Visual Terrain Classification for Selecting Energy Efficient Gaits of a Hexapod Robot

Steffen Zenker[1], Eren Erdal Aksoy[2], Dennis Goldschmidt[2], Florentin Wörgötter[2], and Poramate Manoonpong[2]

***Abstract*— Legged robots need to be able to classify and recognize different terrains to adapt their gait accordingly. Recent works in terrain classification use different types of sensors (like stereovision, 3D laser range, and tactile sensors) and their combination. However, such sensor systems require more computing power, produce extra load to legged robots, and/or might be difficult to install on a small size legged robot. In this work, we present an online terrain classification system. It uses only a monocular camera with a feature-based terrain classification algorithm which is robust to changes in illumination and view points. For this algorithm, we extract local features of terrains using either Scale Invariant Feature Transform (SIFT) or Speed Up Robust Feature (SURF). We encode the features using the Bag of Words (BoW) technique, and then classify the words using Support Vector Machines (SVMs) with a radial basis function kernel. We compare this feature-based approach with a color-based approach on the Caltech-256 benchmark as well as eight different terrain image sets (*grass, gravel, pavement, sand, asphalt, floor, mud, and fine gravel*). For terrain images, we observe up to 90% accuracy with the feature-based approach. Finally, this online terrain classification system is successfully applied to our small hexapod robot AMOS II. The output of the system providing terrain information is used as an input to its neural locomotion control to trigger an energy-efficient gait while traversing different terrains.**

I. Introduction

Terrain surfaces influence the locomotion of legged robots. Therefore, recognition of different terrains is required to control their gait for efficient energy consumption. Current studies in terrain classification employ a combination of passive and active sensors (like stereovision, 3D laser range, and tactile sensors) [1], [2]. For example, Ascari et al. [1] combined texture analysis of statistical vision content (stereo camera and laser line scanner system) with tactile information produced by a sensing system from the robot feet to estimate terrain characteristics. Lu et al. [2] used a laser stripe-based structured light sensor with a camera to extract spatial relations between different gray levels for terrain surface classification. While such combined sensor systems show good performance, they require more computing power, produce extra load to legged robots, and/or might be difficult to install on a small size legged robot.

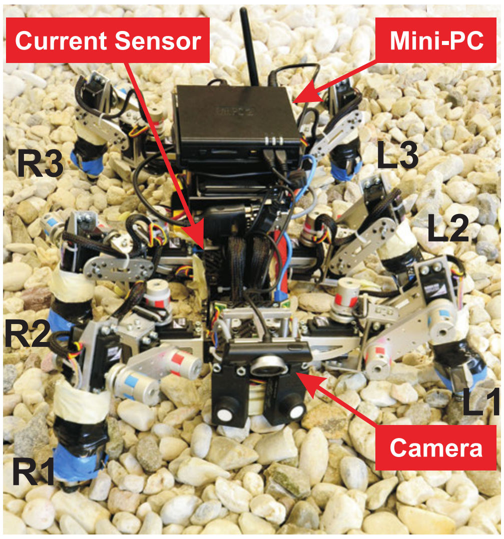

Fig. 1: The hexapod robot AMOS II. A USB camera with 720p, 30fps, and auto-focus is attached to the front chassis frame. The camera is connected to a mini-pc with a 2GHz Intel Atom which is set on top of the back chassis. The current sensor installed inside the body of AMOS II is used to measure the energy consumption on different terrains.

In the domain of computer vision, terrain classification is essentially considered as surface texture identification and can be grouped into three different techniques based on spectral [3], [4], color [5], [6], and feature [7], [8]. Spectral-based methods focus on spatial frequencies of texture distributions. Unlike color-based methods, local features are invariant to scale, rotation, brightness, and contrast and hence became popular in image classification. A few studies [7], [8] showed the success of scale invariant local features for terrain classification and robot control with some restrictions due to motion blur of the captured images. For example, Khan et al. [7] used a wheeled mobile robot to obtain smooth motion while Filitchkin [8] used a legged robot where the robot must stop its locomotion while taking an image to prevent camera blur effect. In both approaches, generalization of the classifier on a different benchmark was also not justified.

The work described here continues in this tradition with the extension of the use of visual information to trigger the most energy-efficient gait at a reasonable frame rate while our hexapod robot AMOS II (see Fig. 1) is continuously

This research was supported by Emmy Noether grant MA4464/3-1 of the Deutsche Forschungsgemeinschaft and Bernstein Center for Computational Neuroscience II Göttingen (BCCN grant 01GQ1005A, project D1).

[1] S. Zenker is with the Faculty of Mathematics and Computer Science, Institute of Computer Science, Georg-August-Universität Göttingen, Goldschmidtstrasse 7, 37077 Göttingen, Germany s.zenker@physik3.gwdg.de

[2] E. E. Aksoy, D. Goldschmidt, F. Wörgötter, and P. Manoonpong are with the Faculty of Physics, Third Institute of Physics - Biophysics, Georg-August-Universität Göttingen, Friedrich- Hund-Platz 1, 37077 Göttingen, Germany (eaksoye, degoldschmidt, worgott, poramate)@physik3.gwdg.de

traversing different terrains without the need of blur-free images and stopping its motion as required for the system shown in [8]. Our online terrain classification system uses only a monocular camera with a feature-based terrain classification algorithm. For our algorithm, we investigate two popular local feature detection methods, Scale Invariant Feature Transform (SIFT) [9] and Speed Up Robust Feature (SURF) [10]. SIFT and SURF detectors generate various key points (i.e., features) which are to some degree invariant to scale, rotation, viewpoint, and illumination changes. These key features are more descriptive and hence reliable color-based visual cues. We represent the extracted key points with visual words by using the k-means algorithm. The Bag of Word (BoW) method then encodes frequency of each visual word by a histogram. Classification of terrain images is completed by applying Support Vector Machines (SVMs) [11] that extract the maximum-margin hyperplane between the histograms. The final output of the terrain classification algorithm is then used as an input to neural locomotion control for activating an energy-efficient gait. A current sensor mounted on AMOS II (see Fig. 1) is used to measure the energy consumption for different terrains.

