

Article

Terrain Perception in a Shape Shifting Rolling-Crawling Robot

Fuchida Masataka ^{1,*}, Rajesh Elara Mohan ², Ning Tan ³, Akio Nakamura ⁴ and Thejus Pathmakumar ⁵

¹ Department of Advanced Multidisciplinary Engineering, Tokyo Denki University, Tokyo 120-8551, Japan

² Engineering Product Development Pillar, Singapore University of Technology and Design, Singapore 487372, Singapore; rajeshelara@sutd.edu.sg

³ Department of Biomedical Engineering, National University of Singapore, Singapore 117583, Singapore; tanningrobotics@gmail.com

⁴ Department of Robotics and Mechatronics, Tokyo Denki University, Tokyo 120-8551, Japan; nkmr-a@cck.dendai.ac.jp

⁵ SUTD-JTC I³ Center, Singapore University of Technology and Design, Singapore 487372, Singapore; thejus08@gmail.com

* Correspondence: fuchida@is.fr.dendai.ac.jp; Tel.: +81-3-5284-5604

Academic Editor: Huosheng Hu

Received: 30 July 2016; Accepted: 2 September 2016; Published: 27 September 2016

Abstract: Terrain perception greatly enhances the performance of robots, providing them with essential information on the nature of terrain being traversed. Several living beings in nature offer interesting inspirations which adopt different gait patterns according to nature of terrain. In this paper, we present a novel terrain perception system for our bioinspired robot, Scorpio, to classify the terrain based on visual features and autonomously choose appropriate locomotion mode. Our Scorpio robot is capable of crawling and rolling locomotion modes, mimicking Cebrenus Rechenburgi, a member of the huntsman spider family. Our terrain perception system uses Speeded Up Robust Feature (SURF) description method along with color information. Feature extraction is followed by Bag of Word method (BoW) and Support Vector Machine (SVM) for terrain classification. Experiments were conducted with our Scorpio robot to establish the efficacy and validity of the proposed approach. In our experiments, we achieved a recognition accuracy of over 90% across four terrain types namely grass, gravel, wooden deck, and concrete.

Keywords: terrain perception; reconfigurable robot; support vector machine; rolling-crawling locomotion; bag-of-words

1. Introduction

Living beings in nature possess innate abilities to adapt their gaits in response to the nature of terrain being traversed which could vary from smooth, flat ground to bumpy, slippery regions posing serious hazard. Some creatures adapt their locomotion mode to navigate such extreme terrain types, whereas a smaller group of other creatures uses multiple locomotion modes and switches to appropriate locomotion type to overcome the terrain challenges. Just as in nature, the characteristics of a given terrain greatly influence the locomotion performance of a robot that navigates in that terrain. Perceiving the type of terrain not only helps the robot in realizing efficient locomotion but also aids in avoiding obstacles and for path planning purposes. Numerous studies have been done with respect to terrain perception in fixed morphology robotic systems, especially involving legged and wheeled robots.

A number of sensing modalities and connected approaches have been put forward and validated towards solving terrain perception problems. In [1], Weiss proposed a method for classifying terrains

by analyzing the vibrations of the robot body using accelerometer. This method showed good results in gravel, grass, and clay with classification error of 2.5% and an overall misclassification error of 5%. In [2], Graeme Best proposed a terrain perception method for legged robots by attaching force sensors on the legs. This approach also helped to analyze how the robot interacted with the terrain. This approach reports a high accuracy across all five type of terrains types considered. In [3], an assistive system for the visually impaired was proposed. This system used Kinect sensor for identifying 3D obstacle and for classification of terrain types. This system also interacted with the user by synthesizing tactile feedback in response to perceived terrain. In [4], Steffen Zenker proposed terrain classification system for adopting an energy efficient gait for hexapod robot using monocular vision. This online terrain classification showed 90% accuracy in terrain classifications. In another study [5], a terrain perception system using a laser stripe-based structured light sensor for autonomous ground vehicles was discussed. This system was reported to have terrain classification accuracy over 90%. Beyond the use of single sensor for terrain classification, a number of studies have been reported that using a sensor fusion approach. Ascaris [6] used a fusion of vision and distributed tactile sensors installed on the feet of the robot for terrain classification. One more study, [7], discussed a terrain perception for multi resolution digital terrain model generation for the Mars landing site assessments. This system used stereo vision methodologies for analyzing the geographical characteristics of spaceship landing sites on Martian terrain. The main challenges in designing a terrain perception system for the reconfigurable robotic platform under testing are reduced computational efficiency of the on board microcontroller system on the robotic platform, low resolution camera used, and associated noise and the space constraints to accommodate multiple sensors for understanding the terrain characteristics. From a research method point of view, a number of strategies have been put forward and validated. A machine learning based approach had been proposed in [8,9] for terrain perception using visual information. In [10], a neural network method was used for classification of terrains. This research work showed robust visual terrain perception method using feature sequence generation followed by a recurrent neural networks. The strategy presented in [11] uses probabilistic terrain analysis algorithm for drivable and non-drivable terrain identification for an autonomous vehicle. This method used single-scan laser sensor for terrain analysis which helps off-road robotic driving easy. A Support Vector Machine (SVM) based approach had been used in [1–4] for classifying terrains using visual information from an on board camera. In [12], SVM was used in addressing long range terrains classification problem where it has been proven in aiding autonomous navigation. The results showed that the color-based feature extraction followed by SVM performed good in near field terrain identification.

