Terrain Classification From Body-Mounted Cameras During Human Locomotion

Nantheera Anantrasirichai, *Member, IEEE,* Jeremy Burn, and David Bull, *Fellow, IEEE*

Abstract—This paper presents a novel algorithm for terrain type classification based on monocular video captured from the viewpoint of human locomotion. A texture-based algorithm is developed to classify the path ahead into multiple groups that can be used to support terrain classification. Gait is taken into account in two ways. Firstly, for key frame selection, when regions with homogeneous texture characteristics are updated, the frequency variations of the textured surface are analyzed and used to adaptively define filter coefficients. Secondly, it is incorporated in the parameter estimation process where probabilities of path consistency are employed to improve terrain-type estimation. When tested with multiple classes that directly affect mobility—a hard surface, a soft surface, and an unwalkable area—our proposed method outperforms existing methods by up to 16%, and also provides improved robustness.

Index Terms—Classification, recursive filter, terrain classification, texture.

I. INTRODUCTION

HUMANOID robots have been developed in recent decades to replicate human movement, and cameras are frequently employed as primary sensors, to emulate the way our eyes perceive the navigable environment. Visual information can enhance a robot's capabilities in terms of scene/object recognition and adaptation according to environment, aspects that are key for locomotion control and path planning. In this paper, we propose the use of such information to predict the type of terrain ahead of the robot via textures presented in single-view videos. In addition to humanoid robots, this paper could also benefit robots with multiple legs which are required to function in dangerous areas where wheeled robots are not suitable. It is also relevant to the design of aids for the visually impaired.

Several authors have previously proposed algorithms for terrain classification. In the main, these provide only binary classification, such as whether an area is road or vegetation [1]. When more complicated classifications are required, to recognize multiple classes or to provide probabilities of terrain prediction, a vision-based method is often combined with other sensors to confirm terrain types [2], [3], or a stereo-based vision system is used to assist geometric analysis of the near areas [4], [5]. While geometry-based approaches can provide the shape of the surface and are invariant to lighting conditions, texture-based visual analysis can focus on areas of interest and offer better resolution for finer classification tasks. There are only a few approaches that solely use on monocular video (color and textural features) for multiclass prediction [6], [7]. Although, both the visually-based and hybrid methods mentioned above perform well, most of them have only been proposed for wheeled vehicles.

A recognition technique using a bag of visual words was introduced in [8] for small legged robots. However, this paper employs video captured from a camera facing vertically downward, at a small distance from the ground. An example of a humanoid robot application is given in [9], where texture features are exploited to discriminate between only two classes, in order to determine whether the path is traversable. Generally humanoid and legged robots only exploit vision to control their walking [10], but not vice versa. The use of gait bounce signals and terrain classification is mentioned in [11] when different floor materials affect different limb motions. However, this works only with small and light-weight legged robots and does not exploit visual information.

To the best of our knowledge, this is the first paper on texture-based terrain identification for legged robots that exploits walking behaviors to improve classification performance. We present a textured-based terrain classification method for a legged system using a single camera that offers the following novel contributions.

1) A recursive temporal filter with adaptive filter coefficients computed from major uncertainties.
2) A compensation for the perspective foreshortening.
3) A new path consistency estimation.
4) A technique for performance improvement in terms of classification accuracy and computational cost using the motion characteristics of a biped humanoid robot.

The system used in this paper is demonstrated in Fig. 1, showing a camera located at a distance h from the ground and at an angle θ_x from the vertical axis. The proposed method is illustrated in Fig. 2. It employs a recursive filter where filter coefficients are updated adaptively from frequency projection and path consistency. The recursive filter ensures that information from a sequence of frames is weighted appropriately. Frequency projection—i.e., the textural change due to a forward moving camera, and path consistency—i.e., the possibility of combining different materials across frames, are

Fig. 1.   Human locomotion with a body-mounted camera.

Fig. 2.   Process of terrain classification for tracked regions.

employed to estimate the uncertainty of the information as it passes between frames. The algorithm begins by segmenting key frames into nonoverlapping regions. These key frames are selected from the sharpest frames of the walking cycle, as determined by the plot of sharpness values measured from the mean of highpass magnitudes. Only key frames are segmented, so as to reduce computational time, employing a wavelet-based watershed segmentation [12]. Many existing methods divide an image into equal-sized rectangular patches [6], [13]. Although these require less computational time for segmentation, an increased number of patches can lead to higher overall complexity in feature extraction and classification. Moreover, our method achieves better boundary definition for different textures, an aspect that is important in order to indicate where robots need to change their motions or directions.

Next our classification process is applied to each region and the associated classification probability is stored. These regions are tracked across successive frames until the next key frame, when the regions are updated. The outcome of classification for each region is accomplished using a recurrence relation of probability within a temporal sliding window defined adaptively according to walking cycle. A shorter window is used when walking fast, because each area and object disappears sooner than when walking slowly. We compute the decaying weights based on the major possible uncertainties due to walking and camera settings, which are motion blur, path consistency, and frequency variation caused by perspective view. These factors cause a change in texture characteristics and the information from the affected areas is weighted accordingly. The model used in the classifier is updated when new information from the upcoming path is obtained.

The algorithm is highly effective yet simple, and maintains the video processing load within the bounds of what is likely to be feasible for real-time computation. Moreover, unlike existing methods where obstacle detection is separately achieved using geometry-based algorithms (either with visual information [14], [15] or other sensors [16] or both [17]), our method is inspired by human vision which actually exploits monocular vision to perceive information at a distance. Obstacles, defined as unwalkable areas, are detected simultaneously with other terrain types using texture information.

The remainder of this paper is organized as follows. Related work is reviewed in Section II, and the proposed framework is overviewed in Section III. The proposed texture-based classification method, comprising a recursive probability estimation,
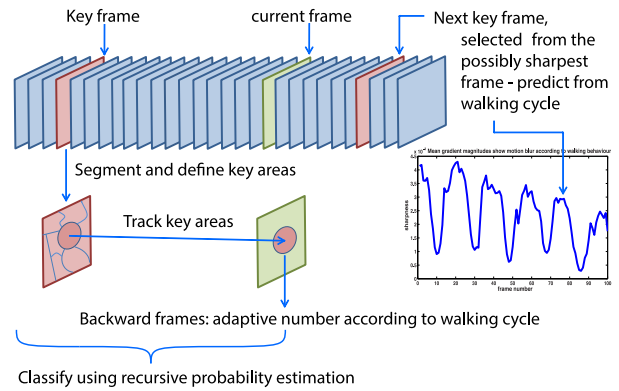
an uncertainty-based combination and a model updating technique, is explained in Section IV. The influences of locomotion are described in Section V, and the method performance is evaluated in Section VI. Finally, Section VII presents the conclusion of this paper.

II. TEXTURE-BASED TECHNIQUES FOR LOCOMOTION

In locomotion applications, textures extracted from frames of a video sequence were initially employed for optical flow calculation. Later, they were widely used for classification purposes [18]–[20], including terrain classification [21]. Classification of textures is, however, not straightforward due to the high variability of the data within and between images, particularly in natural scenes where effects, such as texture nonhomogeneity, light variation, and shadows are common.