Although our approach makes use of traditional vision approaches, the main novelty is coming with continuous smooth walking while traversing different terrains and autonomously switching to an energy efficient walking pattern accordingly. This is a crucial step as the robot can adapt its walking behaviour to the environment. Unlike the other approach [8], with our system the robot does not have to stop its locomotion between gait transitions. In addition, we emphasize here the embeddedness of our system, i.e., a neural controller for gait generation and the online terrain classification system for energy-efficient gait selection are implemented on the mobile processor (mini-PC) of AMOS II (see Fig. 1).

The structure of the paper is as follows. In Section II, we introduce our online terrain classification system together with its performance. In Section III, we briefly present our hexapod robot AMOS II and its neural locomotion controller. In Section IV, we show experimental results of using the terrain classification system for energy efficient gait control of AMOS II on different terrains. Finally, in Section VI we provide conclusions and future works.

II. Online Terrain Classification System

Figure 2 illustrates the general overview of the online terrain classification system. It uses a single USB-camera (*Creative Live! Cam Socialize HD* with 720p, 30fps and auto-focus) installed on AMOS II to acquire terrain images. The camera is interfaced to a mini-PC *CompuLab fit PC 2i* with a 2GHz Intel Atom which is fitted on top of the back chassis of AMOS II as shown in Fig. 1. The acquired images are classified by a feature-based terrain classification algorithm in a supervised manner. The algorithm has two phases: Training and testing.

The training phase performs off-line where all images are first stored in the memory. Data flow in this phase is indicated by arrows as shown in the upper part of Fig. 2. It is initiated by the extraction of local features. Here, we apply two well known feature descriptors: Scale Invariant Feature Transform (SIFT) [9] and Speed Up Robust Feature (SURF) [10], details of which are provided below. SIFT and SURF detectors extract various features which are to some degree invariant to scale, rotation, viewpoint, and illumination changes. However, due to differences in perspective or occlusion the number of extracted features has a large variance even for similar (terrain) images. Hence, we apply the Bag of Words (BoW) method to generate vocabulary from the extracted features. In this method, all features are collected and clustered with the k-means algorithm which yields a comprehensive BoW dictionary with a certain size. The detected features of each image are encoded with the closest visual words in the dictionary. Next, a corresponding histogram which shows the frequency of each visual word in images is computed. The training loop is completed by Support Vector Machines (SVMs) [11] with radial basis function kernel. The SVM method, details are provided below, essentially maximizes the geometric margin between the computed histograms.

The testing phase of terrain images includes the same steps. Given a test image again, local features are detected, encoded referring to the dictionary, and finally converted into a histogram. As the last step, the histogram is used to get a prediction from the previously computed classifier as indicated in the lower part of Fig. 2. This testing phase is implemented on the mini-PC of AMOS II and performs online when a new terrain image is acquired directly by the camera. Its output providing terrain information is further sent to neural locomotion controller of AMOS II for controlling its gait according to terrain as depicted in the lower part of Fig. 2.

We compare the performances of feature-based image

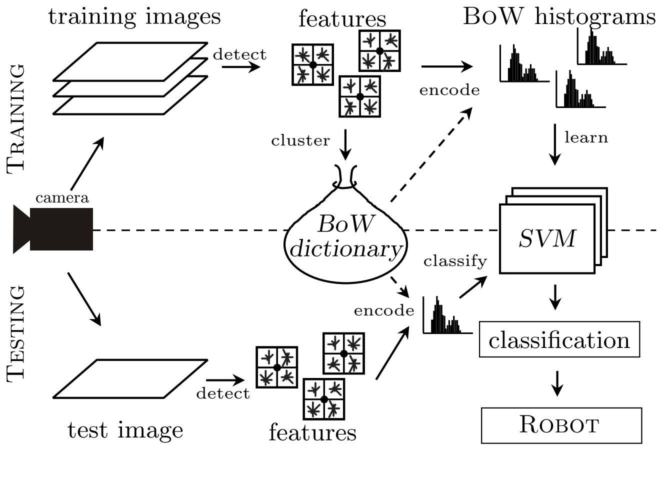

Fig. 2: Overview of the feature-based classification and gait control algorithm. Upper and lower parts indicate the data flow in the training and testing phases, respectively.

classifiers with that of color-based image classifiers. Since SIFT and SURF feature descriptors do not consider color cues, we compute the Hue histograms of the color variation in each image and classify this separately with SVMs. In the following subsections we provide a brief overview on the local feature descriptors (SIFT, SURF) and the image classifier (SVMs), respectively. Afterwards, we show the performance of the feature-based terrain classification algorithm with a comparison to a color-based classification algorithm.