Reconfigurable robots are gaining wide spread popularity because of their abilities to change their morphologies in relation to task or terrain needs. In [13], authors proposed a circular soft robot which can crawl and jump. The shape deformation capabilities of this robot helped to perform both crawling and jumping motions. The physical simulations exhibited that this robot could execute jump up to 80 mm. The design and performance analysis of a novel leg wheel transformable robot has been discussed in the work mentioned in [14]. Authors presented a unique transformation method that switches its morphology that enables the transformation from a driving mechanism on wheels to a walking mechanism with 2 degrees of freedom in the leg. Authors also validated the performance of the robotic platform on stair climbing using legged mode and multiple terrain locomotion using wheel driving mode. A Salamander inspired robot was described in [15] which can switch mode of locomotion from swimming to walking in relation to the terrain being traversed. However, most research in reconfigurable robots were limited to mechanism designs and control studies with very minimal effort related to sensing or autonomy. Terrain perception is a key characteristics essential for most reconfigurable robot that allows for morphology change in relation to perceived terrain. In this work, we propose a novel biomimetic robot, Scorpio, which is capable of switching between rolling and crawling locomotion morphology for traversing different terrains by visually recognizing the type of terrain. In our prior work, we used mean shift image segmentation and probabilistic terrain

classification with three different terrain types [16]. However, this approach was susceptible to high levels of false alarms and a high degree of computational needs making the robot unattractive for real life deployment.

In this paper, we present a novel terrain perception system for our bioinspired robot, Scorpio, to classify the terrain based on visual features and autonomously switch between rolling or crawling morphology locomotion modes in relation to perceived terrain. Our proposed approach uses a Speed Up Robust Feature (SURF) description along with color information. The feature extraction will be followed by use of Bag of Word method (BoW) and Support Vector machine (SVM) for terrain classification. Bag of Word method is used for generating histogram representation of the frequency of extracted features. In the learning phase, sample images of different terrain are taken and classified to create a database.

In our experiments, the Scorpio robot captured images using its on board camera at the end of every rolling and crawling cycle when the robot comes to rest in a default gait best suited for capturing terrain images. In the testing phase, once the input images are captured, it undergoes the same process as in the training phase and logical decision is taken by comparing with the database. We use Arduino Pro Mini as the on board embedded client controller for our Scorpio robot. The terrain perception system in our case is implemented in the server, a personal computer using Open CV open source image processing platform.

Terrain perception is studied under the surface texture identification problem in computer vision domain. Surface texture identification is usually done using the color information of the captured images, but this method alone is not at all efficient. The influence of shadows and reflected light from the outdoor environment makes the terrain perception more tedious. There are also possible cases such as different terrain with the same color or vice versa. For example, the images of a green carpeted concrete floor and a savannah might be identified as the same color. The terrain perception systems using force sensors [2] and Kinect depth sensors [3] performed better in such scenarios. However, one of the constraints in our project is to design a robot that is less than 10 cm in diameter, thereby making use of Kinect or multiple force sensors highly impossible. Therefore, a light weight low-resolution camera is used in our Scorpio robot for image capturing. The usage of low-resolution cameras also reduces computational complexity for robot controller. Surface texture identification using local feature extraction is preferred in our case because local features are invariant to rotation, shadows, viewpoints, and brightness among others.

2. Scorpio Robot: System Overview

Mobile robots capable of traversing rough terrains are highly desired for numerous applications including search, rescue, reconnaissance, and surveillance missions. A vast majority of biological species realize locomotion by crawling using legs, but additional locomotion gaits such as climbing, jumping, and rolling are also realized using the legs as well. Most of the studies into reconfigurable robotics tend to add new mechanisms to realize additional locomotion gaits. Such an approach would result in increased size, a high level of computational complexity, and numerous controllability hurdles.

In this work, we are developing a novel class of bio-inspired self-reconfigurable architecture deriving inspiration from *Cebrennus rechenbergi*, a species of huntsman spider inspired design approaches to achieving rolling-crawling-climbing locomotion [17,18]. *Cebrennus rechenbergi* was found in the deserts of Morocco which, in addition to crawling like any other spider species, propels itself off the ground and moves its legs in a flic-flac somersault motion to go uphill, downhill, or on level ground. Unlike the natural counterpart, our design for the Scorpio robot (Figure 1) has only four legs for realizing both rolling as well as crawling locomotion modes due to consideration for power optimization. The system specification of Scorpio robot is listed in Table 1. Three motors are used in each leg of the Scorpio robot where the actuation is achieved using JR ES 376 sub micro analog servomotors. These servomotors are capable of delivering a torque of 2 Kg·cm on supply of 4.8 V. The three servo motors in each leg rotate about yaw, pitch, and roll axis, respectively. Hence, each limb possesses three degrees of freedom. A Pololu servo controller is used for driving all the servo motors

which is interfaced with the central embedded microcontroller. Pololu micro maestro servo controllers are USB programmable and are capable of signaling 18 servo motors. The outer structural framework of Scorpio robot is 3D printed where the material used for fabricating the Scorpio robot is polylactic acid which is a biodegradable polymer. The limbs of the Scorpio robot are made hollow to minimize the weight. Hence, we could actuate the limbs with sub micro servos which are capable of delivering small torque. The hollow limbs also allow to give the internal circuitry connections efficiently.

The crawling locomotion gait is achieved by simultaneous rotation of two motors about pitch and yaw axis as shown in Figure 1a. The design of the robot allows for the legs to be folded to form a perfect circular shape which helps to achieve the rolling locomotion gait. The rolling motion of the robot is nearly five times as fast as crawling motion on a near flat terrain offering improved time and power efficiency in such terrains. Scorpio transforms to the rolling mode by placing the pair of posterior legs parallel to the ground and curling itself to form a circular disk. The rolling motion is achieved using a single degree of freedom movement in each of the limbs as shown in Figure 1b. The simultaneous motion of a single motor in the bottom lying pair of limbs helps the robot to lift itself up from the ground thereby shifting the center of gravity to upper portion, and ultimately results in rolling the whole body forward due to its own weight. This cycle of basic motion primitives is continuously repeated to achieve a continuous rolling motion. The crawling motion is achieved by the synchronous use of actuators in all four limbs. Scorpio robot uses an Ai-Ball camera and Inertial Measure Unit (IMU) sensors for perceiving its environment. The IMU sensors used on board Scorpio combines gyro sensor and accelerometer which helps in achieving closed loop rolling and crawling locomotion gaits. In addition, the IMU sensor is also used to identify whether the robot is in a stable position or it requires to initiate a recovery gait to recover from an unstable posture after a fall. Figure 2 shows the hardware architecture of our Scorpio robot.



