluated. This allows for fast update in online learning. It is the result of constructing of a final cost function in such a way that competition among receptive fields is promoted, i.e., a receptive field is encouraged to fully approximate the required value of the function rather than splitting the responsibility among many receptive fields (Jacobs et al., 1991). The cost function, implicit in the receptive field regression algorithms, is the following:

$$\min_{b_0^c, b_r^c, \mathbf{d}_r^c} \sum_{c=1}^C \sum_{i=1}^N K(\mathbf{y}_i, \mathbf{y}_c) \left[ S_i - \left( b_0^c + \sum_{r=1}^R b_r^c \langle \mathbf{d}_r^c, \mathbf{y}_i \rangle \right) \right]^2.$$

In other words, the function is required to approximate well the observed output of an arbitrary data point by an individual receptive field, rather than approximating the data point output using multiple receptive fields. As a result of optimizing this cost function, the updates to the parameters of one receptive field are done independently of the parameters of the other receptive fields. This is an important point, because when new data arrive in a subregion of the input domain, only a subset of the parameters of the function will need to be adjusted, rather than having to reevaluate all the parameters of the function, as is done with neural networks, for example. This property also prevents the "catastrophic forgetting," typical of global approximation methods, and the algorithm does not need to store training examples in memory.

For now, the training is done in a batch mode, but the LWPR algorithm has been selected in view of future training online, onboard the rover. In particular, the properties of the receptive field regression approach that we find valuable are: the concept of a receptive field which makes keeping of a huge amount of data in memory unnecessary; the adaptability of creating and removing receptive fields as needed; and the possibility to easily extend the approach to online learning. We shall note that modeling with local nonlinear regression imposes very little restriction on the functional dependency. It allows for it to be nonlinear but does not assume any

particular model; instead, the model is learned from the data. An experimental comparison with a neural network algorithm is given in Section 7.3.3.

## 7.2. Implementation Details

In this section we describe in detail the information we use for training purposes. A 2D map of the environment is built using range information from the stereo pair images. The map has a cell representation with a cell size of  $0.2 \times 0.2$  m. The information kept per cell is its extents, average elevation, and a pointer to an image which has most recently observed this cell. This is sufficient to retrieve the required inputs when needed, i.e., when the cell is traversed, and does not overburden the system with keeping a huge volume of data per cell. To collect an example for the training data we do the following: for a particular cell in the map which is seen by the rover at a distance, we can compute information about appearance and measure the slopes (the input vector); when the rover traverses this cell the slip in  $X$ ,  $Y$ , or yaw (the output value) is measured. To be more efficient, the data collection goes in the reverse way: in each map cell the average elevation and a pointer to the image viewing it are stored, because it is not known which cells are to be traversed. It is only after the rover traverses some region that computations about slopes and terrain appearance are done and are added to the training data.

To estimate the slope at a particular location, we do a local plane fit to the average elevation in each cell in its neighborhood (Gennery, 1999). A slope estimate can be missing if there are not enough cells under the robot to do a plane fit. This can happen due to missing range data, e.g., in sparse vegetation or at the borders of the map. The slope is decomposed into a longitudinal (along the forward motion direction) and lateral (along the wheel axis, perpendicular to the forward motion) component with respect to the current position of the rover, i.e., the pitch and roll slope angles. The initial attitude of the rover, received from the IMU, is used to transform the retrieved longitudinal and lateral slope angles from the terrain into a gravity leveled frame. The slope angles cannot be perfectly evaluated because of noise in the range data and because the locally planar terrain assumption might be violated. As each location in the map is seen by many frames while the rover approaches it, we average the roll and pitch estimates to smooth some noise effects.

Localization is important for the success of this method. VO is used for the vehicle's localization. In the case of an outlying VO position estimate, the step is skipped and the map and rover position are reinitialized.