Texture analysis is generally performed in the spatial domain and/or the transform domain to measure local variations in image intensity. In the spatial domain, statistical models are often applied, including second-order gray level statistics (e.g., contrast, angular second moment, and entropy or correlation), gray level run length statistics and co-occurrence matrices [22]. In the transform domain, conventional approaches make use of the Fourier power spectrum. Later, filter banks were employed to perform spatial-frequency analysis as these can extract more robust frequency characteristics for spatially and temporally varying natural images [20], [23]. Features can be extracted at a pixel level, but generally these perform poorly. Alternatively region-based techniques can be used that spatially group the pixels of an object or areas that contain similar characteristics [24] thereby achieving better performance particularly in the presence of noise. As locomotion applications generally require real-time processing, speeded up robust features [25] have also used to identify key features in texture areas [8], [26].

Extracted texture features can provide descriptions of a given terrain region, so are suitable for distinguishing navigable paths for autonomous vehicles. Angelova *et al.* [6] proposed a fast multiclass prediction algorithm which employs simple descriptors in different regions and applies more complicated approaches when more time is available. The texture-based method in [7] further applies a temporal label transfer where the results of previous frames are copied to

corresponding patches in the current frame. Recent reviews of visual terrain classification and techniques used in terrain traversability analysis can be found in [27] and [28].

For humanoid robot research, motion planning techniques primarily focus on 3-D geometric reconstruction using onboard stereo cameras [29]–[31] or laser range sensors [32]–[34]. Environments may also be simplified using edge detection to indicate the obstacles [10], [35]. However, such techniques have only been applied to indoor scenes. Recently, texture information has become a focus for improving locomotion. This has been inspired by human vision system where texture gradient cues are crucial for obtaining depth estimation in case of limited field of view (e.g., monocular vision) or when viewing at ranges beyond 2–3 m [36]. In [9], texture information in monocular images is employed together with a laser to identify the traversable areas, for a robot.

III. OVERVIEW OF THE PROPOSED FRAMEWORK

We classify the regions appearing in each frame into three classes: 1) hard surfaces (e.g., tarmac, bricks, tiles, deck, rough metal, and cement); 2) soft surfaces (e.g., grass, soil, sand, gravels, snow, and mud); and 3) unwalkable areas (e.g., static and moving obstructions). These classes could influence a robot's posture and dynamic stability when walking on the surface. The diagram of the proposed framework is illustrated in Fig. 3. The process starts by generating the instant walking pattern, using the sharpness value of each input frame. This is defined over a sliding window and is employed for selecting the next key frame and skipped frames. Each frame is thus processed as either a key frame, a skipped frame or a normal frame, as described below.

1) In normal frames, the regions of previous frame are matched to the corresponding areas in the current frame using a multiscale gradient matching method [37]. To reduce computation time, the regions in the previous frame that are classified as hard surfaces are warped together (similarly applied to soft surfaces). These are generally parts of a uniformly oriented surface from the one-point perspective which can exploit the same homograph parameters. Conversely, each unwalkable region generally contains an individual transformation, hence they are warped separately.

2) Key frames are segmented into nonoverlapping regions for which the texture characteristics of adjacent areas are different. We employ a wavelet-based watershed segmentation [12] where the gradient map is generated using the dual-tree complex wavelet transform (DT-CWT) [38]. The newly formed regions are used in the initial classification process. Next, the regions that are not classified as unwalkable areas are matched to the corresponding regions in the previous frames, while the unwalkable regions are processed using a refreshment scheme (described in Section IV-B). This is because detecting obstacles is considered to be very important; and thus, any results from previous frames should not affect the decision of these being obstacles in the current frame.
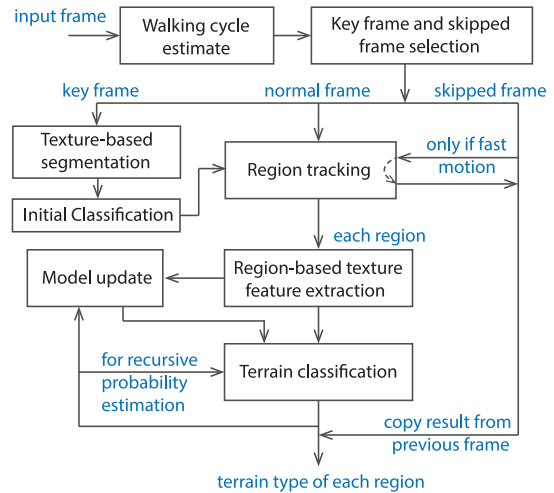

Fig. 3. Proposed terrain classification framework.

3) For skipped frames, a warping process is performed only under conditions of fast movement (e.g., running, rapid camera panning, etc.), since the associated large displacements can cause tracking difficulty. Here we use a simple threshold applied to the walking speed. If the current walking speed is faster than 5 km/h, the region tracking process will be applied. Possible improvements, such as using global displacements, will be developed in future work. None of the skipped frames are used in the classification process because the texture features from blurred frames could deteriorate the overall performance.

Next, for normal and key frames, the texture features are extracted for each region. These are employed in the classification process and stored for model updating when the classes of such regions are predicted at a sufficient confidence level. In the classification process, a support vector machine (SVM) [39] is employed to compute the probability of each region, and then the recursive probability estimation process is applied for the final decision. Further details of each stage in the algorithm are provided in the following sections.

IV. PROPOSED TEXTURE-BASED TERRAIN CLASSIFICATION

A. Texture Features

Texture is an efficient tool for characterizing various material properties, such as structure, orientation, roughness, smoothness, or regularity differences within an image. The texture features used in this paper are given in Table I and these include intensity level distribution, wavelet features, and the local binary pattern, extracted from each region. Only the intensity (Y) channel, extracted from the YC_bC_r color transformation, is used here.

For the intensity level distribution, five parameters are extracted, including mean, variance, skewness, kurtosis, and entropy. As one of the most important aspects of texture is scale, which provides both spatial and frequency information, a multiresolution approach is utilized based on wavelet features. We employ the DT-CWT [38] which employs two different real discrete wavelet transforms (DWT) to provide the real

TABLE I
LIST OF FEATURES USED IN CLASSIFICATION

Features	# dimensions
Intensity level distribution (ILD)	
Mean, Variance, Skewness,	
Kurtosis, Entropy	5
Complex wavelet transform (CWT)	
(4 decomposition levels)	
Mean, Variance of magnitudes	8
Mean of magnitudes of each sub-band	48
Local Binary Pattern (LBP)	
Histogram	59

and imaginary parts of the CWT. This increases directional selectivity over the DWT and is able to distinguish between positive and negative orientations giving six distinct sub-bands at each level, corresponding to $\pm 15°$, $\pm 45°$, and $\pm 75°$. This provides near shift-invariance and good directional selectivity. With four decomposition levels, the mean and variance of magnitudes across all subbands in each region produce eight features and those of each subband produce further 48 features (2×4 levels $\times 6$ subband/level).

The local binary pattern labels the pixels in an image by thresholding the neighborhood of each pixel, considering the result as a binary number [40]. Uniform patterns are generated using eight sampling points on a circle of radius 1 pixel. There are a total of 256 patterns, 58 of which are uniform, which produces 59 output labels. A histogram with 59 bins is obtained, and the frequency of each bin is used as one feature.

Other textural features were also investigated, including run-length measures [41], the gray-level co-occurrence matrix [22], and Gabor filter parameters [42]. However, the features in Table I were found to give the best terrain classification performance for body-mounted cameras. A SVM was employed to exploit these texture features to compute the probability of each terrain class.