Figure 1. The figure showing Scorpio Robot: (a) The Scorpio robot in crawling locomotion gait; (b) The Scorpio robot in rolling locomotion gait.

Table 1. Specifications of the Scorpio Robot.

Components	Specifications
Controller	Arduino Mini Pro 328
Servo Motor	JR ES 376
Servo Controller	Pololu micro Maestro 18-Channer
Sensors	WiFi Ai-Ball Camera; MinIMU-9 v2
Battery	LiPo 1200 mAh
Communication	Xbee Pro S1
Full body material	Polylactic acid or polyclactide (PLA)
Diameter (rolling form) in mm	168 mm
L × W × H (crawling form) in mm	230 mm × 230 mm × 175 mm
Weight	430 g

Scorpio uses an Ai-Ball camera for capturing the images of the terrain being traversed. The Ai-Ball camera has a resolution of 640×480 (VGA), focal length of 200 mm, frame rate of 30 fps and a view

angle of 60 degrees. One of the major benefit of using the Ai-Ball camera is that they are compact and lightweight, weighing less than 100 g with a diameter of 30 mm. This makes the device more suitable for application in robot platforms like Scorpio that are small with limited on board computational power. It also allows for direct wireless mode of connectivity. The wireless interface between the camera and computer is established using IEEE 802.11 (Wi-Fi) communication. Because the Ai-Ball camera is a low resolution one, the captured images are often blurred, noisy, and contain false color as shown in Figure 3. The captured images have different color balance each 8×8 pixels block and, the images captured have a latticed noises of 8×8 pixels. Arduino Pro Mini based on ATmega328 compact microcontroller with a clock speed of 16 MHz is used as the central embedded controller for the robot.

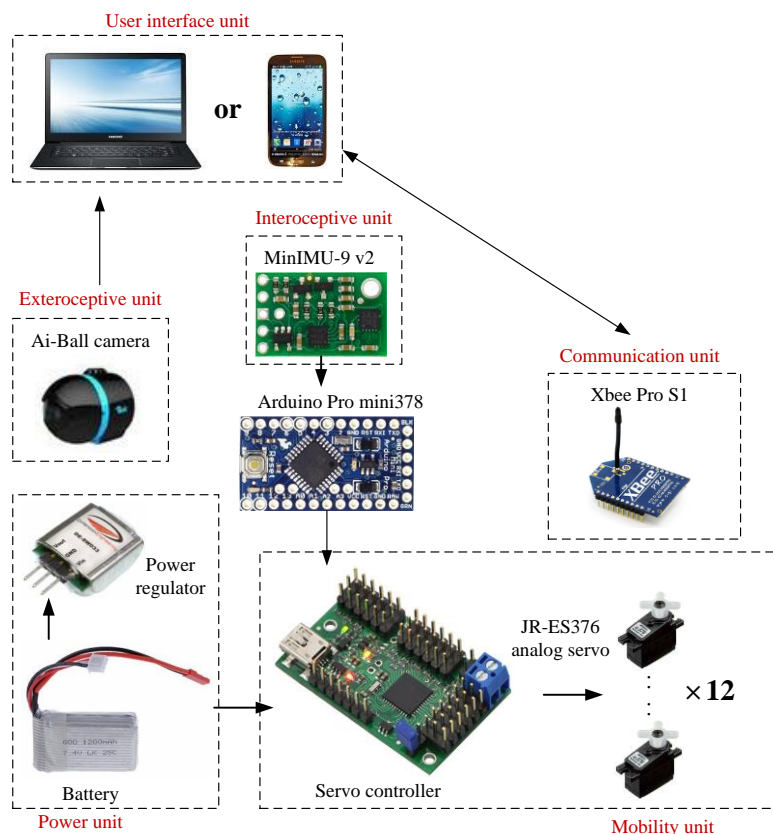


Figure 2. The hardware architecture of Scorpio Robot.



Figure 3. The images taken from the Ai-Ball camera.

3. Terrain Perception

3.1. System Overview

This section presents the overview of our developed terrain perception system. Figure 4 presents the flow chart of our developed terrain perception system. The terrain perception system contains a learning phase for database establishment and a testing phase for real time terrain classification. Since the feature extraction and classification required a high degree of computing power, we used a client-server approach where a remote server was responsible for handling the computational intensive vision processes and communicates the final decision to the client running on the robot while the client implements the appropriate locomotion gaits and ensures seamless transmission of the perceived sensor data to the server.

In the learning phase, a command for crawling or rolling is sent to the robot's client from the remote server. Once the command is received by the robot's client, it changes to crawling or rolling mode and starts locomotion in that mode. The server sends commands to take images of the terrain and the robot responds by interrupting the locomotion cycle, assuming an intermediate default gait phase most suited for taking images of the terrain and starts capturing visual data. This process is implemented both for rolling and crawling locomotion modes. By this process, the images of different terrains for both database establishment and classification are taken. Once the image is captured and sent to the server, it undergoes color and SURF feature extraction followed by bag of feature (BoF). SVM is used for database establishment. We choose 30 sample images of grass, gravel, wood deck, and concrete terrains for training SVM and database establishment. For real time terrain classification phase, after BoF classification, logical comparison with a database is done to identify the terrain.