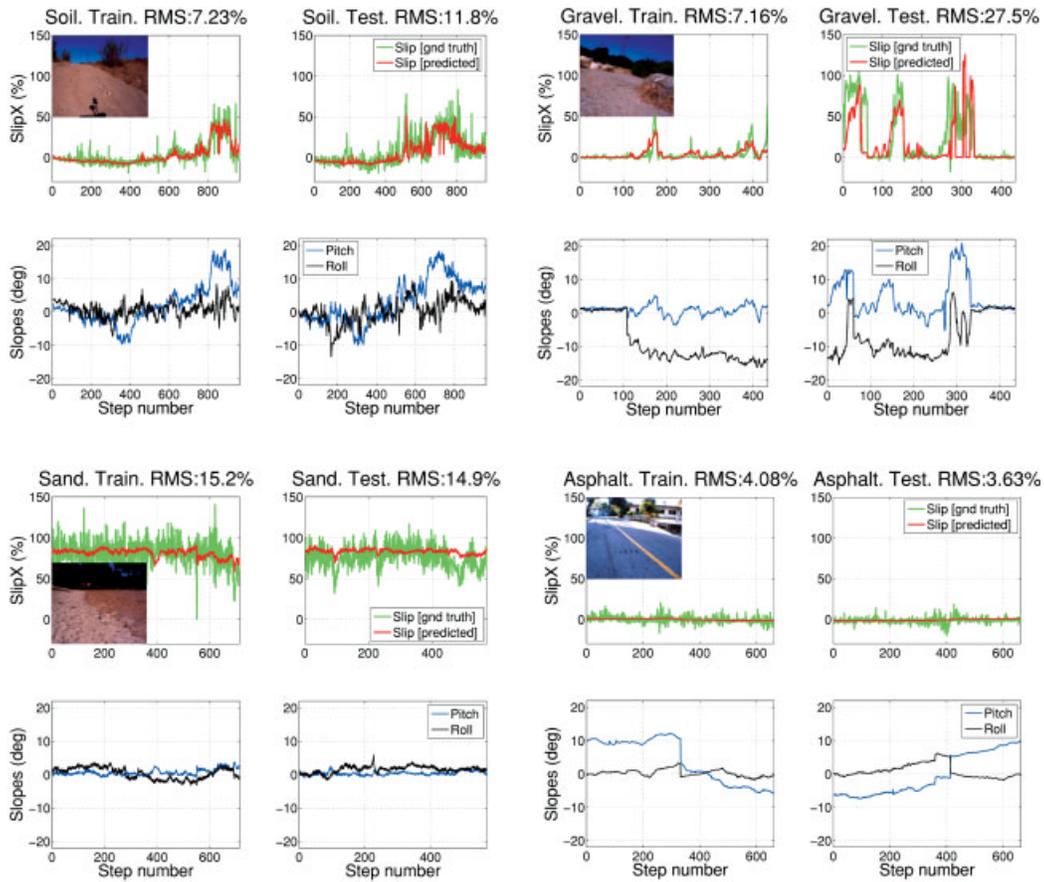
## 7.3. Experimental Results

### 7.3.1. Experimental Setup

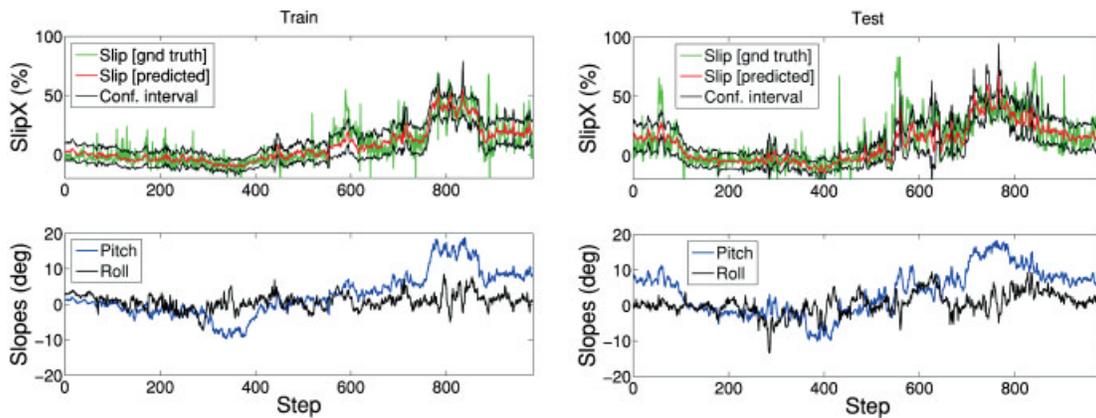
In this section we give experimental results of learning and prediction of slip from terrain slopes (estimated from visual information) when the traversed terrain type is known. Our dataset is composed of long stereo sequences (1000–2000 frames) which were taken on one terrain type at a time. We report below both training and test error. The training data are used to learn the regression function. After learning, the function is tested on the same data (training error) and also on data not used in training (test error). Naturally the training error will be smaller, but the test error is a criterion for the learning method's *generalization* abilities, i.e., how well it will perform on new, unseen data. To be able to measure the test error, we predict slip only on locations traversed by the rover but, in principle, prediction could be done at each point of the local map (wherever there are sufficient range data). So, slip can be predicted on different locations on the whole visible map, without the need for the rover to traverse them.

To do learning, for most of the experiments in this paper, we perform a sequential split of the data into training and test sets. That is, for each terrain type we take the frames up till some time for training, and test on all the frames after that. Some small portion of the data, between the training and test sets, is held out for validation. This is a more realistic scenario than the commonly used random split in the machine learning community, because the robot is expected to train on some portion of the terrain first and then continue to traverse the terrain applying what it has learned (testing). It is also more difficult because the distribution of input variables during training might shift to unexplored regions while testing, which makes it much harder to generalize.

Slip prediction error is measured by the average absolute error,  $\text{Err} = \sum_{i=1}^n |P_i - T_i| / n$ , or by the root means squared (RMS) error,  $\text{RMS} = \sqrt{\sum_{i=1}^n (P_i - T_i)^2 / n}$ , where  $P_i$  is the predicted and  $T_i$  is the target slip at a particular step  $i$ . The latter is more adequate for



**Figure 9.** Prediction of slip in X on soil (top left), gravely transverse slope (top right), flat sandy terrain (bottom left), and up and downslope asphalt (bottom right). Each panel contains the predicted and ground truth slip (top row) for the corresponding slope angles estimated from vision (bottom row); training data (left column), test data (right column). LAGR vehicle.



**Figure 10.** The slip prediction and its 1-sigma confidence intervals for the soil dataset (top) and the corresponding slope angles (bottom). Training mode (left). Test mode (right). Note that the uncertainty for the test set is at times larger (e.g., around step number 600) and much more spiky compared to the training set. This is because some test examples occur away from the regions covered by the training examples. Soil data. LAGR vehicle.

measuring the error of a regression function, but is more prone to outliers and can give an incorrect idea of the error. We do training and testing pointwise, i.e., not considering potential correlations between consecutive points, which do exist, and could be exploited in a more advanced prediction algorithm.

To allow for comparisons among datasets and platforms, slip will be represented in percent, by normalizing by the average velocity at which the dataset is taken.

### 7.3.2. Slip in $X$ for the LAGR Robot on Off-Road Terrain

The first experiment is done with the LAGR robot on five different off-road terrains (Figure 4). The first 45% of the data are used for training, the next 10% for validation, and the remaining 45% are used for testing. The data are taken by either manually joysticking the rover (soil and gravel datasets) at a speed of about 1 m/s, which can create variability in the commanded velocity, or by autonomous driving at a controlled straight constant velocity of 0.3 m/s (all the remaining datasets). The data are normalized by the average velocity for each dataset.

The results of slip prediction with the LAGR vehicle on soil, gravel, sand, and asphalt are presented in Figure 9. The actual learned nonlinear function of slip as dependent on terrain slopes for the soil terrain is shown in Figure 11 (left). The soil dataset consists of going up and down a slope twice, which helps the testing because similar slope angles have been seen in training. However, this does not happen in the gravel dataset where there is very little variability in the data it was trained on. This is a result of the consecutive split of the data. Also, notice that a lot of the input test slope angles have not been seen during training (this is also true for the soil data, although to a less extent). Still, the algorithm manages to generalize well, i.e., to extrapolate to unseen examples, in those circumstances. For some areas of large pitch angles in the gravel data the algorithm returned an invalid prediction (displayed as 0 on the plot). Although, in this case, these measurements have not been excluded from the prediction and contribute a lot to the final error, the ability of the algorithm to also provide a confidence or validity of the prediction can be certainly taken advantage of in practice. For the gravel dataset we used the vehicle's tilt angles (from the IMU) instead of the ones from the visual information because of

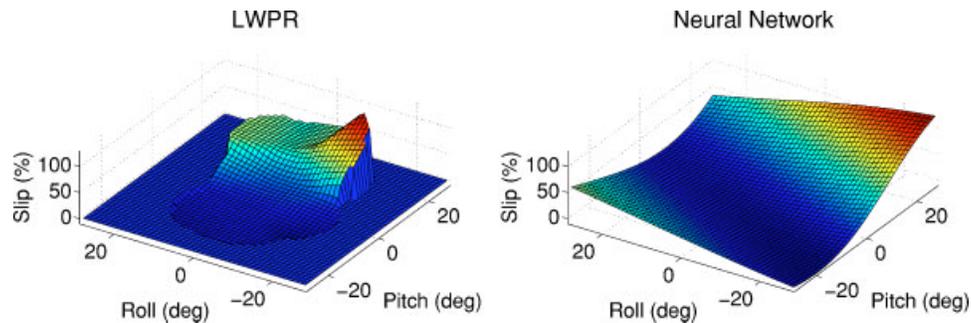
localization problems (due to occasional large rotations between consecutive frames which resulted in incorrect position estimates), but, with good localization, there are no significant differences between the two (Angelova, Matthies, Helmick, Sibley & Perona, 2006).