A. Scale Invariant Feature Transform (SIFT)

SIFT [9] is a method that combines detection and description of interest points for further use for object recognition and image registration tasks. SIFT creates a scale-space by progressively applying different degrees of Gaussian blur to the input image at various scales. The interesting keypoints are detected by the Laplacian of Gaussian which is approximated in a faster way by computing the difference between Gaussians at two consecutive scales. To calculate keypoint locations, SIFT detects the local extrema at these DoG images by comparing each pixel with all its neighbors including those at consecutive scales. A further refinement step extracts the most distinctive keypoints by rejecting points that lie on edges or have little contrast. To achieve rotation invariance, each keypoint is assigned a consistent orientation based on local image properties. Gradient magnitudes and orientations at a 4×4 subregion around each keypoint are calculated and weighted by their distance to further form a histogram of orientations with 8 bins. Hence, the SIFT descriptor becomes a vector in 128 ($4 \times 4 \times 8$) dimensions.

B. Speed Up Robust Feature (SURF)

SURF [10] is based on similar concepts as SIFT, but is faster. SURF creates a scale-space by applying box filters of varying size to the input image. The scale space is analyzed by up-scaling the filter size instead of iteratively reducing the image size as implemented in the SIFT method. The SURF detector essentially relies on the Hessian matrix. To localize keypoints, SURF interpolates the local maxima of the determinant of the Hessian matrix in scale-space. Instead of gradients, a distribution of Haar-wavelet responses around the neighborhood of keypoints is used in SURF. As in the case of SIFT, a dominant orientation is assigned to each keypoint to achieve rotation invariance. The descriptor is calculated at a 4×4 subregion around keypoints. Within each subregion, Haar-wavelet responses are computed. The feature vector for the corresponding subregion is finally calculated by considering the sum of Haar-wavelet responses with their absolute values both in horizontal and vertical directions. This leads to a feature vector with 64 dimensions.

C. Support Vector Machines (SVMs)

SVMs [11] are popular and powerful supervised classification techniques. SVMs construct a hyperplane with the largest distances between training instances of different classes. Closest instances to the separating hyperplane are named as support vectors and the distance between those support vectors is called margin. The hyperplane is described by $w \cdot x + b = 0$, where w is the vector normal to the hyperplane, x is the training input vector, and b is a scalar bias. Determination of the optimal hyperplane with

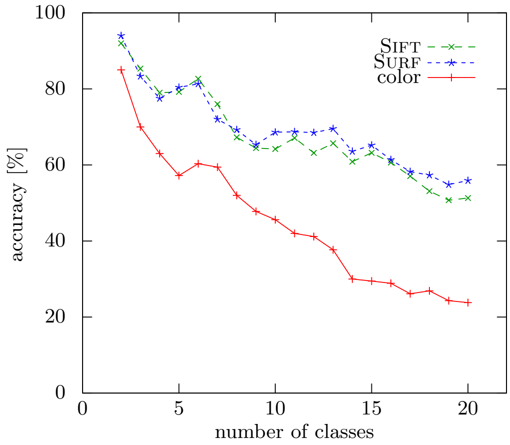

Fig. 4: Accuracy of the feature and color-based classifiers with the Caltech-256 image set. The SURF and SIFT feature descriptors show higher accuracy among 20 classes compared to the color-based classifier.

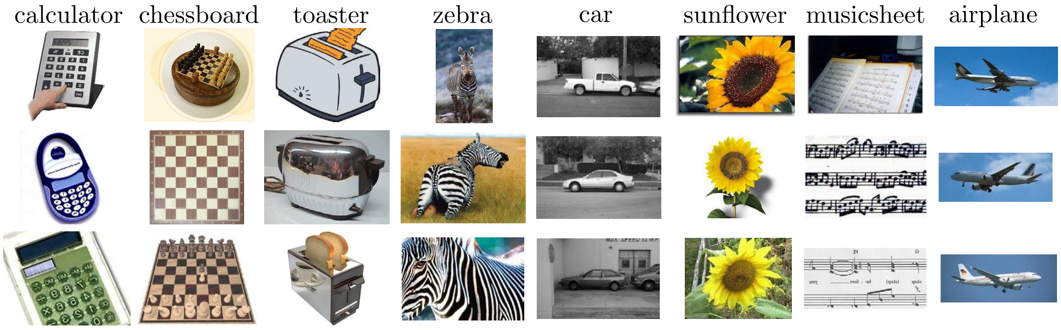

Fig. 3: Sample images from the Caltech-256 image dataset [12]. We chose 20 different image classes (e.g. calculator, chessboard, toaster, zebra, car, etc.) with at least 80 samples per category. Each image class has samples with cluttered background and large variations in perspective, color, and size.