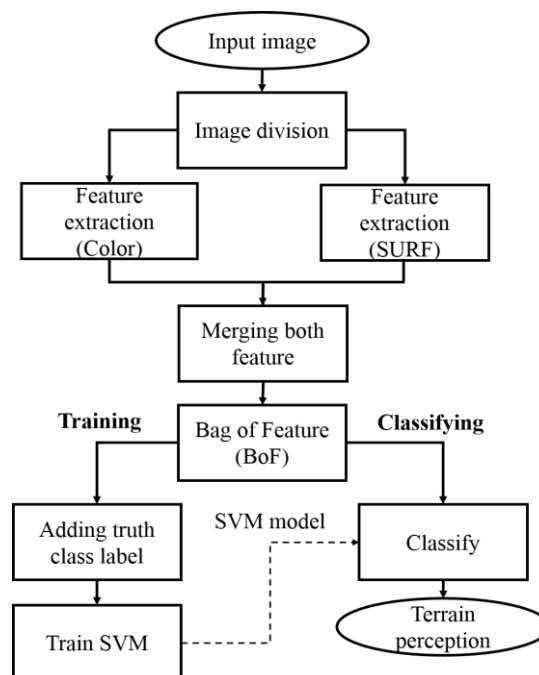


Figure 4. Flow chart of our terrain perception system.

3.2. SURF (Speed Up Robust Feature) Descriptor

SURF is a novel method for rotation and scale invariant feature interest point extraction and descriptor proposed by Herbert Bay [19]. The SURF algorithm runs faster in interest point location and matching than similar feature extraction algorithms like SIFT [20]. Typical SURF descriptors give 64 length feature vectors. In our experiments, we use only SURF descriptor. The interest point is

identified using grid sampling or dense interest point locators. SURF algorithm uses Hessian matrix to detect key points. Given a point $u = (x, y)$ in a given image M , the Hessian matrix $H(u, \sigma)$ at point u and scale σ , is defined as follows,

$$H(u, \sigma) = \begin{pmatrix} L_{xx}(u, \sigma) & L_{xy}(u, \sigma) \\ L_{xy}(u, \sigma) & L_{yy}(u, \sigma) \end{pmatrix} \quad (1)$$

where, $L_{xx}(u, \sigma)$, $L_{yy}(u, \sigma)$, and $L_{xy}(u, \sigma)$ are the second derivatives of the image. Once the key points and descriptors are extracted from the training set, then the descriptors are clustered into N centroids using standard K-means unsupervised clustering algorithm. The extracted descriptors are then treated as a “Bag of Words”.

3.3. BoW (Bag of Words)

Bag of Words is a method used to classify the image descriptors [21]. Basically, BoW will result in a histogram representation of the image features. This method considers the image as a document and the feature vectors or descriptors as the words in the document. BoW counts the frequency of occurrences of each word in the document and generates the histogram representation. Since, the histogram only represents the frequency of occurrences, the spatial information of the image descriptors will be unknown. Before the BoW representation, the image descriptors should be clustered and quantized to code words. These code words can be considered the “visual words” that can be represented using BoW. This quantization is achieved using K-Means clustering algorithm. K-Means clustering is an unsupervised learning algorithm which clusters n number of feature vectors to K number of cluster centers. Each cluster center will be encoded to a code word which becomes the visual words for BoW representation.

3.4. SVM (Support Vector Machine) Classifier

SVM is a classifier which classifies data vectors [22] that represents data vectors in higher dimensional space trained with the positive and negative examples which helps to identify a hyperplane that becomes the decision boundaries in the multidimensional space. SVM chooses the most optimal hyperplane by maximizing the Euclidean distance between the data points residing on the margin (Figure 5). The data points or vectors residing on the margin are called support vectors. With the help of support vectors, SVM identifies the optimal hyperplane which classifies the data points. Hence, this method is known as Support Vector Machine classification. In a two-dimensional plane, the hyperplane can be considered as a straight line. In some special cases, SVMs transform the data points to other dimensional spaces and calculate the optimal hyperplane for classification.

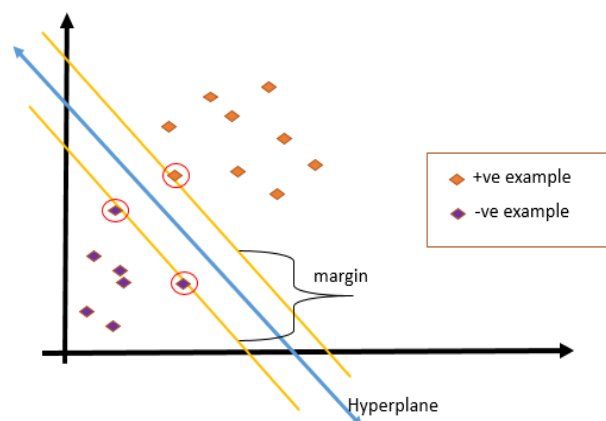


Figure 5. The Support Vector Machine Classifier.

3.5. Database Establishment

The efficiency of terrain perception depends on how effectively we build the database during the learning phase. In this work, we mainly concentrated on classifying and identifying four types of terrain examples. The first step in database establishment is image acquisition. The robot is made to roam in both crawling and rolling modes on different terrains to collect the images of various terrains. If robot is executing a rolling motion, it is transformed to default gait before capturing images. Capturing images in the default gait state will result in motion blur free data. In our experiments, once the image is captured by the camera, it undergoes the process of image division. This is done in order to handle situations wherein the input image may contain more than one terrain. Therefore, we divide a single input image into multiple small images based on terrain types recognized for efficient learning.