Prediction of slip in  $X$  for sand and asphalt terrains is given in the bottom row of Figure 9. Unfortunately, the LAGR vehicle mobility in deep sand turned out to be extremely poor. On a flat sandy terrain the vehicle experienced a consistent slip of about 80% (Figure 9; compare to the mobility of Rocky8 on sandy slopes described later on in Figure 13) and it was not possible to collect a dataset on sandy slopes with the LAGR vehicle. The consistent 80% slip in sand forces an almost constant function to be learned (Figure 9), which is quite natural in this case. Other slip behavior in our dataset that was uninteresting from a learning point of view was observed on asphalt and woodchip terrains. Similar to sand, a constant function is learned, because the measured slip for these datasets is approximately constant and independent of the slope angles.

On average we get slip error of about 3%–15% for all the datasets (except for gravel, with 27% RMS error, corresponding to 16% absolute error, which is achieved in a hard to generalize learning setup). This is quite a satisfying result in this type of data where a lot of noise is involved. In general, our results show very promising prediction of slip in real off-road outdoor environments.

As mentioned earlier, we are using the slope angles retrieved from stereo imagery (i.e., vision information). We have previously compared the slip prediction results when learning with respect to the vehicle's tilt angles (retrieved by the vehicle's IMU) and with respect to the slope angle estimates which are computed from the range data using visual information (Angelova, Matthies, Helmick, Sibley & Perona, 2006). Both are, in general, noisy measurements of the actual slope angles: the IMU based measurement gives the tilt of the robot, not of the ground plane, which might be erroneous if the robot traverses a rock, for example; the geometry based slope estimation is susceptible to outliers and can be wrong if there are obstacles in the plane fit area. Our results show (Angelova, Matthies, Helmick, Sibley & Perona, 2006) that they give comparable test performance.

For the purposes of using the slip prediction for planning, it is important to have a confidence value



**Figure 11.** The learned nonlinear function of slip as dependent of the two terrain slopes. Learning with LWPR (left) and with a neural network (right). The LWPR algorithm returns an invalid response (denoted with 0 on the plot) for regions which are far away from any receptive fields formed during learning. The neural network extrapolates easily but incorrectly in areas far from training data. Both methods manage to approximate the function in a similar way in the domain covered by training data. Soil data. LAGR vehicle.

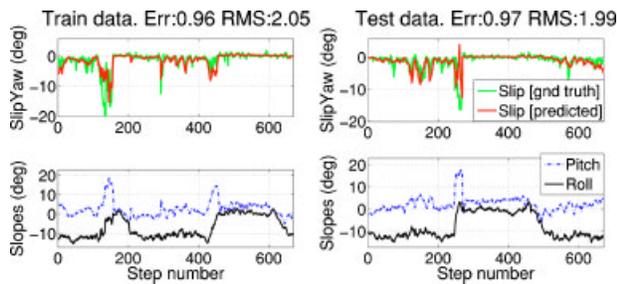
on each prediction, in order to know how much to trust it. We have computed the confidence intervals on individual query predictions as in Vijayakumar et al. (2005). The main assumptions in computing the variances are that two independent sources of noise are present in the case of LWPR: one coming from the locally linear fit in each receptive field and the other from the differences between the prediction of a local model and the final prediction. The latter measures how much local models agree in areas of overlap; it contributes significantly less to the uncertainty of the estimation. Figure 10 shows the confidence intervals for each query point for both training and test datasets for the soil terrain. The dataset has been normalized stepwise and is split consecutively into two equal size sets, without using a validation set. As we can see, query examples among the training data have smaller variance, whereas some test examples have larger variance whenever they fall into regions of the input space not covered by training data, or where the training examples are noisy or contradict each other. The most uncertainty (a large confidence interval) occurs on the boundary of the region within which any prediction response is given. No prediction is available outside this region, as it is too far from any receptive fields, see also Section 7.3.3.

### 7.3.3. Comparison of the LWPR Method to a Neural Network

In this section we compare the results of learning with the LWPR algorithm and with a standard non-

linear approximation method, e.g., a neural network, on the soil dataset (used also in Section 7.3.2). The neural network has ten hidden nodes, it has been trained for 10,000 epochs, uses early stopping and does not use weight decay. The LWPR has used 12 receptive fields to cover the input data domain. Figure 11 shows the learned nonlinear function (representing slip as a function of the longitudinal and lateral slopes, i.e., pitch and roll angles) evaluated for a range of values for both angles.

The test results showed comparable performance of both methods with some advantage to the LWPR (RMS of 11.89% and 12.64% for the LWPR and neural network, respectively, when the training is done on a sequential split of the data into equal sizes of training and test portions, with 5% of the examples in between held out for validation). The training data include pitch angles of only up to 17 deg and roll angles up to 8 deg in absolute value and include slip measurements of up to 65% with occasional outliers of up to 80%. Both methods generalize to regions which have not been observed during training, i.e., have reported slip predictions outside the training slope ranges (i.e., for pitch and roll angles larger than 17 and 8 deg, respectively). However, considerably different approaches to generalization to areas of the space, which have not been seen during training, can be seen in Figure 11. The neural network extrapolates incorrectly to regions where no training data are available. For example, it predicts  $\sim 50\%$  positive slip on a more than 20 deg downslope (see upper left corner on the right



**Figure 12.** Predicted slip in yaw (in deg) on a transverse gravelly slope (the slip in yaw has not been normalized). Training mode (left), test mode (right). LAGR vehicle.

subplot of Figure 11), which is wrong, because slip on a downslope is expected to be negative or zero. Instead, the LWPR method returns a confidence value on its prediction or in the simplest case a flag denoting that the predicted response is invalid. The latter happens if the query point has negligible weights with respect to all receptive fields. Naturally, if the training method had data covering the whole input space that would not be an issue, but usually, in practice, the available training data are not as variable or abundant as desired. The LWPR nonlinear approximation both gives better generalization performance and alerts of areas of the space where the result is not reliable. This adds more advantages of the LWPR method together with the previously mentioned fast online update, training in a memory efficient way (i.e., it does not need all the training data in memory), and lack of catastrophic forgetting when the input distribution is shifted to a new, unexplored domain (Vijayakumar et al., 2005).

#### 7.3.4. Slip in Yaw for the LAGR Robot on Off-Road Terrain

Apart from slip in the forward motion direction (i.e., slip in  $X$ ), slip in the other DOFs of the rover,  $Y$  and yaw, can also affect the rover mobility. For example, large amounts of slip in  $Y$  and yaw will prevent the rover from executing a planned path and therefore reaching the predetermined goal (Helmick et al., 2004), so predicting them, as well, would be very beneficial for the planning.