B. Recursive Probability Estimation

One of advantages of continuous video is the large amount of information provided in both the spatial and temporal dimensions. In addition, it also provides a basis for temporal noise filtering. The simplest way to exploit temporal information is to use averaging across a group of frames. Provided that successive frames are accurately registered, the average is generally better than any of the individual frames. A recursive averaging process is employed here using exponentially decaying weighting of previous frames (w_k), in order to address the issue of error integration.

A multipass algorithm could be used to produce a classification result for a single frame, with a fresh sliding ($N+1$)-frame window comprising current, backward and forward frames. Unfortunately this would be computationally demanding because the registration process would need to be repeated for all N frames for each time shift. Also, buffering the forward frames is not ideal for a real time application. Therefore, we apply a recursive strategy to tracked regions. Equation (1) describes data processing for class c, $c \in \{1, 2, 3\}$, of region r in the current frame n with previous

N frames. This probability combination is similar to applying an Nth-order recursive filter with adaptive filter coefficients w_k

$$P_n^{r,c} = \sum_{k=0}^{N} w_k^r P_{n-k}^{r,c}. \tag{1}$$

1) Refreshment Scheme: In each key frame, the segmentation process is performed. This will either generate new regions corresponding to the first appearance of objects at distance or will update the existing regions to reduce tracking error. The regions that are classified as walkable regions (classes 1 and 2), are applied to the recursive probability estimation process with the best matched regions of the previous frames. In contrast, the $P_0^{r,3}$ of the region that is classified as unwalkable regions (class 3), is not combined with that of any regions of the previous frames, but acts as the refresh point and will be used in the recursive process for the next frame. This means, at the key frame: $N = 0$, $P_n^{r,3} = P_{n-0}^{r,3}$, and at the next frame: $N = 1$, $P_n^{r,3} = w_0^r P_{n-0}^{r,3} + w_1^r P_{n-1}^{r,3}$, and so on. We set this rule because the detected unwalkable regions in the key frame often correspond to new objects that appear on the walking path and require attention, whereas other regions are the continuous paths that are already visible in previous frames. Experiments show that the refreshment scheme can improve the classification accuracy by approximately 5% for general outdoor paths and by up to 10% for other complicated routes that have many obstacles.

C. Uncertainty-Based Combination

Leading on from Section IV-B, an important step in our algorithm is the probability combination. In this section, we explain the proposed scheme to compute w_k in (1). Generally the shape of the probability density distribution conveys the amount of certainty of information, i.e., a narrow distribution implies that most probability is concentrated in a narrow band, while a wide shape means the probability is spread over a wider range. This indicates confidence in a value. Therefore, the variance σ^2 of the probability distribution is used to compute the weight as shown

$$w_k^r = \left(\frac{1}{\sigma_{k,r}^2} \right) \Big/ \left(\sum_{k=0}^{N} \frac{1}{\sigma_{k,r}^2} \right). \tag{2}$$

The uncertainty occurring due to walking and environmental conditions is considered when estimating each $\sigma_{k,r}^2$. The variance of the probability distribution of the frame that is further from the current frame is generally larger since uncertain conditions increase. Firstly, the change of texture frequency distribution on the image plane along the direction of foreshortening is analyzed. Secondly, the consistency of the path ahead is exploited to adapt the weights used for probability combination. Later in Section V, we describe a technique which uses the walking cycle for key frame selection, adaptive sliding window calculation, and blur frame suppression.

1) Projective Frequency Compensation: Since the walking path appears from a one-point perspective, the corresponding areas in the different frames may contain different frequency characteristics depending on the position on the image where
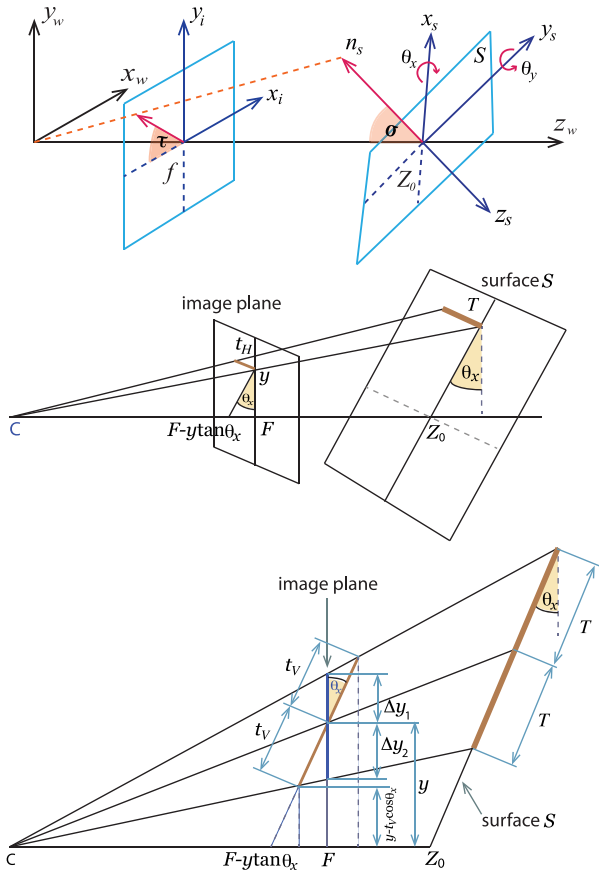
Fig. 5. Comparison of estimated frequency and actually frequency of texture slant surfaces. Top: horizontal frequency $\tilde{f}_y^H$. Bottom: vertical frequency $\tilde{f}_y^V$.

Fig. 4. Graphical representation used for estimating frequency on image plane. Top: projective geometry. Middle: geometry when $\theta_y = 0°$ and $\theta_z = 0°$ for f_y^H estimation. Bottom: 2-D geometry of y-z plane for f_y^V estimation.

the ground surface projects to. We therefore analyze the frequency change on an image plane due to a camera under forward motion. Fig. 4 shows the geometric system we use. Frequencies projected on the image plane in horizontal and vertical directions are estimated from the local frequency f_S on a ridge surface S of the terrain at an incident angle θ_x with the image plane. We assume $\theta_y = 0°$ and $\theta_z = 0°$ regardless of whether the head (camera) turns left-right or moves side-side during walking.[1] The horizontal projective frequency f_y^H in the image plane at y comes from the projection t_H of $T = 1/f_S$ as shown in Fig. 4 (Middle). Using the property of similar triangles, $f_y^H = 1/t_H$ can be estimated as 3, where F and Z_0 are focal length and surface distance from the camera, respectively

$$t_H = \frac{T}{Z_0}(F - y \tan(\theta_x)) \tag{3a}$$

$$f_y^H = \frac{Z_0 f_S}{F - y \tan(\theta_x)}. \tag{3b}$$

We estimate the vertical projective frequency f_y^V at position y from an average between the local projective frequency at just above and below point y, $f_{y1}^V = 1/(\Delta y_1)$ and $f_{y2}^V = 1/(\Delta y_2)$, respectively. The estimation starts with the projection t_V of T on the plane that is parallel to S and intersects