Our region of interest resides on the terrain near to the legs of the robot, therefore only an image strip of 180-pixel height is used for the processing and we exclude the upper portion of the image. This image strip is further divided to 80×80 pixel small snippets of images. These smaller image snippets then undergo feature extraction process. Figure 6 shows the image division method we implemented in our system. Since we are using smaller low resolution images for computation, the system is more robust in terms of speed of execution. In our experiments, we use both SURF and color based feature extraction techniques. Since, the images in our case are captured using a low-resolution camera—often noisy and of low quality. Use of both SURF and Color feature extraction help to enhance efficiency in both database establishment and real-time terrain classification stages.

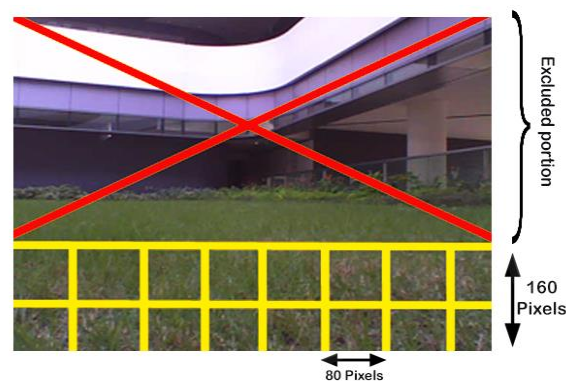


Figure 6. The image division operation.

In our developed approach, image division operation is followed by feature extraction using both grid sampling and color information. Since, in some situations, there are terrains which do not exhibit an identifiable pattern or texture. Therefore, to ensure the robustness of the system, we use dense interest point location [23] where local features are thoroughly extracted at a 5×5 pixel interval (Figure 7a). Unlike other feature extraction methods, dense interest point extraction uniformly chooses features throughout the image area. This method of feature extraction will help to achieve better description of the image. Once the features are extracted, the next step is to give a description for the extracted points.

In our work, we used SURF descriptor for describing the image features or interest points. SURF describes each of the interest point as a 64-length vector by luminance gradients around feature coordinates. The feature scale of scene recognition is of 3, 5, and 7 pixel radii. The next step to be applied is feature extraction using color information. The digital image representation of the images from the Ai-Ball camera resides in the RGB color space. The red, green, and blue values of an object represented in RGB color space is proportional to the amount of light irradiated on that object. The feature discrimination is tedious in RGB color space as the images we captured are subjected to varying light intensity. In order to exclude the problems caused by light intensity changes, we transform the images from RGB to HSV color space. In HSV color space, the images are expressed

in terms of hue, saturation, and value which makes the color information independent of intensity. The color information is important in terrain classification. However, it is highly inexpedient to recognize and discriminate a wide spectrum of colors using image processing technique.

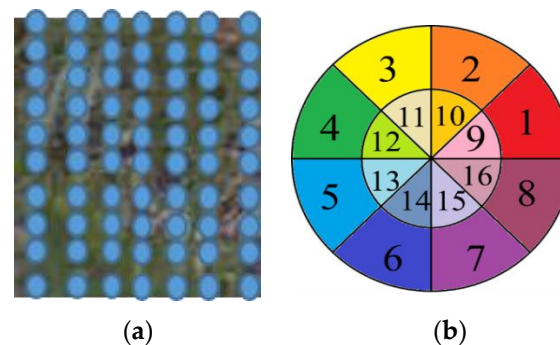


Figure 7. Figures showing the grid sampling and color reduction: (a) Grid sampling of image in an interval of 5×5 ; (b) Color reduction done by mapping 360 hue levels to 16 in HSV space.

Considering this fact, we reduce the image color scale by quantizing the whole range of hue values and mapping them to 64 colors. Hue value in HSV space ranges from 0–359. We divided the value into 30 levels and each level again subdivided into 12 (Figure 7b). We separated the tint colors and solid colors by saturation. The threshold is 128. At this point, we expressed 60 colors. When the saturation is low and the value is too high or too low, we ignored the hue due to the associated uncertainty. We regard pixels of saturation of less than 20, value of less than 20, and value of more than 200 as achromatic color pixels. As for the achromatic color pixels, when the pixel has a saturation of 0–20, the pixel is considered as black; when the pixel has a saturation of 20–128, the pixel is considered as dark gray; when the pixel has a saturation of 128–200, it is considered as light gray; and when the pixel has 200–255, it is considered as white (Figure 8).

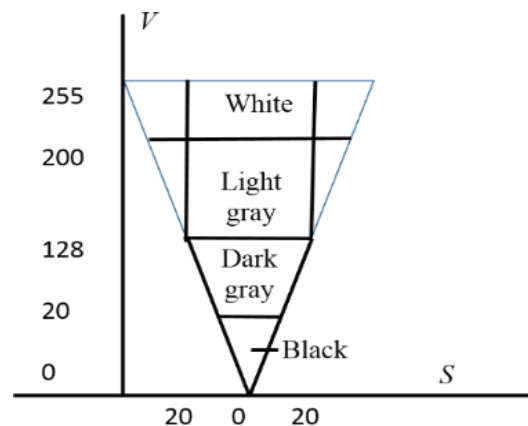


Figure 8. Figure explaining the achromatic color reduction.

We generate the histogram representation of surrounding pixels of a feature point as shown in Figure 9. The histogram thus generated can be treated as a vector. This vector is treated as the color feature vector of the captured image. In the next step, we combine the color feature vector and the SURF descriptor vector from each of the feature coordinates to form an augmented feature vector. However, the individual elements in the color feature vector have values larger than SURF descriptor values. Therefore, we normalize the entire vector values by dividing each element by the sum of all elements in the augmented feature vector. The augmented feature vectors extracted from an image represents its features completely.