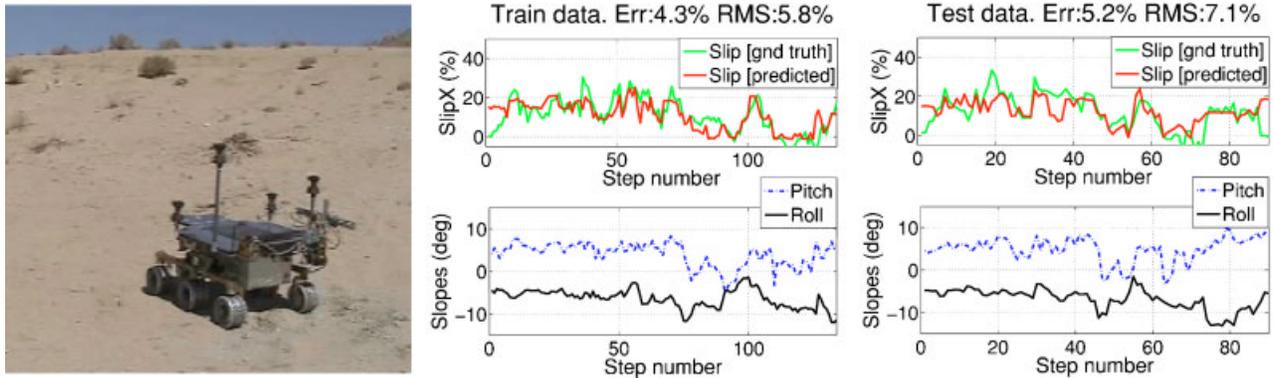
Figure 12 shows the result of learning and prediction of slip in yaw on a transverse gravelly terrain for the LAGR vehicle. An interesting functional de-

pendence is learned for this dataset: large slip in yaw, corresponding to large pitch angle, is learned whenever the roll angle is large, but an almost zero slip in yaw is learned when the roll angle is small, regardless of the pitch angle. This means that the pitch and roll angles work in conjunction to approximate the final slip well, i.e., both inputs are relevant for the measured quantity (here, slip in yaw). A similar effect has been observed in learning of slip in  $X$ , although the dependence of slip in  $X$  on the roll angle is significantly less pronounced. On the same dataset, a small amount of negative slip in  $Y$ , consistent with the large roll angle, could also be learned by our algorithm, see Angelova, Matthies, Helmick, Sibley & Perona (2006) for details. No significant slip in  $Y$  or yaw could be detected in any of the other datasets we have for this vehicle.

#### 7.3.5. Slip in $X$ for the Rocky8 Rover in the Mojave Desert

Another experiment with learning and prediction of slip in  $X$  is done for the Rocky8 rover, traversing sandy slopes in the Mojave desert. Figure 13 (left) shows the terrain where the data was collected. The dataset consists of about 220 steps and is taken on slopes which range from  $-5^\circ$  to  $10^\circ$  in pitch and up to  $12^\circ$  in roll. The ground truth for this dataset is obtained with a Total Station, tracking four prisms mounted on the rover, providing the 6-DOF pose within 2 mm and  $0.2^\circ$  accuracy in position and attitude, respectively. In this experiment we have used the roll and pitch angles provided by the ground truth. Slip is measured between steps which coincide with stops of the rover. Each step is taken in approximately constant time. As the step size can vary slightly, we average the slip across several neighboring steps. Here again, we normalize slip by the average step size ( $\sim 0.22$  m) to represent the slip in percent.

As the available data are rather small, we split the data randomly, rather than consecutively, into training and test portions (more examples are given to the training set, about 130 examples). To achieve statistically significant results, the experiment is performed multiple times with different random splits of the data into nonoverlapping training and test subsets. The test errors from 100 trials of this experiment are as follows: the average test RMS error is 6.5% with standard deviation of 0.6%, the average test absolute error (Err) is 5.5% with standard deviation of 0.5%.



**Figure 13.** Prediction of slip in  $X$  based on terrain slope angles. The dataset has been collected with the Rocky8 rover on sandy slopes in the Mojave Desert (left). Training mode (middle), test mode (right). The test error for this run is 5%–7%.

tion of 0.34%. Performing multiple trials and using the average and standard deviation prevents us from reporting the result of a single particularly favorable or unfavorable random split of the data. The consecutive split of the data (as performed in Sections 7.3.2, 7.3.3, and 7.3.4) is a much harder learning scenario and the split is uniquely defined, given the sizes of the training and test datasets.

The results for learning of slip in  $X$  from one of the trials and its corresponding errors are given in Figure 13. For this trial, slip prediction captures correctly (with error for the whole data within 5%–7%) slip of about 20% for high pitch angles. Note that in this dataset there are combinations of roll and pitch angles in the second part of the data (if split consecutively), which have not been seen in the first half to allow us to do a reasonable sequential split.

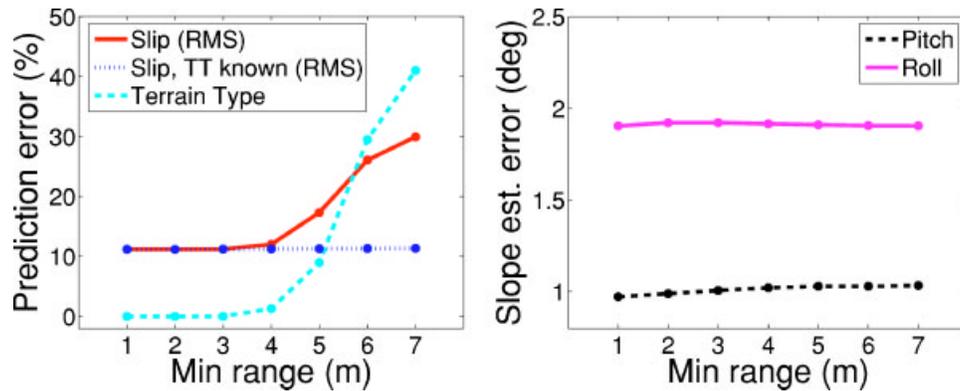
## 8. SLIP PREDICTION IN THE FULL FRAMEWORK

In this section we test the full slip prediction algorithm in which stereo imagery and the IMU are the only input and slip at a remote location is the output. The prediction works as follows: given an input example  $(x, y)$ , first the terrain type  $T_0 = T(x)$  is estimated from appearance  $x$  (using the terrain classifier described in Section 6) and then the learned slip model  $S_{T_0}(y)$  for the terrain type  $T_0$  is activated to produce slip results, given the measured terrain slopes  $y$  (Section 7). We present the final quantitative results by comparing the actual measured slip to the predicted slip.

### 8.1. Test Procedure

This experimental setup is similar to the one in Section 7.3.1 with the main difference that the terrain type in each patch is recognized first and a different slip model is used dependent on the terrain. The implementation details are described in Section 7.2. Some other minor differences in the final system are the change of cell size to  $0.4 \times 0.4$  m and the local neighborhood to  $4 \times 4$  cells (because larger cell regions are preferred by the texture classifier), and the mechanism for combining slope measurements about each location, obtained from different frames that have observed it. Here we average the measurements, weighting them by the inverse of the range at which they were obtained; no significant differences were noticed by changing the combining coefficient in the slope estimation. The same  $4 \times 4$  cell neighborhood and the same averaging scheme ( $1/\text{range}$ ) is used for both terrain classification and plane fit. The slip measurements in this dataset have been normalized pointwise by the commanded velocity, rather than normalizing all slip measurements by the average velocity. There were no significant differences, except that the pointwise normalized data are slightly noisier (see Figure 10, compare to Figure 9, top left). Here, again, to measure test performance, we predict slip only on the path which was later traversed by the rover. VO is used for the vehicle's localization.