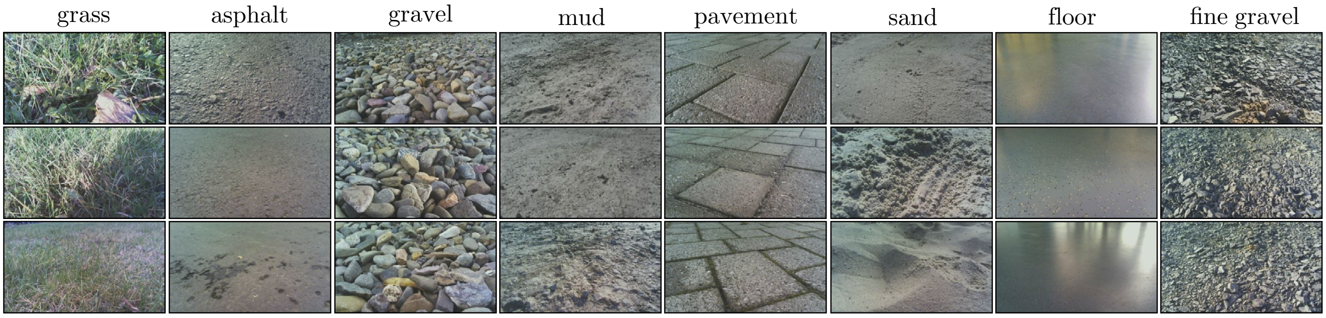

Fig. 5: Sample images from the terrain image set with eight different classes: grass, asphalt, gravel, mud, pavement, sand, floor, and fine gravel. Each terrain class has at least 80 samples taken under different lighting and weather conditions.

the maximum margin is calculated by minimizing $\|w\|$ through quadratic programming optimization. The optimum orientation of the separating hyperplane is then used to define decision boundaries between instances of different classes. To classify data points which are not linearly separable, SVMs use a trick that maps those points from the input space to a higher dimensional feature space. The mapping is based on a kernel function, e.g. radial basis function kernel as used in our experiments.

D. Classification with Benchmark Images

We empirically evaluated the performance of both, feature and color-based, classification algorithms with a goal independent benchmark: the Caltech-256 image dataset [12]. We arbitrarily chose 20 different image classes (e.g. calculator, chessboard, toaster, zebra, car, etc.) with at least 80 samples per category. As depicted in Fig. 3 each image class has samples with cluttered background and large variations in perspective, color, and size.

Figure 4 depicts accuracy of the feature and color-based classifiers with the Caltech-256 image set. Note that unless otherwise stated, accuracy means the number of correctly classified samples among all images. Here, 30 image samples from each class are used for training and the rest is for evaluation. It can be seen that in all cases the accuracy decreases as the number of classes increases. However, SIFT and SURF based classifiers exhibit similar behavior with almost 60% accuracy among 20 classes. As expected, a color-based classifier performs much worse than the others due to high color variation between image samples even in the same class. Consequently, these results show that feature-based classifiers are more distinctive even with high number of classes.

E. Classification of Terrain Images

Since there is no available terrain image benchmark, we created a dataset with eight different terrain surfaces: *grass, asphalt, gravel, mud, pavement, sand, floor*, and *fine gravel*. Each terrain class has at least 80 samples captured on different places. These sample images, having a resolution of 640 × 360 pixels, were taken with the integrated USB-camera under different lighting and weather conditions while the robot was standing still. Figure 5 shows three different images from each class to give an impression of differences between samples.

We applied both feature and color-based classifiers to our terrain image set. Figure 6 depicts the observed accuracy of each classifier as the number of training samples increases. The accuracy values of the SIFT and SURF feature descriptors are quite similar and the maximum for accuracy eight classes is almost 90% with both methods. The color-based classifier exhibits less accurate performance due to color differences between samples.

F. Computational Complexity

Not only the accuracy of a classifier but also the consumed time for the classification task is an important value to be considered. Despite of its limited accuracy, the color-based classification is a computationally cheap method. It yields frame rates up to 30 fps (independent of the image content but limited by the camera) and has a fast training phase which takes at most 10 seconds for the terrain image set.

The feature-based methods require more computational power due to the feature extraction step. Figure 7 shows the averaged number of key points detected in each terrain class.

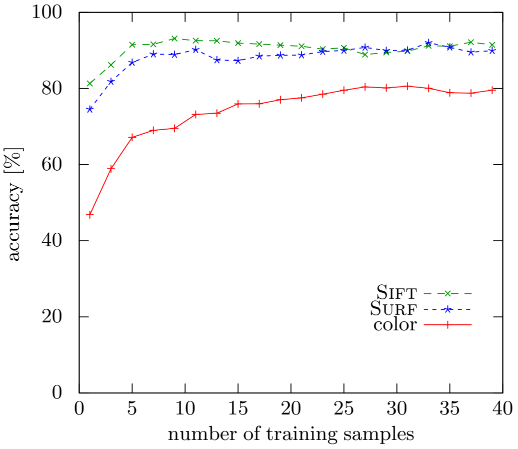

Fig. 6: Accuracy of the feature and color-based classifiers with the terrain image set. The maximum accuracy among eight classes is almost 90% and is observed with the SIFT and SURF feature descriptors.

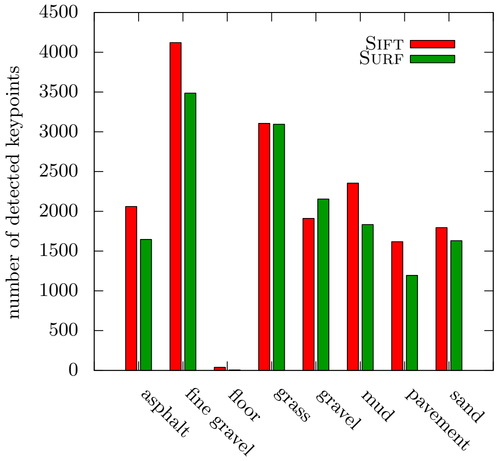

Fig. 7: Average number of key points detected in each terrain class.