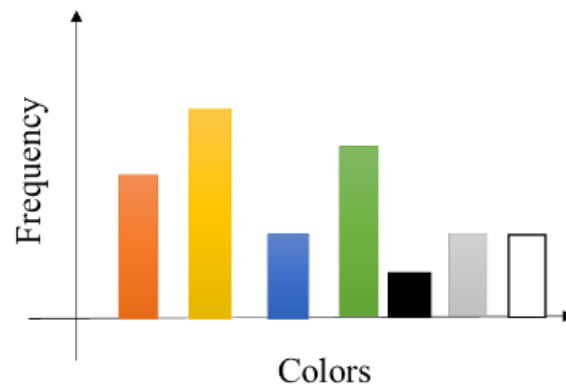


Figure 9. Color histogram.

After the augmented feature vectors in a terrain image are extracted, it undergoes a clustering process. This process is achieved using K-mean clustering algorithm. K-mean clustering algorithm encodes the feature vector to a code word in higher dimension. This code word can be a visual word. Figure 10 shows the clustering of augmented feature vectors in different cluster centers. In our experiments, we used 1000 visual words. Let us assume that we have a set of data, data $D = \{X_1, X_2, X_3, \dots, X_n\}$, in this scenario, the data is our augmented feature vector. Assume we have k number of clusters. $D = \{C_1, C_2, C_3, \dots, C_k\}$ Where, C is the cluster center. K mean clustering algorithm iteratively finds out the cluster centers by minimizing the sum of Euclidian distance from each data point to respective cluster centers.

$$L = \sqrt{\sum_{i=1}^N (X_i - C_i)^2} \quad (2)$$

However, this method increases the computational cost and takes a lot of time for processing. Therefore, we apply a K-Dimensional Tree (K-D tree) [24] for finding out the neighbor as shown in Figure 11. The whole of the data is represented in a binary tree data structure. The system searches for the nearest neighbor by traversing the binary tree. The advantage of this approach is that finding the nearest neighbor is quickly done since the system searches only the nearest regions from input data. After the clustering, the images are represented as histograms showing the frequency of occurrence of feature vectors. For this process, we used Bag of Features method of classification as shown in Figure 12. This histogram representation of each terrain image is taken as the training examples for the SVM classifiers.

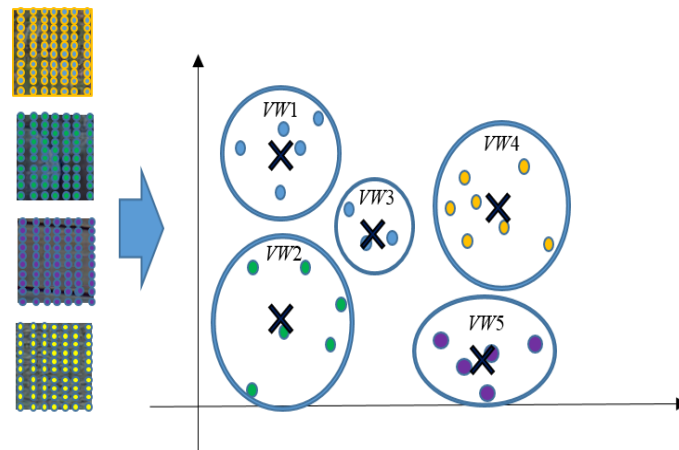


Figure 10. Visual word representation.

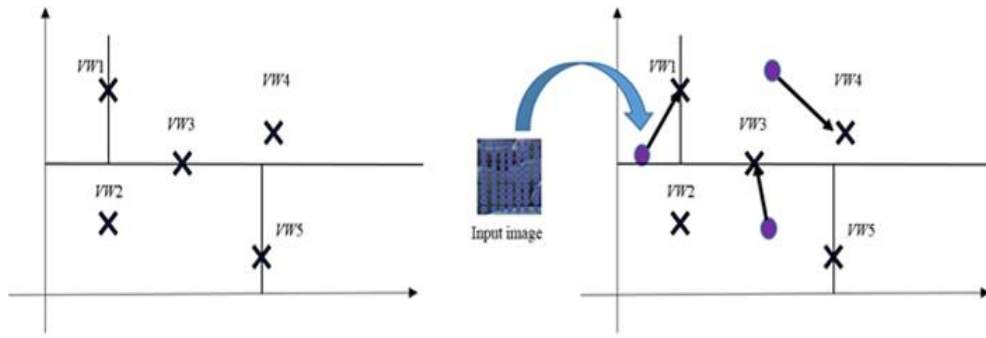


Figure 11. Figure showing the K-D tree representation and mapping to nearest visual word.

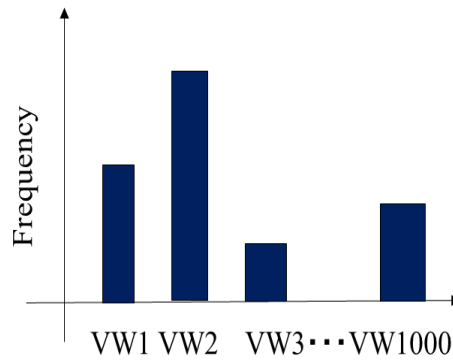


Figure 12. Figure showing BoW representation.