There is one more issue of deciding at what range to start reporting the predicted slip and accumulating information as a particular location is be-



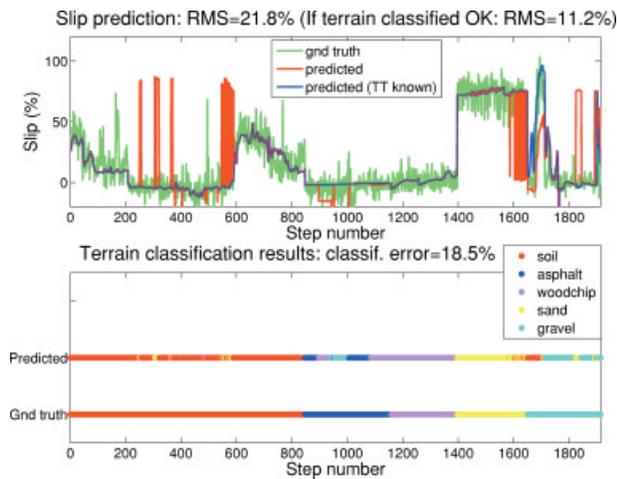
**Figure 14.** Slip prediction and terrain type classification errors (left) and slope estimation errors (right), as a function of the minimum range at which prediction is performed. Slip prediction error, if the terrain type were known, is also shown to the left. The terrain type classification error measures the percentage of misclassified cells along the rover traverse. This experiment is done for a  $15 \times 15$  m map on a subset of the soil terrain dataset.

ing approached (we call it “minimum range”). We explore what is the farthest minimum range for this robot. Naturally, a potential path planner would benefit more, the farther we can make a good slip prediction. On the other hand, locations observed at a large distance might give unreliable or noisy slope estimates, or provide very little information for the terrain classifier to be correct. Results of the slip prediction error, as a function of the minimum range at which prediction has started, are given in Figure 14. We can see that much better slip prediction is received for smaller initial ranges and that the deterioration in slip prediction is mainly due to terrain classification errors occurring at far ranges. The slope angle estimation seems to be much more stable with range for this dataset. The slope angle errors are computed against the roll and pitch angles received from the vehicle’s IMU, which are approximations of the actual slope angles. For our further experiments we will fix the minimum range at 3 m as a trade-off between a good enough slip prediction and a far enough initial range, preferred from the point of view of the planner. This means that if a location is seen at a closer than 3 m range we would not use any information we acquire about it (through vision or other sensors) to improve our slip prediction. So, all estimations or predictions about slope angles, terrain type, and slip will be accumulated between the ranges of 3 m and possibly 8.4 m (8.4 m is the diagonal distance from the center to the corner of a  $12 \times 12$  m map; in practice, very few cells will occur at ranges larger than 6 m).

To briefly summarize the test procedure: at each step a 2D map of the terrain is built (the map is of size  $12 \times 12$  m, each cell is  $0.4 \times 0.4$  m), the terrain classifier is applied to each visible cell in the map and the terrain slope is estimated whenever there are enough cells in the neighborhood. For each future rover position which is within the map we save the estimated slope with a coefficient of  $1/\text{range}$  and all the terrain classification responses in the cell neighborhood with their corresponding  $1/\text{ranges}$ . The final slope measurements are weighted averages, and the final terrain classification is resolved with voting among all terrain types that were recorded in the neighborhood when the same location is seen multiple times, each vote counting according to its weight. This mechanism is very useful to remove terrain classification errors, initially occurring at far range. The final measured slope angles are given as input to the slip behavior predictor learned for the detected terrain type of the cell in question. In the experiments the statistics are accumulated at ranges larger than 3 m and the predicted slip is reported only for cells which have been observed at least 3 m away.

## 8.2. Results

In this section we describe the results of final slip prediction for the dataset collected with the LAGR robot. The test dataset in this section is a composite of sequences of frames in which the terrain type is



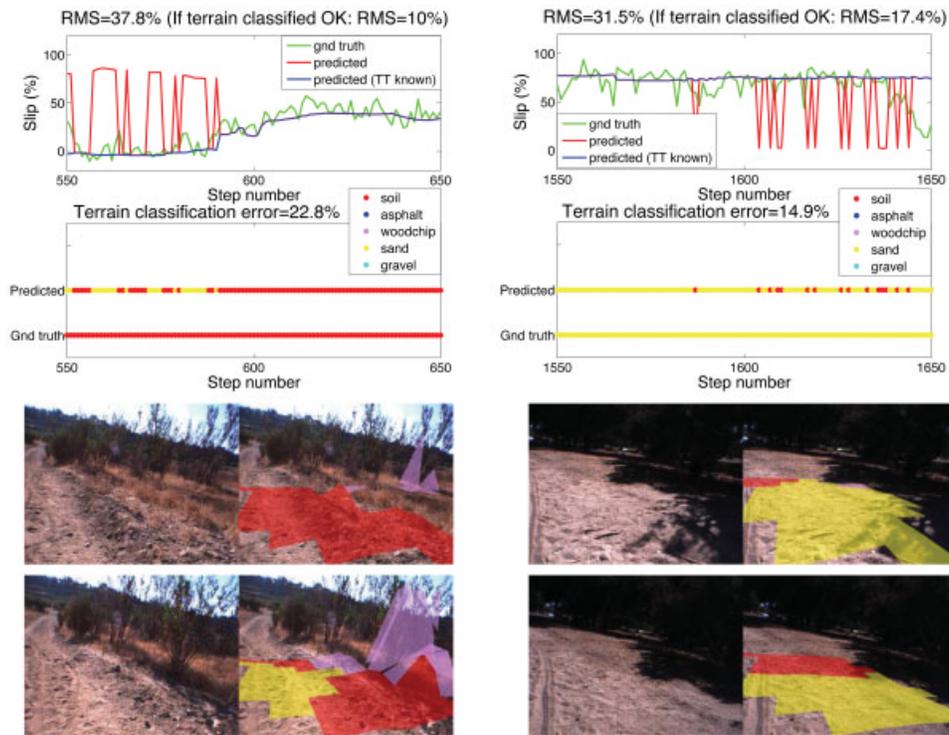
**Figure 15.** Results of slip prediction from stereo imagery (terrain geometry and appearance) on the whole dataset. Top: The predicted and measured slip for the corresponding test sequences. Slip prediction, assuming correctly recognized terrain type, is also shown (naturally, it coincides with the final slip prediction, whenever the terrain type is classified correctly). Bottom: The predicted and correct terrain types across the dataset. The test data spans about 300 m.

constant within a sequence but can change to another terrain for the next sequence. In this way a human operator can specify the terrain type of a long image sequence, instead of giving ground truth for each image. The terrain classification algorithm does not have the knowledge that the terrain is continuous for some number of frames and then can abruptly change. The algorithm which estimates the slopes, however, is aware of that change because a new frame sequence has to come with a different initial gravity leveled (IMU based) pose. A sequence size varies between 60 and 200 frames and the whole composite dataset contains about 2000 test frames. We have made sure that the test dataset has not been used for training.