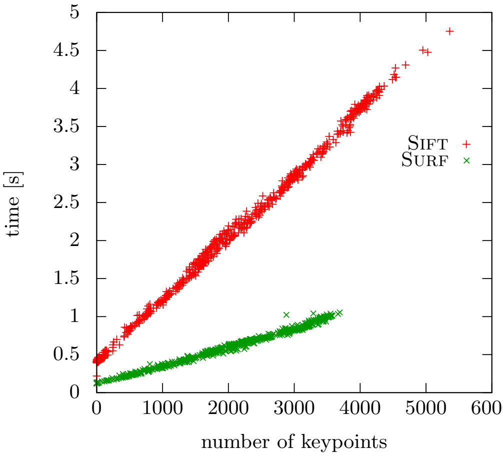

Fig. 8: The relation between the number of keypoints and required time for the SIFT and SURF feature descriptors.

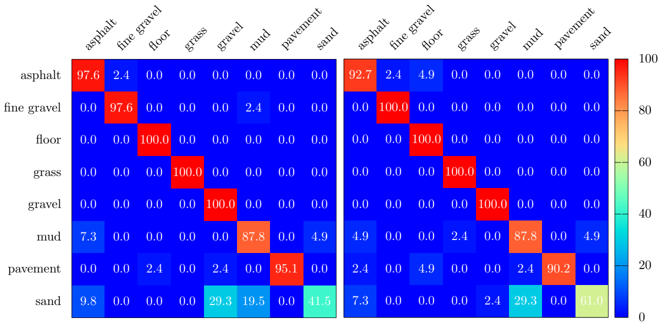

Fig. 9: Confusion matrix calculated with the SURF (left) and SIFT (right) method.

The minimum number of key points is extracted from the floor images due to little texture and strong light reflection (see Fig. 5). Note that during the online terrain classification less features can be detected due to blurry images or extreme lighting conditions observed while the robot is walking.

The relation between the number of key points and required time for the SIFT and SURF methods is given in Fig. 8. In the worst case the classification lasts 5 seconds (observed with the SIFT method) and in the best case 0.2 seconds (observed with the SURF method) with the compact hardware described in section II. Better hardware provides up to 3-5 fps even with the SIFT method. As indicated in Figs. 6, 7 and 8, the SURF feature descriptor outputs almost the same accurate results as SIFT, but requires less time. The SURF method yields frame rates up to 5 fps during the online classification depending on the number of features detected in the image. Figure 9 shows the final confusion matrix calculated with the SIFT and SURF methods, respectively. Here, 39 samples from each terrain class are used for training and the rest is for evaluation. Although the SURF feature descriptor shows fast classification, it has less true positive rate for sand. In contrast, the SIFT detector is a more robust against false classifications.

III. The Hexapod Robot AMOS II and Its Neural Locomotion Control

AMOS II (see Fig. 1) is a biomechatronical hexapod robot [14]. The robot body is inspired by the morphology of cockroaches. Its six identical legs are connected to the trunk which consists of two thoracic jointed segments. Body flexibility is assured by an active backbone joint. The 19 active joints (three at each leg, one body joint) of AMOS II are driven by servomotors *HSP-5990 TG HMI Digital* delivering a stall torque of $\tau = 2.9$Nm at 5V. In addition, the body joint torque has been tripled by using a gear to achieve a more powerful body joint motion. The thoracal-coxal (TC) joint controls forward/backward motion of the leg, the coxal-trochanteral (CTr) joint has the role of extension and flexion of the second limb and the motion of the third limb (up and down) is driven by the femoral-tibial (FTi) joint. Besides the motors, AMOS II has 19 embedded sensors perceiving its environment: two ultrasonic sensors at the front body part, six foot contact sensors in its legs, six infrared reflex sensors at the front of its legs, three light dependent sensors and one camera on the front body part, and one current sensor inside the body (see Fig. 1). All these sensors and motors are deployed for generating various behaviors (e.g., obstacle avoidance, climbing) [14], [15], [13], [16]. We use a Multi-Servo IOBoard (MBoard) installed inside the body to digitize all sensory input signals and to generate a pulse-width-modulated signal to control servomotor position. The MBoard is here connected to a mini-PC via an RS232 interface. Electrical power supply for all servomotors, the MBoard and all sensors is given by lithium polymer batteries with a voltage regulator producing a stable 5 volt supply.

Basic walking behavior including omnidirectional walking is driven by neural control (shown in Fig. 10) implemented on the mini-PC. The controller consists of three main modules:

- a Central Pattern Generator (CPG) including a postprocessing unit,
- a Phase Switching Network (PSN),