[1]The impact of camera panning on the overall performance of the system is described in Section VI-B1.
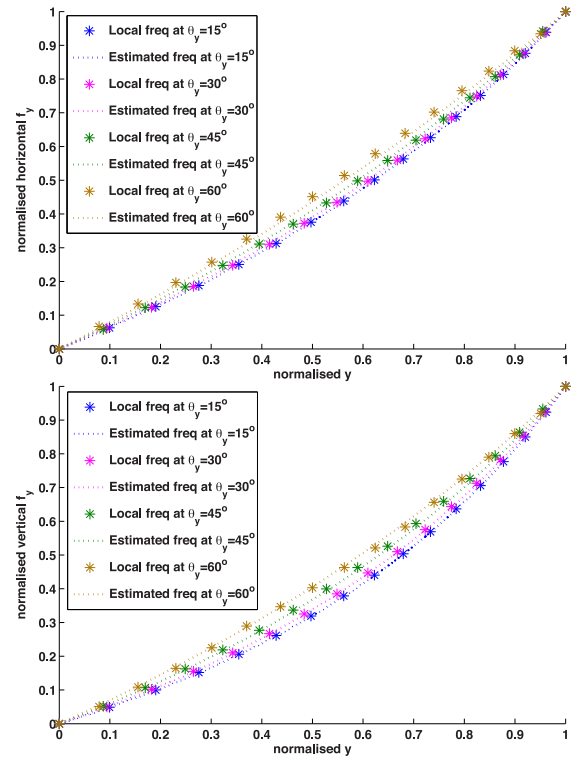
the image plane at y as shown in Fig. 4 (Bottom), which is computed as $t_V = (T/Z_0)(F - y \tan(\theta_x))$. Again, using the property of similar triangles, Δy_1 and Δy_2 are computed

$$\frac{\Delta y_1 \tan(\theta_x)}{F - y \tan(\theta_x)} = \frac{t_V \cos(\theta_x) - \Delta y_1}{t_V \cos(\theta_x) + y} \tag{4a}$$

$$\Delta y_1 = \frac{t_V \cos(\theta_x)(F - y \tan(\theta_x))}{F + t_V \sin(\theta_x)} \tag{4b}$$

$$\frac{F - t_V \sin(\theta_x)}{F} = \frac{y - t_V \cos(\theta_x)}{y - \Delta y_2} \tag{5a}$$

$$\Delta y_2 = \frac{t_V \cos(\theta_x)(F - y \tan(\theta_x))}{F - t_V \sin(\theta_x)}. \tag{5b}$$

Replacing t_V, f_y^V is estimated as shown

$$f_y^V = \frac{f_{y1}^V + f_{y2}^V}{2} = \frac{F Z_0 f_S}{\cos(\theta_x)(F - y \tan(\theta_x))^2}. \tag{6}$$

It can be seen that an inverse f_y^V is a parabolic function of y which agrees with the estimation in [43] where the nonlinear spatial frequency in the image plane caused from the perspective projection is approximated as the gradient of phases.

The values f_y^H and f_y^V are scaled in the range 0–1 across the image height, $\tilde{f}_y^\varphi = (f_y^\varphi - f_{(-M_h/2)}^\varphi)/(f_{(M_h/2)}^\varphi - f_{(-M_h/2)}^\varphi)$, $\varphi \in \{H, V\}$, where M_h is an image height. Thus, knowledge of Z_0 and f_S is not required. Fig. 5 compares the estimated frequency with the actual local frequency on the image plane when the surface is inclined at various θ_x. The surface is generated by projecting the rectangular image of a bi-directional sinusoid to each θ_x. The position in the y direction is also
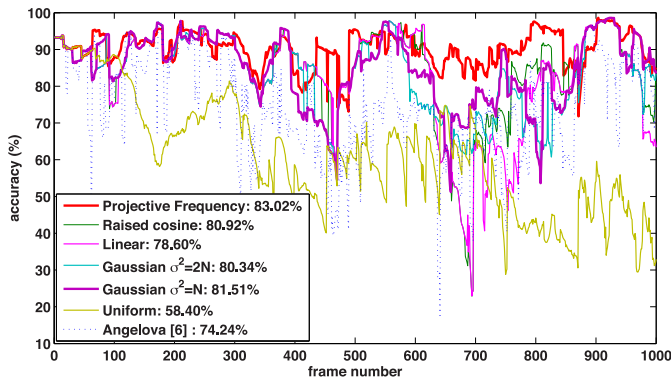
Fig. 6.   Classification accuracy comparison of 1000 frames of a video.

Consistent path   Inconsistent path

Error histograms of consistent path

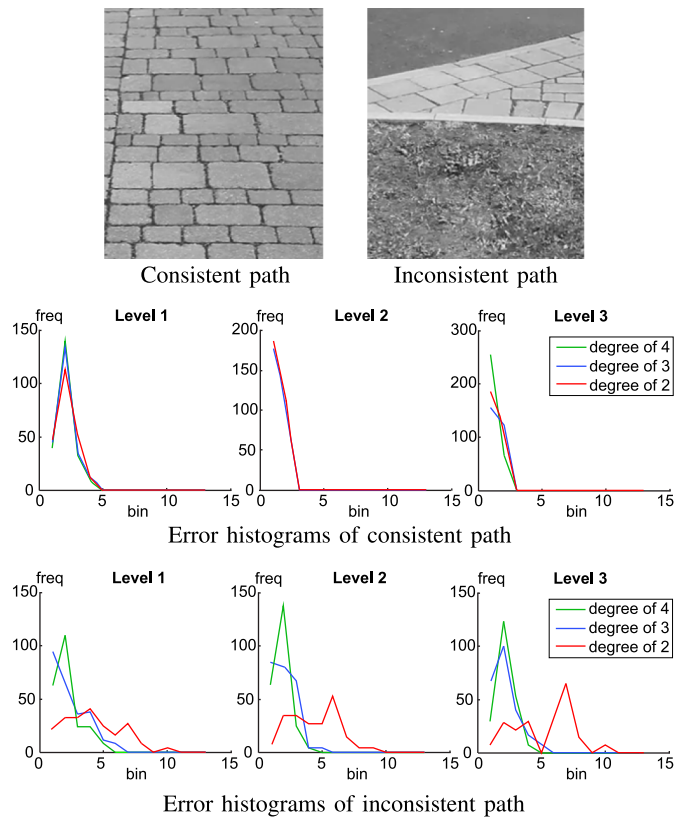Error histograms of inconsistent path

Fig. 7.   Comparison of histograms of consistent and inconsistent paths.

Fig. 8.   Weight at different probability of path consistency when window size is 20 frames.

normalized so that the results of different surface angles can be plotted on the same graph.

The variance of the probability at the frame which is at distance k from the current frame is estimated as shown in (7), where $(\partial \tilde{f}_y^H)/(\partial y)$ and $(\partial \tilde{f}_y^V)/(\partial y)$ are the differential frequencies between corresponding position y (centroid of region r) on frame $k-1$ and k in the horizontal and vertical, respectively. $\sigma_{0,r}^2$ is the variance obtained from the probability distribution of the training dataset. We use a library for support vector machines to compute the probability of each training data (further details about probability estimation can be found in [39, Ch. 8])

$$\sigma_{k,r}^2 = \sigma_{k-1,r}^2 + \frac{\partial \tilde{f}_y^H}{\partial y}\frac{\partial \tilde{f}_y^V}{\partial y}. \tag{7}$$

We compared the performance of our method using defined weights based on estimated projective frequency, with approaches using a uniform weight and several common decayed weight methods. The most similar weight distribution to ours is a Gaussian with variance equal to filter length, N. However, these weights are constant across the whole image, whereas ours are adaptive according to the position of the region being processed. A performance comparison is presented in Fig. 6. The accuracies (%) in this plot and the rest of this paper were computed from the number of pixels classified correctly (three classes) over the total number of pixels in each frame. It can be seen that the wider weight distributions benefit the prediction for the near areas, while the narrower shapes give better prediction for the far areas. Our method can clearly be seen to outperform the other approaches considered.