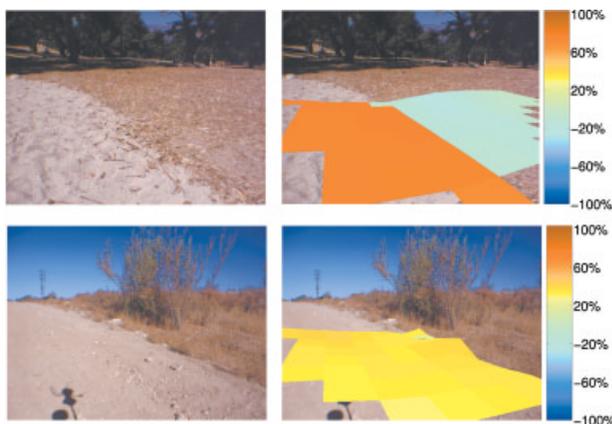
The results of the full slip prediction experiment for the abovementioned large “composite” dataset are shown in Figure 15. The figure shows the color coded terrain type classification results, the measured slip, the predicted slip, and the predicted slip, if the terrain type were known. The final slip prediction error for the whole dataset is 21.8%. When the terrain type is classified correctly, the slip prediction error is 11.2%. As seen in the figure, large slip errors

come from misclassified terrain types (usually soil and gravel are misclassified for sand). Figure 16 shows more details on some of the frames which incur large slip prediction errors, in particular, frames in which soil and sand are misclassified. The predicted terrain type is based on a single frame and is determined by weighted voting of the predictions in the cell neighborhood. The color coding is consistent with Figure 7. As seen in the figure, the task of discriminating between those two terrains is very challenging in our field test data. Also, notice the inconsistent ground truth slip measurement in the end of this sequence (the rover traversed more solid terrain). In this dataset the error is artificially increased as the slip measured for level sandy terrain is about 80%, which gives a rather large slip error due to terrain misclassification (compare to the error if the terrain type were correct). This result also shows that some errors are more dangerous than others. In other words, that the terrain classification algorithm should be applying different penalties for different types of error, i.e., terrain misclassification which leads to large slip errors should be given larger cost. In general, those results are very promising given the level of difficulty that the field test data offer.

Up until now, we have shown the performance on the future rover path only, but in actual test setup the result of the algorithm will be as shown in Figure 17, namely, each cell on the visible map will have predictions of slip in X (or Y, yaw) for potentially different orientations of the rover. The results in Figure 17 show the predicted slip in X for a fixed, neutral (i.e., the rover yaw is 0) orientation of the rover. The rover has detected the flat sandy area as hazardous because of large slip, as opposed to the adjacent flat woodchip area which would incur almost 0% slip. Slip of about 40–50% is predicted on a slope covered with soil when traversed upslope. The results are done on an individual map (each location is observed only once) but the terrain type in each cell is determined by weighted voting in the neighborhood, which helps reduce occasional terrain classification errors. No slip is predicted for the cells close to the corners of the image as there is not enough support of neighboring cells to estimate the slopes. No prediction is given for terrain cells which are classified as grass (bottom row of the figure) as there is no slip model for the grass terrain type. Slip prediction based traversal cost will be handed down to a path planner which can avoid very slippery ter-



**Figure 16.** Expanded results of slip prediction from Figure 15 for the areas incurring most error. Frames 550–650, soil terrain is sometimes misclassified for sand (left column). Frames 1550–1650, sand is misclassified for soil (right column). The terrain classification on several frames from these sequences is also shown in the bottom rows.



**Figure 17.** Prediction of slip in X on the forward looking map. Input images (left), result prediction (right). Sand, level ground (top row), soil, uphill (bottom row). LAGR vehicle.

rains. The prediction on the whole map takes about 6 s per frame for a  $12 \times 12$  m map on a 3.6 GHz P4 processor (from which 77% of the time is devoted to terrain type recognition).

### 8.3. Notes from the Field Tests

Several issues regarding slip prediction done on our real-life dataset are worth mentioning.

While collecting slip data with the LAGR vehicle, we have encountered the problem of how to factor out the dependence of slip on the vehicle’s velocity. This problem could be ignored for a Rocky8 type rover, as the Mars rovers drive at relatively low speeds and the dependence of slip on the commanded velocity is known to be insignificant for small velocities. But this was an issue with the LAGR robot, especially as originally the data were collected by joysticking the rover, thus introducing variability in rover velocity. We addressed this problem by forcing the vehicle to drive at relatively low

constant speeds in our later data collections, but for the datasets collected at higher speeds only normalization is performed, which does not fully solve the problem. Extending the application for Earth based rovers (or rovers driving faster) would require the training and testing to be performed at similar speed ranges. It is also possible to introduce speed as an input to our system, with the caveat that in this case training data at various speed ranges need to be collected.

Driving the rover at higher speeds caused some occasional errors with the VO-based ground truth estimation. This was not surprising as there was not always a sufficient overlap between two consecutive stereo pairs (especially when the rover is turning; in particular, we observed that behavior while driving the robot on a transverse gravelly slope and, although not commanded to rotate, the rover was experiencing significant slip in yaw). The occasional VO errors practically disappeared when the rover was forced to drive at a lower speed. Additionally, we had to avoid test scenarios in which the robot was following its own shadow, in the cases in which the shadow took a large portion of the stereo images. This case is particularly problematic for VO, as the contour of the rover shadow offers prominent features which are often found and matched in the next frame sequence, but produce a wrong motion estimate. In most cases though, the VO algorithm is robust enough to identify them as outliers. The proposed method assumes good vehicle localization and, while VO provides satisfactory results here, robot localization is still a topic of ongoing research.

As the dataset is collected in the field, the slip measurements themselves are quite noisy because of random effects coming from the terrain, measurement errors, etc. Also, because of the adopted macromodeling, there are events from the terrain which we do not model and therefore cannot predict. For example, when one of the wheels hits a small rock or falls in a gutter on the path, an unexpected motion (e.g., in yaw) occurs, which will not happen on an otherwise homogeneous terrain. In general, we are aware that the slip could vary in some range on the same terrain type and slope, for reasons we cannot control.

#### 8.4. Discussion

In this section we have performed prediction of slip from visual information only. From our results we

can conclude that learning of slip is possible and can lead to successful prediction from visual information. There are other factors, such as uneven wheel sinkage, unequal vehicle weight distribution, and unequal traction across different wheels, which also influence slip. This makes slip prediction a hard problem a priori, from both mechanical and machine vision point of view (Leger et al., 2005; Biesiadecki et al., 2005). So, there will be cases when slip prediction would not be possible with the available tools. Still, there is a strong motivation for enhancing the terrain perception for detecting nongeometric hazards whenever possible.

We believe that learning from examples is a viable approach when a physical model is not available, or is too complex to compute. We want to note that from a mechanical point of view slip behavior of rigid wheels (Rocky8) and of pneumatic wheels (LAGR) is described by different physical models; in addition, slip on off-road terrains, such as soil and sand, is modeled differently from slip on a rigid surface, such as asphalt (Wong, 1993). Nevertheless, we could learn all those different behaviors from examples because of the flexibility of our learning approach to adjust the learning model according to the data.