After the images are expressed using BoW, we designed a classification scheme which uses multi-class support vector machine (SVM) [22]. For the SVM classifier, radial basis function (RBF) is used as kernel function. We modeled the database using 30 images of each terrain such as grass, gravel, wood deck, and concrete. A single terrain image was decimated into 16 small images during the image division process. Therefore, the SVM classifier was trained using 1920 samples. We set the parameter for using SVM to find out the optimal decision rule for classification. SVM calculates w , the weight vector, by minimizing Equation (3). ξ_n is a slack variable, and ξ_n is added as the penalty. C is the weight of penalty. It is required to set the values for C . For larger values for C , the SVM do not perform effectively. In this case, we set C value as 3.0.

$$L = \frac{1}{2} ||w||^2 + C \sum_{n=1}^N \xi_n \quad (3)$$

In addition to choosing value for C , we need to set proper parameters for the kernel function $K(x_1, x_2)$ also. RBF is represented by the following Equation (4), where x_1 and x_2 are feature quantities. We need to set a value for σ . The larger the σ values, the larger the complexity of boundary lines will be. Here, we set σ as 1.0×10^{-7} .

$$K(x_1, x_2) = \exp\left(-\frac{||x_1 - x_2||^2}{2\sigma^2}\right) \quad (4)$$

4. Experiments and Results

After establishing the database for terrain perception, the real time testing of the robot is done. As in the learning phase, the images required for real time testing are taken by the robot in default gait state both in rolling and crawling modes. Then, each of the test images is subjected to feature extraction and description. As in the learning phase, the test images will undergo both color and grid

sampling. Then, we represent the input image as histograms by clustering the feature vectors followed by BoW and SVM based classification.

4.1. Conditions for the Experiment

For testing, 100 image samples were taken apart from the training database. These image samples include verities of grass, gravel, wood, and concrete terrains. All our tests involving the Scorpio robot were conducted at the SUTD campus premises. The maximum WiFi connection range between the OpenCV server and Ai-Ball camera on board the robot was 19 m. However, a stable Bluetooth link was only valid over 10 m range between the Arduino on board the robot and the OpenCV server. Hence, the communication radial range between the client running on the Scorpio robot and the OpenCV server was 10 m.

4.2. Terrain Classification Results

We analyzed the accuracy of our terrain classification system based on the 100 test images of grass, gravel, wood deck, and concrete. Each image was divided into 16 regions. Hence the total number of samples used for the experiment is 1600 terrain images. Table 2 shows the results of our terrain perception experiments with the Scorpio robot.

Table 2. Table showing results of terrain perception experiments involving Scorpio Robot.

Terrain	Grass	Gravel	Wood Deck	Concrete	Precision (%)
Grass	1579	2	0	18	98.7
Gravel	11	1544	49	67	92.4
Wood deck	1	30	1530	74	93.6
Concrete	9	24	21	1441	96.4
Recall (%)	98.7	96.5	95.6	90.1	

5. Conclusions

In this paper, we presented a new approach for vision based terrain classification, which is based on support vector machines, for a rolling-crawling self-reconfigurable robot using a low resolution camera. Since low resolution cameras are highly susceptible to noise, we implemented a terrain perception method that uses combined vector of luminance gradient and color information for successful implementation. Then, we represented the input image as histograms by clustering the feature vectors followed by BoW. For classification, our method used an SVM which is trained offline on the image data. The trained SVM classified the terrain images to grass, gravel, wooden deck, and concrete terrain classes online as shown in Figure 13. Our experimental results showed very good results, with an accuracy of over 90% for all test cases. We found that the proposed terrain perception system yielded maximum accuracy for grassy terrains, whereas the performance for other terrain depended on the clarity of the edge information from the image data. Future work would focus on integration of additional sensors to further improve the terrain classification performance beyond what is currently possible. Another aspect of future work would be to extend the terrain types to include mulch and soil.

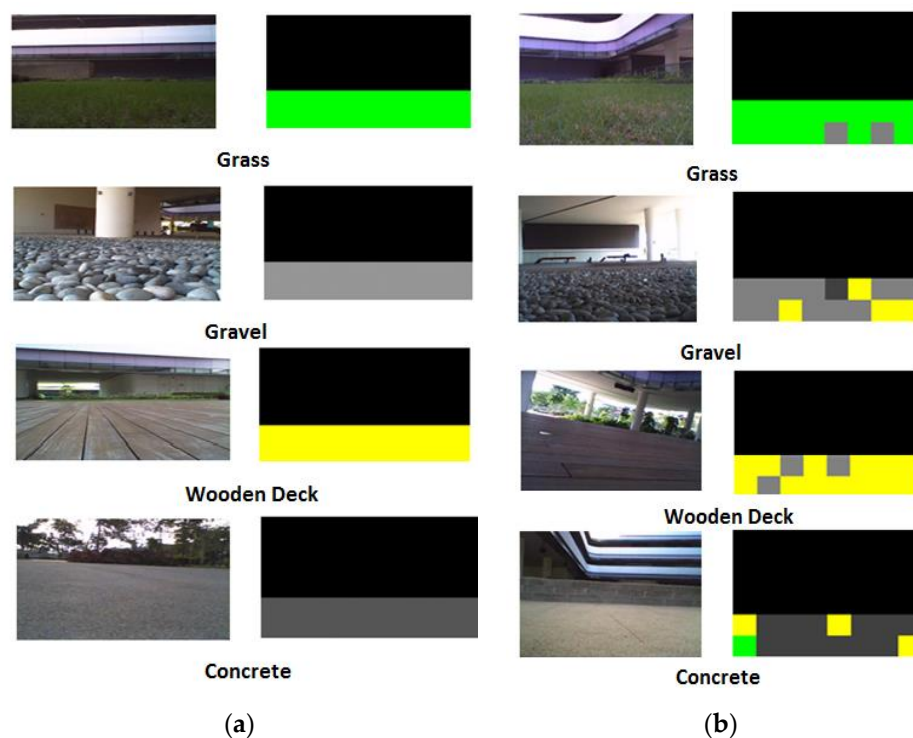


Figure 13. Terrain classification system, test images, and results: (a) High quality test images and results; (b) Low quality test images and results.

Acknowledgments: We acknowledge that this work is supported by the Temasek Laboratory and SUTD-MIT International Design Center at Singapore University of Technology and Design.