2) Path Consistency: A path that contains several different materials or obstructions makes tracking the corresponding areas between frames less reliable and adds more uncertainty to the probability estimation. In this case, more weight should be given to the current frame and the weights should decay faster than in the case of a consistent path. To estimate the probability of path consistency, a row-wise sum of highpass magnitudes of the DT-CWT is employed, before applying polynomial curve fitting with degree of 2 to construct error histograms (12 bins for each decomposition level). A degree of 2 is used because the projective frequency exhibits a near parabolic characteristic as shown in Fig. 5. Fig. 7 shows examples of both type of paths and their error histograms. With three decomposition levels, the values of
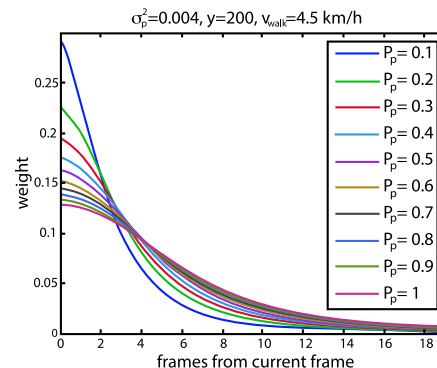
the row-wise sums and the error histograms of levels 1–3 are shown in columns 1–3 of Fig. 7, respectively. The plots also show estimations using polynomial curve fitting, with degrees 2, 3, and 4. The error histograms reveal that the best discrimination between consistent and inconsistent paths is achieved using degree 2 fitting.

The value of each bin of the error histogram is used as a feature for computing probability P_p by SVM classification. P_p is employed to adjust the weight as shown in (8) and an example of weights used for the sliding window of 20 frames with various P_p values is shown in Fig. 8. A small P_p indicates that the path ahead could be inconsistent, so the weight decays quickly for older previous frames. P_p can also be sent to the control system to assist awareness of obstructions or
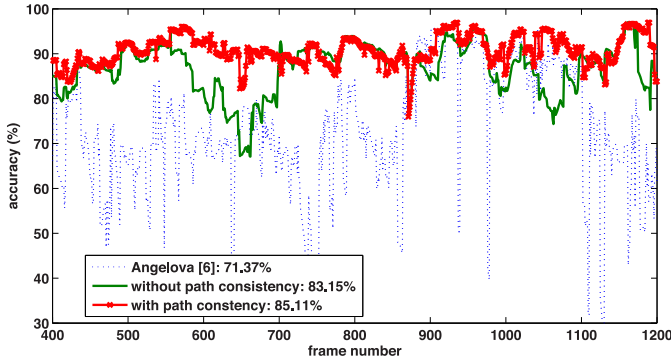
Fig. 9. Classification accuracy improvement when the path consistency approach is employed.
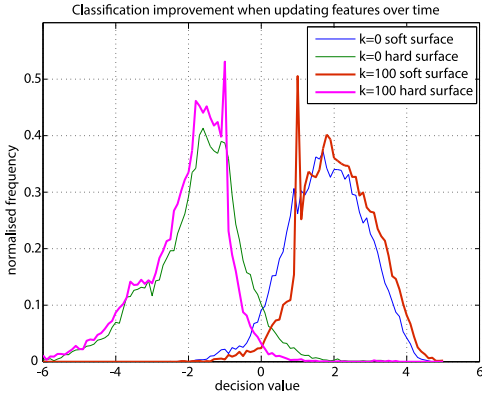

Fig. 10. Histogram of decision values shows improvement when applying model update (k is the number of new samples).

changes in terrain type. The boundaries between regions that are classified as different types can also be used to indicate locations which the robot should be made aware of. Operating adaptively on the instant video content, the classification performance is improved as shown in Fig. 9, particularly between frames 550 and 650 where the video contains movement from bricks to grass

$$\sigma_{k,r}^2 = \sigma_{k-1,r}^2 + \frac{1}{P_p} \frac{\partial \tilde{f}_y^H}{\partial y} \frac{\partial \tilde{f}_y^V}{\partial y}. \tag{8}$$

D. Classification Framework With Model Update

Areas that have been tracked from a distance generally show clearer frequency characteristics when they get closer to the observer, since the near areas appear sharper and different terrain types are easier to distinguish. The terrain classification framework therefore includes a parallel process in which the labels of all tracked features from far to middle ranges (row $1 - $ row $(3/4)M_h$) are updated with more accurate results from the classification of near areas (row $(3/4)M_h + 1 - $ row M_h). The new model is used to classify previously unseen appearing areas in the next frame. The system is tested by initially modeling 500 samples of soft surfaces and 500 samples of hard surfaces, and then 100 new samples of each type are included to recompute the model. By using 400 samples of each type for testing, the histogram of decision values is generated as shown in Fig. 10. The plot
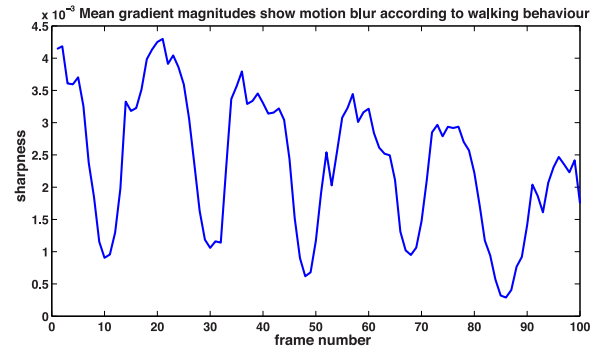

Fig. 11. Sharpness shows walking step ($C_w \approx 18$).

clearly reveals an improvement as there are fewer misclassified samples (showing smaller areas above and below decision value of 0 for hard and soft surfaces, respectively). As the quantity of training data can become very large, an ensemble classifier, along with a feature selection method, may be used to improve predictive performance and to reduce memory requirement [44].

V. INFLUENCE OF WALKING

When the shutter speed of a camera is not fast enough to capture stop motion, such as in low light conditions or with fast camera motion, some frames will exhibit high levels of motion blur which may alter measured frequency properties. Fig. 11 shows the sharpness value Ψ of each frame which is computed from the mean of highpass magnitudes $\Psi = (\sum_{l=1}^{4} \sum_{s=1}^{6} |\psi_{l,s}|)/n_{\text{all}}$, where $\psi_{l,s}$ is a DT-CWT coefficient of subband s at decomposition level l, and n_{all} is the total number of DT-CWT coefficients).

Fig. 11 clearly shows the points where the camera is moving faster. This indicates when the body vaults over the leg at each step during normal walking. Several motion blur removal methods have been proposed previously [45]; unfortunately deblurring often involves blur estimation or blind deconvolution which increases complexity and is not feasible for real time applications. In [46], a motion deblurring technique which does not rely on an iterative process was proposed for video stabilization. This technique replaces blurred pixels with the sharp pixels from the neighboring frames. A transform-based image fusion can also be employed for deblurring. The large highpass magnitudes of wavelet coefficients are selected amongst successive aligned frames to produce a sharp fused image in [47]. Here, we make use of blur information to reduce processing time and simultaneously improve overall classification accuracy via key- and skipped-frame selection. The performance of our terrain classification is tested with the proposed blur compensation and the deblurring methods presented in [46] and [47]. These results are presented in Section VI-A.