## 9. CONCLUSIONS AND FUTURE WORK

In this paper we have proposed to predict slip, a property of mechanical vehicle-terrain interaction, remotely using vision information as input. The idea is to map the appearance and geometry of the forthcoming terrain to its mechanical properties, measured when the robot traverses it, and learn this mapping from previous experience. Our approach is based on texture recognition and nonlinear regression modeling. Experimental evaluation has been performed on several natural terrains with two vehicle platforms. The proposed algorithm gives very satisfactory results for slip prediction, given the fact that a lot of noise is involved in measuring it and that the dataset is taken on completely natural, off-road terrains.

The importance of this method is that it is predictive. That is, the rover can avoid terrains of large slip before getting stuck. This is possible because the input information, obtained from stereo imagery, is available at each location in the map (wherever there are range data) so the rover can predict slip at each

observed, but not necessarily traversed, location. The output of the slip prediction algorithm is intended to be incorporated into a more intelligent path planner, so that areas of large slip are avoided as potential hazards, or more adequate control commands are issued taking into consideration the expected slip.

This work also opens up the topic of learning more about the mechanical properties of the terrain from visual information when the underlying physical model is unknown, and with the possibility of learning in a completely autonomous mode, i.e., with only sensors onboard the vehicle. We have provided a solution for the case of slip prediction, but this can be extended to various mechanical properties, e.g., soil compressibility, or wheel sinkage.

Further efforts are needed to develop a better terrain classification algorithm, to avoid erroneous slip prediction due to terrain type classification errors. Currently, the terrain classification algorithm considers the terrain at a low level, i.e., in small local regions only. A more advanced algorithm to consider terrain classification in a probabilistic framework and to introduce spatial and temporal continuity of the classification over neighboring patches or dependent on terrain geometry will be one of our main problems for future work. Visual information might not be sufficient to distinguish various terrain types and properties, especially considering Mars terrains. It can be complemented with multispectral imaging or other sensors to resolve some inherent visual ambiguities and improve on the classification results.

Our experiments required offline training because a human operator was needed to determine the terrain types. Future work could concern learning in the full MOE framework, i.e., closing the loop on learning with fully automatic measurements from the vehicle's onboard sensors. In that case, a fully automatic online learning will be performed, which has influenced the selection of the algorithms and methods in this work.

Slip prediction from visual information does not undermine the merit of mechanical modeling of terrain. Instead, it exploits sensory information, e.g., vision, which is unavailable or not yet utilized by mechanical models and thus can be complementary to them. An important area of research would be how to combine these two different levels of information.

## ACKNOWLEDGMENTS

The research described here was carried out by the Jet Propulsion Laboratory, California Institute of Technology, with funding from the NASA's Mars Technology Program. The authors thank the JPL LAGR team for giving us access to the LAGR vehicle, Andrew Howard, Steve Goldberg, Gabe Sibley, Nathan Koenig, and Lee Magnone for helping them with the data collection, and Daniel Clouse and four anonymous reviewers for providing very useful comments on the paper.

## REFERENCES

- Andrade, C., Ben Amar, F., Bidaud, P., & Chatila, R. (1998). Modeling robot-soil interaction for planetary rover motion control. Paper presented at the International Conference on Intelligent Robots and Systems.
- Angelova, A., Matthies, L., Helmick, D., Sibley, G., & Perona, P. (2006). Learning to predict slip for ground robots. Paper presented at the International Conference on Robotics and Automation.
- Angelova, A., Matthies, L., Helmick, D., & Perona, P. (2006). Slip prediction using visual information. Paper presented at the Robotics: Science and Systems Conference.
- Arvidson, R., Anderson, R., Bartlett, P., Bell, J., Christensen, P., Chu, P., et al. (2004). Localization and physical properties experiments conducted by Opportunity at Meridiani planum. *Science*, 306(5685), 821–824.
- Batavia, P., & Singh, S. (2001). Obstacle detection using adaptive color segmentation and color stereo homography. Paper presented at the International Conference on Robotics and Automation.
- Bauer, R., Leung, W., & Barfoot, T. (2005). Experimental and simulation results of wheel-soil interaction for planetary rovers. Paper presented at the International Conference on Intelligent Robots and Systems.
- Bekker, M. (1969). Introduction to terrain-vehicle systems. Ann Arbor: University of Michigan Press.
- Bell, J., Squyres, S., Herkenhoff, K., Maki, J., Arneson, H., Brown, D., et al. (2003). Mars Exploration Rover Athena panoramic camera investigation. *Journal of Geophysical Research*, 108(E12), 8063, doi: 10.1029/2003JE002070.
- Bellutta, P., Manduchi, R., Matthies, L., Owens, K., & Rankin, A. (2000). Terrain perception for DEMO III. Paper presented at the IEEE Intelligent Vehicles Symposium.
- Biesiadecki, J., Baumgartner, E., Bonitz, R., Cooper, B., Hartman, F., Leger, P., et al. (2005). Mars Exploration Rover surface operations: Driving Opportunity at Meridiani planum. Paper presented at the IEEE Conference on Systems, Man, and Cybernetics.
- Brooks, C., Iagnemma, K., & Dubowsky, S. (2005).