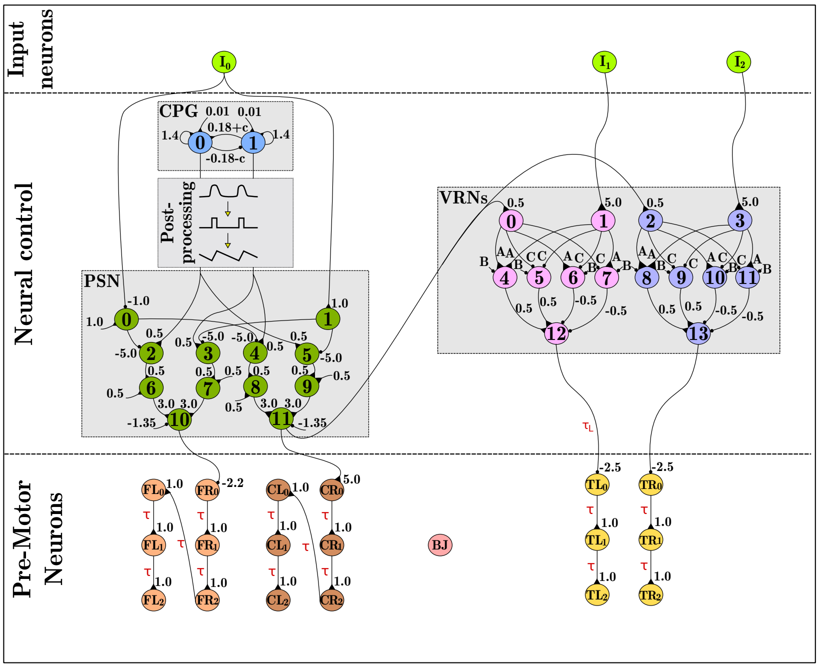

Fig. 10: Neural locomotion control of AMOS II consisting of three main modules (CPG, PSN, and VRNs). There are also input neurons ($I_{0,1,2}$) for controlling walking directions and pre-motor neurons (TR, TL, CR, CL, FR, FL, BJ) which directly command the position of the servomotors of the leg and backbone joints of AMOS II. BJ = backbone joint, TR(L) = TC-joints of right (left) legs, CR(L) = CTr-joints of right (left) legs, FR(L) = FTi-joints of right (left) legs. All connection strengths together with bias terms are indicated by numbers except those of the VRNs given by A = 1.7246, B = -2.48285, C = -1.7246. Note that postprocessing of the CPG outputs provides a triangle function which creates a smoother and more natural movement of the legs (see [13] for more details). Note that here the pre-motor neuron BJ controlling the backbone joint is not activated by neural control; i.e., the backbone joint is kept fixed. However, it can be controlled by a reactive backbone joint control module for climbing [14].

- two Velocity Regulation Networks (VRNs) working in parallel.

This neural control has been presented in [15], [13], [16]. Thus, here, we only briefly discuss these modules. All neurons used in the modules are modelled as standard additive non-spiking neurons. Their activity develops according to $a_i(t+1) = \sum_{j=1}^{n} w_{ij}\sigma(a_j(t)) + \Theta_i$; $i = 1, \ldots, n$, where n denotes the number of units, Θ_i represents a fixed internal bias term together with a stationary input to neuron i, and w_{ij} the synaptic strength of the connection from neuron j to neuron i. The output of the neurons is given by the hyperbolic tangent (tanh) transfer function $\sigma(a_i) = \tanh(a_i)$. Note that the update frequency of the neural controller is approximately 26 Hz.

The first module of the controller, the CPG, is a recurrent two-neuron network (see Fig. 10). Recurrent weights $w_{00,11}$ are set to 1.4 while weights between both neurons are determined by $w_{01} = 0.18 + c = -w_{10}$. c is a control input which is used to generate different walking patterns. Here we use the signal from the terrain classification system described above to set the control input value, thereby activating a predefined energy efficient gait with respect to the terrain.

The next module PSN is a generic feed-forward network, which reverses the phase of the periodic signals driving the CTr- and FTi-joints. As a consequence, these periodic signals can be switched to lead or lag behind each other by $\pi/2$ in phase in accordance with the given input I_0. The PSN has been implemented to allow for sideways walking, e.g., for obstacle avoidance (see [15] for more details on parameters and specific experiments on sensor-driven sideways walking).

The last two VRN modules are also simple feed-forward networks (see [15]). Each VRN controls the three ipsilateral TC-joints on one side. Because the VRNs behave qualitatively like a multiplication function [15], they have capability to increase or decrease the amplitude of the periodic signals by the magnitude of the inputs $I_{1,2}$. Consequently, the walking velocity of the machine will be regulated, i.e., the higher the amplitude of the signal the faster it walks. Furthermore they can be used to achieve more walking directions, like forward and backward movement (sign inversion of the multiplication) or turning left or right where the directions are driven by the preprocessed infra-red and light dependent resistor sensor signals through $I_{1,2}$.

At pre-motor neurons, we introduce $\tau = 0.8$ s and $\tau_L = 2.4$ s which are given delays between the outputs of the neural control and pre-motor neurons. This setup leads to biologically motivated leg coordination. In summary, while the CPG sets the rhythmic movements of the legs leading to different gaits, the PSN and the VRNs provide a certain steering capability. As a consequence, this neural controller enables AMOS II to move in different gaits as well as in any direction.

IV. Real Robot Experiments

In order to observe energy-efficient gaits for different terrains, we first categorize the eight terrain surfaces (described above, see Fig. 5) into four different groups:

- stiff ground = { floor, pavement, asphalt },
- loose ground = { fine gravel, sand, mud },
- rough ground = { gravel },
- vegetated ground = { grass }.

For each of these terrain groups, we let AMOS II walk from slow to fast gaits by increasing the control input value and we also calculate the electric energy consumption of each walking pattern as:

$$E = IVt, \tag{1}$$

where I is average electric current in amperes used by the motors during walking one meter. It is measured using the Zap 25 current sensor. V is electric potential in volts (here five volts). t is time in seconds for the travel distance (here one meter) and is used for task competition time for different gaits. Figure 11 shows the energy consumptions measured in these four terrain groups.