A. Key Frame Selection and Adaptive Window

A walking cycle, C_w (unit: frame/step), is employed to predict the next key frame in which the segmented regions are updated. It is also employed to adapt window size in our recursive method, i.e., $N = C_w$. If the walking cycle is short (e.g., walk fast or run), a shorter window is used.
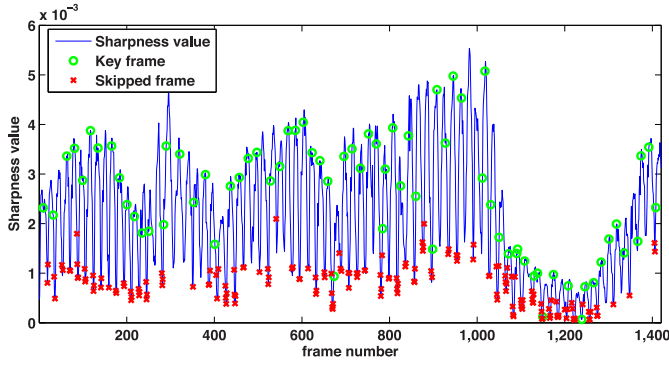
Fig. 12. Sharpness value plot with key frames and skipped frames.

We estimate C_w using the sharpness values, as shown in Fig. 11. The values obtained from the last 5 K_w frames (the number of frames just needs to be large enough to capture the walking behavior at that time), where K_w is a number of frames in one walking step. The initial K_w is 20 frames/step,[2] which is also reasonable to use if the walking speed is unknown.

The fast Fourier transform is employed to perform dominant frequency estimation. Consequently, the next key frame will be C_w frames from the previous key frame which is detected as a local maximum. Adaptively selecting key frames will prevent using blurred frames for segmentation. The maximum value of C_w is limited to the framerate for the case of missing the next local maximum (sharpest frame of the next step). If C_w reaches the maximal limit, the earliest frame that is sharper than the average sharpness value is selected as the key frame. Setting C_w equal to framerate ensures that the system is guaranteed to receive at least one key frame per second. Fig. 12 shows positions of key frames with sharpness value of each frame in the sequence.

B. Skipping of Blurred Frames

When an image is excessively blurred, it should not be employed in the classification process. This is because motion blur causes textural characteristics to change which could deteriorate the classification performance. It also influences the results for following frames when a recursive technique is employed. A frame skipping strategy improves overall system performance—not only is classification accuracy increased, but computational time is also reduced.

To identify the skipped frames, the estimated C_w is also employed. The local minima of the sharpness values are detected, and then the next $C_w \pm \lfloor K_w/8 \rfloor$ frames from the previous minimal point are checked (the maximum number of skipped frames in one walking step is limited to $K_w/4$). There are two cases for computing the threshold used for defining the skipped frames, which are when walking on: 1) the consistent path and 2) the inconsistent path. This can simply

employ the result of the path consistent prediction (described in Section IV-C2) of the previous frame.

1) *Case I—Consistent Path:* If the sharpness value of the current frame is less than the maximum of the local minima of previous 5 K_w frames, they are defined as skipped frames.

2) *Case II—Inconsistent Path:* If the sharpness value of the current frame is less than the average of the local minima of previous 5 K_w frames, they are defined as skipped frames.

This adaptive threshold ensures that the change of terrain characteristics from high detail texture, such as grass and bricks, to low detail texture, such as tarmac, will not cause over skipping. An example sequence showing skipped frames based on sharpness values and path consistency is shown in Fig. 12 (skipped frames are indicated by red cross). Frames #1050–1400 correspond to a tarmac road, so they have average sharpness values much lower than those of the previous frames where the path is made of bricks and grass. If the path consistency constraint is not employed, the sharpest frames around these frames might be defined as the skipped frames.

C. Blur Frame Suppression

The sharpness value is also employed to adjust the weight applied to each frame (1) as shown in (9), where g_k is the mean of the highpass magnitudes of frame k in the sliding window. The simple rule employed here is that information from the blurred frames is exploited less than for the case of sharp frames

$$\tilde{w}_k^r = \frac{g_k}{\max\left(\{g_i\}_{i=0}^N\right)} w_k^r. \tag{9}$$

VI. RESULTS AND DISCUSSION

The sequences used for testing were in 1920×1080 ($M_w \times M_h$) format with 24-bit RGB color acquired at 30 frames/s, using a Canon EOS 5D with a fixed 28 mm lens. Automatic mode was used so the camera selected the aperture, ISO and white balance values best suited to the general shooting conditions. The camera was positioned approximately 160 cm from the ground and at 60° from the vertical axis. Example frames are shown in Fig. 13, where walking speed was 4–6 km/h, measured using GPS on a mobile phone. We reduced the processing time by segmenting only part of the far area in the key frame (row $1 -$ row $M_h/3$, col $M_w/5 -$ col $4M_w/5$), and performing principal component analysis to reduce feature dimensions to 12—accounted for 99.9% of the variance. The radial basis function (RBF) kernel was employed in the SVM classification. The parameters used in the RBF were selected by grid search using cross validation (initially the penalty parameter C was 7 and kernel parameter γ was 7.8).

A. Multiclass Classification

We tested our framework with the three classes of terrain described in Section III. Image size was reduced by a scale factor of 4 to speed up the segmentation process, but features were extracted at full resolution. For training purposes,

---

[2]Based on the average step length of 90 cm and walking speed of 5 km/h [48], one step takes approximately 20 frames when a 30 frames/s camera is used ($K_w = $ (framerate[frames/s] · step length[cm/step])/ (walking speed[cm/s]) $=$ $(30 \times 90)/((5 \times 10^5/(60 \times 60)))$ $=$ 19.44 frames/step).

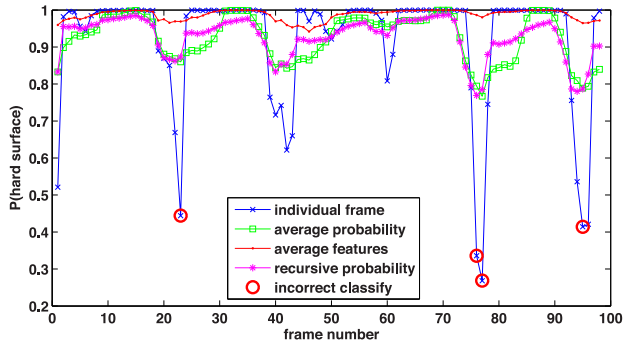Fig. 13.   Example frames from walking videos.

Fig. 14.   Comparison of classification performance when using average and recursive methods.
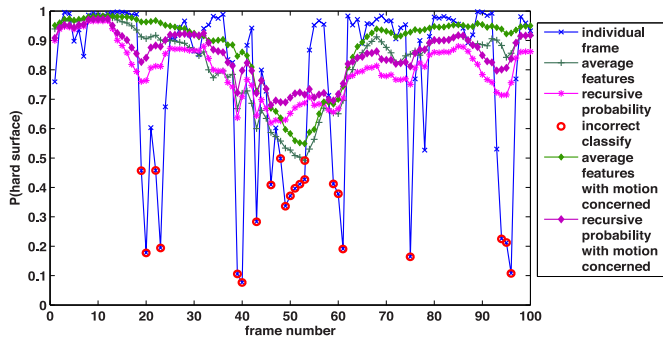

Fig. 15.   Comparison of classification performance when motion blur is employed.

only sharp frames from a range of videos including all material types were used. These were segmented into various region sizes to generate 1000 training samples for each class.[3] These videos are independent of the testing videos.