- Vibration-based terrain analysis for mobile robots. Paper presented at the International Conference on Robotics and Automation.
- Cheng, Y., Maimone, M., & Matthies, L. (2005). Visual odometry on the Mars Exploration Rovers. Paper presented at the IEEE International Conference on Systems, Man and Cybernetics.
- Daily, M., Harris, J., Keirse, D., Olin, D., Payton, D., Reiser, K., et al. (1988). Autonomous cross-country navigation with the ALV. *International Conference on Robotics and Automation*, 2(1), 718–726.
- Dima, C., Vandapel, N., & Hebert, M. (2004). Classifier fusion for outdoor obstacle detection. Paper presented at the International Conference on Robotics and Automation.
- Farritor, S., Hacot, H., & Dubowsky, S. (1998). Physics based planning for planetary exploration. Paper presented at the International Conference on Robotics and Automation.
- Gennery, D. (1999). Traversability analysis and path planning for a planetary rover. *Autonomous Robots*, 6(2), 131–146.
- Goldberg, S., Maimone, M., & Matthies, L. (2002). Stereo vision and rover navigation software for planetary exploration. Paper presented at the IEEE Aerospace Conference, Big Sky, Montana, 5.
- Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. New York: Springer.
- Helmick, D., Cheng, Y., Roumeliotis, S., Clouse, D., & Matthies, L. (2004). Path following using visual odometry for a Mars rover in high-slip environments. Paper presented at the IEEE Aerospace Conference, Big Sky, Montana.
- Helmick, D., Roumeliotis, S., Cheng, Y., Clouse, D., Bajracharya, M., & Matthies, L. (2005). Slip compensation for a Mars rover. Paper presented at the International Conference on Intelligent Robots and Systems.
- Hoffman, B., Baumgartner, E., Huntsberger, T., & Schenker, P. (1999). Improved rover state estimation in challenging terrain. *Autonomous Robots*, 6(2), 113–130.
- Howard, A., Tunstel, E., Edwards, D., & Carlson, A. (2001). Enhancing fuzzy robot navigation systems by mimicking human visual perception of natural terrain traversability. In *Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference*.
- Iagnemma, K., & Dubowsky, S. (2004). Traction control of wheeled robotic vehicles with application to planetary rovers. *International Journal of Robotics Research*, 23(10), 1029–1040.
- Iagnemma, K., Shibly, H.A., & Dubowsky, S. (2002). On-line terrain parameter estimation for planetary rovers. Paper presented at the International Conference on Robotics and Automation, and Automation in Space.
- Ishigami, G., Miwa, A., Nagatani, K., & Yoshida, K. (2006). Terramechanics-based analysis on slope traversability for a planetary exploration rover. Paper presented at the International Symposium on Space Technology and Science.
- Jacobs, R., Jordan, M., Nowlan, S., & Hinton, G. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3, 79–87.
- Jain, A., Guineau, J., Lim, C., Lincoln, W., Pomerantz, M., Sohl, G., & Steele, R. (2003). ROAMS: Planetary surface rover simulation environment. Paper presented at the International Symposium on Artificial Intelligence, Robotics and Automation in Space (pp. 19–23).
- Kelly, A., & Stentz, A. (1998). Rough terrain autonomous mobility part 2: An active vision, predictive control approach. *Autonomous Robots*, 5, 163–198.
- Kim, D., Sun, J., Oh, S., Rehg, J., & Bobick, A. (2006). Traversability classification using unsupervised on-line visual learning for outdoor robot navigation. Paper presented at the International Conference on Robotics and Automation.
- Kraus, P., Fredriksson, A., & Kumar, V. (1997). Modeling of frictional contacts for dynamic simulation. Paper presented at the International Conference on Intelligent Robots and Systems.
- Lacaze, A., Murphy, K., & Delgiorno, M. (2002). Autonomous mobility for the Demo III experimental unmanned vehicles. In *Proceedings of the AUVSI Conference*.
- Le, A., Rye, D., & Durrant-Whyte, H. (1997). Estimation of track-soil interactions for autonomous tracked vehicles. Paper presented at the International Conference on Robotics and Automation.
- LeCun, Y., Muller, U., Ben, J., Cosatto, E., & Flepp, B. (2005). Off-road obstacle avoidance through end-to-end learning. In *Advances in neural information processing systems*. Cambridge: MIT Press.
- Leger, C., Trebi-Ollennu, A., Wright, J., Maxwell, S., Bonitz, R., Biesiadecki, J., et al. (2005). Mars Exploration Rover surface operations: Driving Spirit at Gusev crater. Paper presented at the IEEE Conference on Systems, Man, and Cybernetics.
- Leung, T., & Malik, J. (2001). Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal on Computer Vision*, 43(1), 29–44.
- Lieb, D., Lookingbill, A., & Thrun, S. (2005). Adaptive road following using self-supervised learning and reverse optical flow. Paper presented at the Robotics: Science and Systems Conference.
- Lindemann, R., & Voorhees, C. (2005). Mars Exploration Rover mobility assembly design, test and performance. Paper presented at the IEEE International Conference on Systems, Man and Cybernetics.
- Macedo, J., Manduchi, R., & Matthies, L. (2000). Ladar-based discrimination of grass from obstacles for autonomous navigation. *Lecture Notes in Control and Information Sciences: Experimental Robotics VII*, 271, 111–120.
- Manduchi, R. (1999). Bayesian fusion of color and texture segmentations. Paper presented at the IEEE International Conference on Computer Vision.
- Matthies, L. (1989). *Dynamic stereo vision*. Unpublished Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA.
- Matthies, L., Bergh, C., Castano, A., Macedo, J., & Manduchi, R. (2003). Obstacle detection in foliage with ladar and radar. In *Proceedings of International Symposium of Robotics Research*.

- Matthies, L., & Schafer, S. (1987). Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, RA-3(3), 239–250.
- Matthies, L., Turmon, M., Howard, A., Angelova, A., Tang, B., & Mjolsness, E. (2005). Learning for autonomous navigation: Extrapolating from underfoot to the far field. Paper presented at NIPS, Workshop on Machine Learning Based Robotics in Unstructured Environments.
- Moore, H., Bickler, D., Crisp, J., Eisen, H., Gensler, J., Haldemann, A., et al. (1999). Soil-like deposits observed by Sojourner, the Pathfinder rover. *Journal of Geophysical Research*, 104(E4), 8729–8746.
- Nister, D., Naroditsky, O., & Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1), 3–20.
- Ojeda, L., Borenstein, J., Witus, G., & Karlsen, R. (2006). Terrain characterization and classification with a mobile robot. *Journal of Field Robotics*, 23(2), 103–122.
- Olson, C., Matthies, L., Shoppers, M., & Maimone, M. (2001). Stereo ego-motion improvements for robust rover navigation. Paper presented at the International Conference on Robotics and Automation.
- Pomerleau, D. (1989). ALVINN: An autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems*, 1, 305–313.
- Rasmussen, C. (2001). Laser range-, color-, and texture-based classifiers for segmenting marginal roads. Paper presented at the Conference on Computer Vision and Pattern Recognition.
- Schaal, S., & Atkeson, C. (1998). Constructive incremental learning from only local information. *Neural Computation*, 10(8), 2047–2084.
- Seraji, H. (2000). Fuzzy traversability index: A new concept for terrain-based navigation. *Journal of Robotics Systems*, 17(2), 75–91.
- Tarokh, M., MacDermott, G., Hayati, S., & Hung, J. (1999). Kinematic modeling of a high mobility Mars rover. Paper presented at the International Conference on Robotics and Automation.
- Terzaghi, K. (1948). *Soil mechanics in engineering practice*. New York: Wiley.
- Ulrich, I., & Nourbakhsh, I. (2000). Appearance-based obstacle detection with monocular color vision. In *Proceedings of the AAAI National Conference on Artificial Intelligence*.
- Vandapel, N., Huber, D., Kapuria, A., & Hebert, M. (2004). Natural terrain classification using 3-d ladar data. Paper presented at the International Conference on Robotics and Automation.
- Varma, M., & Zisserman, A. (2003). Texture classification: Are filter banks necessary? Paper presented at the Conference on Computer Vision and Pattern Recognition.
- Varma, M., & Zisserman, A. (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1), 61–81.
- Vijayakumar, S., D'Souza, A., & Schaal, S. (2005). Incremental online learning in high dimensions. *Neural Computation*, 17(12), 2602–2634.
- Wellington, C., Courville, A., & Stentz, A. (2005). Interacting Markov random fields for simultaneous terrain modeling and obstacle detection. Paper presented at the Robotics: Science and Systems Conference.
- Wellington, C., & Stentz, A. (2004). Online adaptive rough-terrain navigation in vegetation. Paper presented at the International Conference on Robotics and Automation.
- Wold, H. (1966). Estimation of principal components and related models by iterative least squares. In P.R. Krishnaiah (Ed.), *Multivariate analysis* (pp. 391–420). New York: Academic Press.
- Wong, J. (1993). *Theory of ground vehicles*. New York: Wiley.