Each terrain group has a specific gait which leads to the lowest energy consumption. Figures 11a and b suggest using the control input value of 0.19 which produces a fast tripod gait on stiff and vegetated surfaces. Figures 11c and d suggest using the control input values of 0.06 and 0.04 which generate a tetrapod gait and a slow wave gait on rough and loose grounds, respectively.

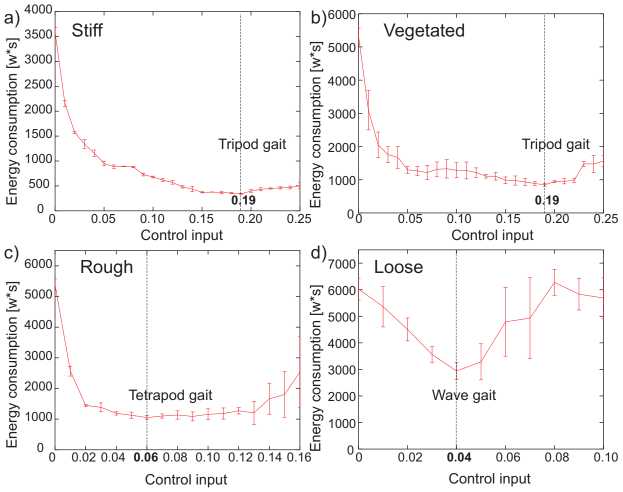

Fig. 11: Electric energy consumptions for different terrain groups and gaits. a) Stiff terrain. b) Vegetated terrain. c) Rough terrain. d) Loose terrain. Each measurement was repeated five times. Note that AMOS II started to slip when the control input value was higher than 0.19 for stiff and vegetated grounds and it cannot properly walk (i.e., it got stuck most of the time) when the control input values were higher than 0.16 and 0.10 for rough and loose grounds, respectively.

This allows mapping the four terrain groups to an energy efficient gait. For online terrain classification experiments, we use the SIFT method which can handle blurry images, hence the robot does not need to stop its motion. Although the SURF method is faster and yields equal accuracy in previous tests (see Figs. 4 and 6), real experiments have shown that the occurrence of false classifications increases dramatically due to motion blur especially for the SURF detector [17] while the robot is walking. Once the classification method responds to a new terrain image acquired directly by the camera, its output provides terrain information used to set the control input of the CPG, thereby triggering the corresponding pre-mapped energy efficient gait. These experimental results are shown in Fig. 12. We recommend readers to also see supplementary movies of these experiments at http://www.manoonpong.com/AIM2013/VTC.mp4 .

V. Conclusions

In this paper, we have analyzed local feature and color-based terrain surface classifiers using a single USB camera. SIFT and SURF feature descriptors as well as color histograms were compared on both the Caltech-256 benchmark with 20 classes and a terrain image set with eight classes. We observed up to 90% accuracy for the feature based classifiers. We also measured the energy consumption of AMOS II on different terrains while walking with different gaits. The most efficient gaits were mapped to the corresponding terrain surfaces. Finally, AMOS II used the SIFT method for online terrain classification to autonomously change its gait at a reasonable frame rate while traversing different terrains. This method is very robust to motion blur which allows the robot to continuously walk without any interruption in the walking performance unlike the other approach [8], thereby maintaining smooth forward locomotion. The here presented framework can easily be adapted to or implemented on any other walking robot for visually driven gait control to effectively walk on different terrains as well as to avoid dangerous areas like waters and icy or slippery surfaces. This approach can also be used for robots walking on textured surfaces, e.g. carpets with certain patterns since the detector needs only descriptive key features.

References

[1] L. Ascari, M. Ziegenmeyer, P. Corradi, B. Gamann, M. Zöllner, R. Dillmann, and P. Dario, "Can statistics help walking robots in assessing terrain roughness? Platform description and preliminary considerations," in *9th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, 2006, pp. 28–30.

[2] L. Lu, C. Ordonez, E. Collins, and E. DuPont, "Terrain surface classification for autonomous ground vehicles using a 2d laser stripe-based structured light sensor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, oct. 2009, pp. 2174 –2181.

[3] X. Liu and D. Wang, "Texture classification using spectral histograms," *IEEE Transactions on Image Processing*, vol. 12, pp. 661 – 670, 2003.

[4] N. Sebe and M. S. Lew, "Wavelet based texture classification," *International Conference on Pattern Recognition*, vol. 3, p. 3959, 2000.

[5] M. Pietikinen, T. Menp, and J. Viertola, "Color texture classification with color histograms and local binary patterns," in *IWTAS*, 2002, pp. 109–112.