Fig. 14 shows the classification performance using: 1) individual frames; 2) a weighted average of features of each frame to compute probability; 3) a weighted average of individual probability of each frame; and 4) the proposed recursive method. All methods using temporal information improved classification accuracy (by 100% in these plots). The average-based features in general show the best probability (P_n close to 1). However, in the difficult scenes (Fig. 15), where incorrect classifications occurred because of high movement or consecutive motion blur, the recursive probability approach outperformed the others. That is, our method offers better robustness. Fig. 15 shows a further improvement when updating weights to compensate for motion blur. Examples of the subjective results are shown in Fig. 16. The right column of the figure shows the results of the high motion-blur

---

[3]Dataset is available at http://seis.bris.ac.uk/~eexna/download.html

Fig. 16.   Subjective results. Label 1 (green), label 2 (red), and label 3 (blue) correspond to the areas classified as hard surfaces, soft surfaces, and unwalkable areas, respectively. The size of the circle indicates probabilities—bigger implies higher confidence of classification. The left and right columns are from sharp and blurred frames, respectively. Classification accuracy (top to bottom) of left column: 87.07%, 89.96%, and 86.79%; right column: 79.75%, 67.32%, and 75.17%.

TABLE II
AVERAGE CLASSIFICATION ACCURACY (%) WHEN
WALKING EFFECTS ARE CONCERNED

Method	[6]	[7]	[8]	w/o walk compensation	w walk compensation
Accuracy	74.2	66.7	78.0	79.4	82.0

frames which can cause incorrect region warping and incorrect prediction. Our method is, however, robust to these influences.

We compared our method with the approaches presented in [6]–[8]. These methods partitioned each frame into near and far patches which exhibited different textural characteristics and which were processed independently. Table II shows the average classification accuracies for 15 test videos containing all types of terrains over a walk of duration 40 s. Ground-truth videos were manually labeled. The accuracy of our method was significantly better, improving classification from 66.7% using the method from [7] to a value of 82% with walking compensation. Our system increased computational time slightly compared to the methods in [6] and [8] by approximately 8%, but was lower than the method in [7] by 5%. The increase over the method in [6] and [8] is primarily due to the region warping process.

We also compared the performance of terrain classification, using the proposed walking compensation, with the video deblurring approaches proposed in [46] and [47]. These two techniques sharpen the blurred pixels using information in the spatial and frequency domains of neighboring frames, respectively. For fair comparisons, all systems update the segmented regions at the same key frames. Table III shows the performance comparison. The complexity was measured as the computational cost compared to that of when using the walking compensation. We tested three window sizes used in the

TABLE III
PERFORMANCE COMPARISON BETWEEN WALKING
COMPENSATION AND DEBLURRING METHODS

Method	# of frames for deblurring	Accuracy (%)	Complexity (times)
Walk compensation	-	82.0	1
spatial deblurring [46]	6	80.8	2.0
	12	81.4	4.5
	C_w	81.8	7.3
freq. deblurring [47]	6	80.5	5.6
	12	82.0	9.8
	C_w	82.1	14.1

Fig. 17. Camera panning scenes (moving from right to left). Left: slow pan. Right: quick pan showing missing area in the left part.
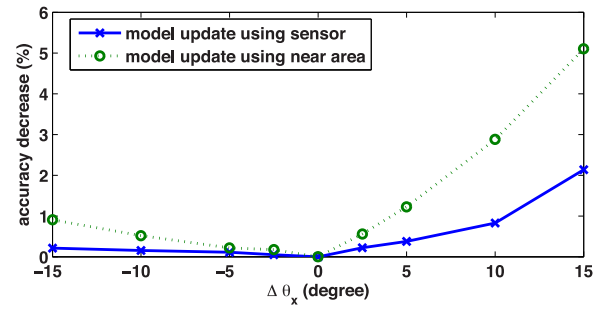

Fig. 18. Comparison of classification accuracy when the errors in θ_x present.

TABLE IV
AVERAGE CLASSIFICATION ACCURACY (%) WHEN
MODEL UPDATE IS EMPLOYED

Method	[6]	[7]	[8]	without model update	with model update	
					near area	sensor
Accuracy	70.4	62.5	71.7	74.2	78.7	80.1

deblurring process, namely 12 frames—similar to that used in [46]; six frames—for lower computational cost; and C_w frames—adaptive according to walking cycle of which the average was 18 frames. It is obvious that using more neighboring frames achieves better classification, but results in a higher computational time. The shorter the window, the more likely it does not contain a nonblurred frame, i.e., there are more consecutive blurred frames than the window length. Based on both classification accuracy and complexity, our walking compensation outperforms the spatial deblurring method in [46]. This is because a slight misalignment amongst frames used for deblurring may result in a large change in the texture in the sharpened frames. The fusion method in [47] does not produce a misalignment, since it includes nonrigid frame registration. However, it requires significantly higher computational cost, whilst offering an accuracy improvement of less than 0.1%.

B. Robust Framework

1) Horizontal Camera Motion: We investigated the performance of the proposed framework when the video content included horizontal movement caused by turning or camera panning. Examples of difficult cases are shown in Fig. 17, including the results from a slow pan containing high motion blur (left) and a fast pan with obstacles (right). The result for the slow pan achieved correct prediction in the areas with low motion blur, whilst suffering where high motion blur occurred; the wall on the right of the image exhibited an incorrect prediction. The system, however, soon restored its performance when the video returned to normal walking in a forward direction. For the fast pan case, the system achieved correct results for all areas on the right of the image, including the fence. However, there were missing areas (on the left of the image) where the system did not perform correct classification. This problem occurred because the fast-moving camera caused many

successive frames to be blurred. These frames were consequently skipped and the newly appearing regions were not updated. The system, however, recovered when the next key frame arrived (at least every 1 s), and the segmentation process updated these new areas.

2) Sloped Ground: When walking on sloped ground, the performance of the system is possibly lower because the actual θ_x is different from the predefined value. After the ground returns to being level, the system will soon recover back to a performance level similar to that of the case of horizontal camera motion. A long slope might, however, affect the performance more severely. Hence, we have developed a parallel process to estimate ground orientation [49], which will be included in the system in the future.

In this section, the effect of θ_x was tested. Fig. 18 shows the performance when errors in θ_x are present. Positive $\Delta\theta_x$ and negative $\Delta\theta_x$ imply ascending and descending slopes, respectively. Interestingly, walking uphill affects system performance more than walking downhill. This is because the weight distribution used in the recursive probability estimation is wider than it should be, which raises a problem of error integration.

3) Camera Type: We tested the robustness of our method by using test sequences from a different camera to that used for the training dataset. A GoPro Hero3 was used to capture similar scenes to those in the test videos in Section VI-A. The videos were acquired at 30 frames/s with 1920 × 1080 spatial resolution using a medium wide angle lens. These videos also exhibited lens distortions which affected the projective frequency analysis in Section IV-C1.