Author Contributions: F.M. and R.E.M. conceived and designed the experiments; F.M. performed the experiments; F.M. and R.E.M. analyzed the data; N.T., A.N., and T.P. assisted in preparing the paper. F.M., R.E.M., and N.T. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Weiss, C.; Frohlich, H.; Zell, A. Vibration-based terrain classification using support vector machines. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 4429–4434.
2. Best, G.; Moghadam, P.; Kottege, N.; Kleeman, L. Terrain classification using a hexapod robot. In Proceedings of the Australasian Conference on Robotics and Automation, Sydney, Australia, 2–4 December 2013.
3. Takizawa, H.; Yamaguchi, S.; Aoyagi, M.; Ezaki, N.; Mizuno, S. Kinect cane: An assistive system for the visually impaired based on three-dimensional object recognition. In Proceedings of the 2012 IEEE/SICE International Symposium on System Integration (SII), Fukuoka, Japan, 16–18 December 2012; pp. 740–745.
4. Zenker, S.; Aksoy, E.E.; Goldschmidt, D.; Worgotter, F.; Manoonpong, P. Visual terrain classification for selecting energy efficient gaits of a hexapod robot. In Proceedings of the 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Wollongong, Australia, 9–12 July 2013; pp. 577–584.
5. Lu, L.; Ordonez, C.; Collins, E.G., Jr.; DuPont, E.M. Terrain surface classification for autonomous ground vehicles using a 2D laser stripe-based structured light sensor. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009, St. Louis, MO, USA, 10–15 October 2009; pp. 2174–2181.

6. Ascari, L.; Ziegenmeyer, M.; Corradi, P.; Gaßmann, B.; Zöllner, M.; Dillmann, R.; Dario, P. Can statistics help walking robots in assessing terrain roughness? Platform description and preliminary considerations. In Proceedings of the 9th ESA Workshop on Advanced Space Technologies for Robotics and Automation ASTRA 2006, ESTEC, Noordwijk, The Netherlands, 28–30 November 2006.
7. Kim, J.R.; Lin, S.Y.; Muller, J.P.; Warner, N.H.; Gupta, S. Multi-resolution digital terrain models and their potential for Mars landing site assessments. *Planet. Space Sci.* **2013**, *85*, 89–105. [[CrossRef](#)]
8. Filitchkin, P.; Byl, K. Feature-based terrain classification for littledog. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, 7–12 October 2012; pp. 1387–1392.
9. Moghadam, P.; Wijesoma, W.S. Online, self-supervised vision-based terrain classification in unstructured environments. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2009, SMC 2009, San Antonio, TX, USA, 11–14 October 2009; pp. 3100–3105.
10. Otte, S.; Laible, S.; Hanten, R.; Liwicki, M.; Zell, A. Robust Visual Terrain Classification with Recurrent Neural Networks. In *Proceedings*; Presses Universitaires de Louvain: Bruges, Belgium, 2015; pp. 451–456.
11. Thrun, S.; Montemerlo, M.; Aron, A. Probabilistic Terrain Analysis for High-Speed Desert Driving. In Proceedings of the Robotics: Science and Systems, Philadelphia, PA, USA, 16–19 August 2006; pp. 16–19.
12. Bajracharya, M.; Tang, B.; Howard, A.; Turmon, M. Learning long-range terrain classification for autonomous navigation. In Proceedings of the IEEE International Conference on Robotics and Automation, 2008, ICRA 2008, Pasadena, CA, USA, 19–23 May 2008; pp. 4018–4024.
13. Sugiyama, Y.; Hirai, S. Crawling and jumping by a deformable robot. *Int. J. Robot. Res.* **2006**, *25*, 603–620. [[CrossRef](#)]
14. Chen, S.C.; Huang, K.J.; Chen, W.H.; Shen, S.Y.; Li, C.H.; Lin, P.C. Quattroped: A Leg—Wheel Transformable Robot. *IEEE/ASME Trans. Mechatron.* **2014**, *19*, 730–742. [[CrossRef](#)]
15. Ijspeert, A.J.; Crespi, A.; Ryczko, D.; Cabelguen, J.M. From swimming to walking with a salamander robot driven by a spinal cord model. *Science* **2007**, *315*, 1416–1420. [[CrossRef](#)] [[PubMed](#)]
16. Sinha, A.; Tan, N.; Mohan, R.E. Terrain perception for a reconfigurable biomimetic robot using monocular vision. *Robot. Biomim.* **2014**, *1*, 1–11. [[CrossRef](#)]
17. Tan, N.; Mohan, R.E.; Elangovan, K. A Bio-inspired Reconfigurable Robot. In *Advances in Reconfigurable Mechanisms and Robots II*; Springer: Gewerbestrasse, Switzerland, 2016; pp. 483–493.
18. Kapilavai, A.; Mohan, R.E.; Tan, N. Bioinspired design: A case study of reconfigurable crawling-rolling robot. In Proceedings of the DS80-2 20th International Conference on Engineering Design (ICED 15) Vol. 2: Design Theory and Research Methodology Design Processes, Milan, Italy, 27–30 July 2015; pp. 23–34.
19. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [[CrossRef](#)]
20. Juan, L.; Gwun, O. A comparison of sift, pca-sift and surf. *Int. J. Image Process. (IJIP)* **2009**, *3*, 143–152.
21. Csurka, G.; Dance, C.; Fan, L.; Willamowski, J.; Bray, C. Visual categorization with bags of keypoints. In Proceedings of the ECCV International Workshop on Statistical Learning in Computer Vision, Prague, Czech Republic, 11–14 May 2004; pp. 59–74.
22. Weston, J.; Watkins, C. *Multi-Class Support Vector Machines*; Technical Report CSD-TR-98-04; University of London: London, UK, 1998.
23. Tuytelaars, T. Dense interest points. In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 2281–2288.
24. Bentley, J.L. K-d trees for semidynamic point sets. In Proceedings of the Sixth Annual Symposium on Computational Geometry, Berkley, CA, USA, 7–9 June 1990; pp. 187–197.