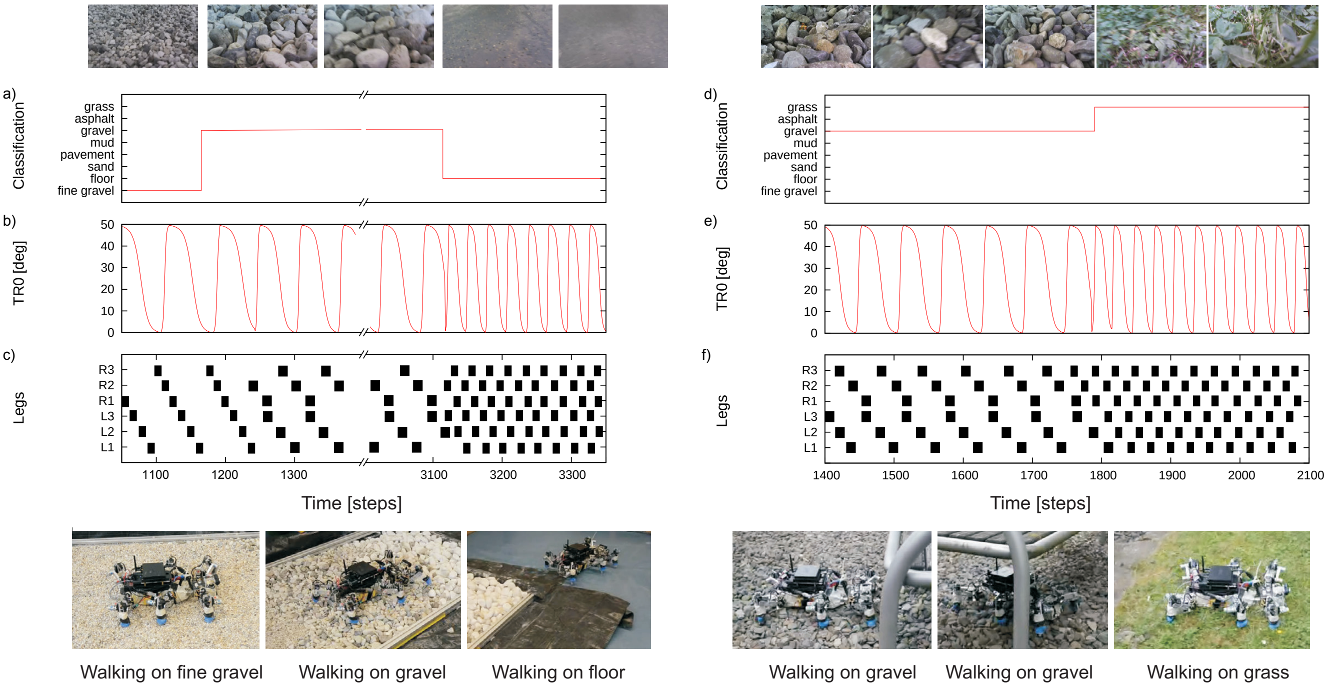

Fig. 12: Real-time data of the experiments of online terrain classification for energy efficient gait control. a) The output of the online terrain classification where AMOS II walked from fine gravel (loose ground) to gravel (rough ground) to floor (stiff ground). b) The TC-joint angle of the right front leg of AMOS II when the control input of the CPG was first set to 0.04 then 0.06 and finally 0.19 by the visual signal. It oscillated from very low to very high frequencies. c) Gait diagram (black boxes indicate the swing phase of a leg) showing that AMOS II used wave, tetrapod and tripod gaits during walking on the fine gravel, gravel and floor, respectively. d) The output of online terrain classification where AMOS II walked from gravel (rough ground) to grass (vegetated ground). e) The TC-joint angle when the control input was first set to 0.06 and then 0.19 by the visual signal. It oscillated from low to very high frequencies. f) Gait diagram showing that AMOS II used tetrapod and tripod gaits during walking on the gravel and the grass, respectively. Above pictures show snap shots from the camera used for the terrain classification while AMOS II walked on the different terrains. Below pictures show a series of photos of the locomotion of AMOS II during the experiments. $R1, R2, R3, L1, L2, L3$ mean right front, right middle, right hind, left front, left middle, and left hind legs, respectively (cf. Fig. 1).

[6] G. Paschos, "Perceptually uniform color spaces for color texture analysis: an empirical evaluation," *IEEE Transactions on Image Processing*, vol. 10, no. 6, pp. 932 –937, jun 2001.

[7] Y. Khan, P. Komma, K. Bohlmann, and A. Zell, "Grid-based visual terrain classification for outdoor robots using local features," in *IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems*, 2011, pp. 16 –22.

[8] P. Filitchkin and K. Byl, "Feature-based terrain classification for little dog," in *IEEE Inter. Conf. on Intelligent Robots and Systems*, 2012.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal Computer Vision*, vol. 60, no. 2, pp. 91–110, nov 2004.

[10] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *ECCV*, 2006, pp. 404–417.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, sep 1995.

[12] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep. 7694, 2007. [Online]. Available: http://authors.library.caltech.edu/7694

[13] S. Steingrube, M. Timme, F. Wörgötter, and P. Manoonpong, "Self-organized adaptation of simple neural circuits enables complex robot behavior," *Nature Physics*, vol. 6, pp. 224–230, 2010.

[14] D. Goldschmidt, F. Wörgötter, and P. Manoonpong, "Biologically inspired reactive climbing behavior of hexapod robots," in *Proceedings of the IROS*, 2012.

[15] P. Manoonpong, F. Pasemann, and F. Wörgötter, "Sensor-driven neural control for omnidirectional locomotion and versatile reactive behaviors of walking machines," *Robot. Auton. Syst.*, vol. 3, no. 56, pp. 265–288, 2008.

[16] P. Manoonpong and F. Wörgötter, "Adaptive sensor-driven neural control for learning in walking machines," in *Neural Information Processing*, no. 2, 2009, pp. 47–55.

[17] L. Juan and O. Gwun, "A comparison of sift, pca-sift and surf," *International Journal of Image Processing (IJIP)*, vol. 3, 2009.