Results are shown in Table IV, where our method, with and without model update, is compared to the methods in [6]–[8]. We updated the model using the result from: 1) the near area and 2) the groundtruth, referred to here as the "sensor" (since a mechanical sensor could be used to confirm the existence of a hard or soft surface when the robot steps on it). Our method showed improved results, even when the update model is not employed. The model updating process further improves
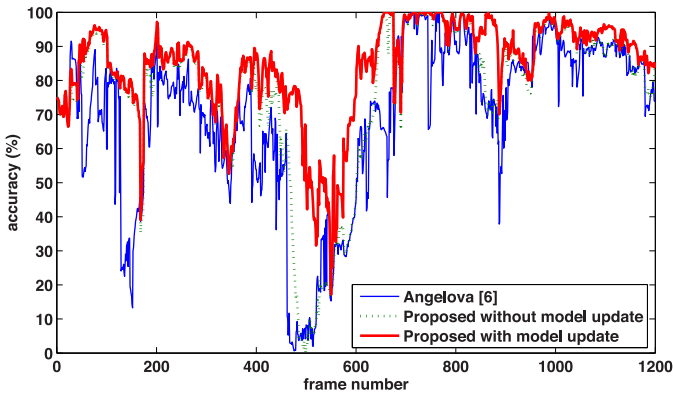
Fig. 19. Classification accuracy per frame of a GoPro video.

Fig. 20. Subjective results of GoPro videos (same labels as Fig. 16). Classification accuracy: (left) 78.99% and (right) 82.23%.

the classification accuracy. With near area information (semi-supervised system) the accuracy was improved by 4.5%, while using groundtruth improved the system performance by up to 6%. The classification accuracy of each frame is shown in Fig. 19, which reveals that our method can deal better with difficult scenes. Subjective results are shown in Fig. 20. The dips in the graph of Fig. 19 correspond either to motion blur or to changes in surface types. For example, when a new terrain type appears in the far distance, the classifier reverts to the model from the training camera which is based on different frequency characteristics. This leads to poorer results. However, as the system continues to receive texture information, the classification performance improves if walking consistently on the same surface type.

VII. CONCLUSION

We have presented a novel framework for terrain type classification based on video acquired when walking. This can be used by autonomous robots to make locomotion decisions when traversing difficult and varied terrain. It also has potential application to guidance aids for the visually impaired. The proposed scheme employs texture parameters along with information about walking behavior to compute the terrain class probability. Using our recursive filtering method with model updating, our framework outperforms existing methods by up to 16% in terms of classification performance. It also provides a robust solution, exhibiting resilience to horizontal camera motion and changes in camera type.

We believe that our method outperforms previous approaches because it exploits information in both temporal and spatial dimensions. It also, for the first time, takes account of blur information during the walking cycle. Finally, our classifier is updated intelligently as new information appears in the scene.

A possible area for future research is how to deal with classification uncertainty. In cases when the probability of the selected class is low (or the probabilities of several classes are similar) a means of further validating surface type is likely to be needed to ensure stability and safety of locomotion.

REFERENCES

[1] Y. Alon, A. Ferencz, and A. Shashua, "Off-road path following using region classification and geometric projection constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1. New York, NY, USA, 2006, pp. 689–696.

[2] R. Manduchi, A. Castano, A. Talukder, and L. Matthies, "Obstacle detection and terrain classification for autonomous off-road navigation," *Auton. Robots*, vol. 18, no. 1, pp. 81–102, 2005.

[3] 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 *Proc. IEEE/RSJ Conf. Intell. Robots Syst.*, St. Louis, MO, USA, 2009, pp. 2174–2181.

[4] K. Konolige *et al.*, "Outdoor mapping and navigation using stereo vision," in *Experimental Robotics* (Springer Tracts in Advanced Robotics), vol. 39, O. Khatib, V. Kumar, and D. Rus, Eds. Berlin, Germany: Springer, 2008, pp. 179–190.

[5] M. Bajracharya, A. Howard, L. H. Matthies, B. Tang, and M. Turmon, "Autonomous off-road navigation with end-to-end learning for the LAGR program," *J. Field Robot.*, vol. 26, no. 1, pp. 3–25, 2009.

[6] A. Angelova, L. Matthies, D. Helmick, and P. Perona, "Fast terrain classification using variable-length representation for autonomous navigation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, 2007, pp. 1–8.

[7] J. Chetan, M. Krishna, and C. V. Jawahar, "Fast and spatially-smooth terrain classification using monocular camera," in *Proc. IEEE Int. Conf. Pattern Recognit.*, Istanbul, Turkey, 2010, pp. 4060–4063.

[8] P. Filitchkin and K. Byl, "Feature-based terrain classification for little-dog," in *Proc. IEEE/RSJ Conf. Intell. Robots Syst.*, Vilamoura, Portugal, 2012, pp. 1387–1392.

[9] D. Maier, C. Stachniss, and M. Bennewitz, "Vision-based humanoid navigation using self-supervised obstacle detection," *Int. J. Hum. Robot.*, vol. 10, no. 2, p. 1350016, 2013.

[10] T. Buschmann *et al.*, "Walking in unknown environments—A step towards more autonomy," in *Proc. IEEE/RAS Int. Conf. Hum. Robots*, Nashville, TN, USA, 2010, pp. 237–244.

[11] A. Larson, R. Voyles, and G. Demir, "Terrain classification through weakly-structured vehicle/terrain interaction," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1. New Orleans, LA, USA, 2004, pp. 218–224.

[12] R. O'Callaghan and D. Bull, "Combined morphological-spectral unsupervised image segmentation," *IEEE Trans. Image Process.*, vol. 14, no. 1, pp. 49–62, Jan. 2005.

[13] M. Turtinen and M. Pietikainen, "Visual training and classification of textured scene images," in *Proc. Int. Workshop Texture Anal. Synth.*, 2003, pp. 101–106.

[14] A. Broggi, C. Caraffi, R. Fedriga, and P. Grisleri, "Obstacle detection with stereo vision for off-road vehicle navigation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, 2005, p. 65.

[15] R. Hadsell *et al.*, "Learning long-range vision for autonomous off-road driving," *J. Field Robot.*, vol. 26, no. 2, pp. 120–144, 2009.

[16] J. Choi *et al.*, "Environment-detection-and-mapping algorithm for autonomous driving in rural or off-road environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 974–982, Jun. 2012.

[17] C. Caraffi, S. Cattani, and P. Grisleri, "Off-road path and obstacle detection using decision networks and stereo vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 4, pp. 607–618, Dec. 2007.

[18] R. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, Nov. 1973.

[19] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *Int. J. Comput. Vis.*, vol. 62, nos. 1–2, pp. 61–81, Apr. 2005.

[20] Y. Dong, D. Tao, X. Li, J. Ma, and J. Pu, "Texture classification and retrieval using shearlets and linear regression," *IEEE Trans. Cybern.*, 2014.

[21] J. S. Weszka, C. R. Dyer, and A. Rosenfeld, "A comparative study of texture measures for terrain classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 4, pp. 269–285, Apr. 1976.

[22] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. 1. Reading, MA, USA: Addison-Wesley, 1991.

[23] A. Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 11, pp. 1186–1191, Nov. 1993.

[24] J. J. Lewis, R. O'Callaghan, S. G. Nikolov, D. R. Bull, and N. Canagarajah, "Pixel- and region-based image fusion with complex wavelets," *Inf. Fusion*, vol. 8, no. 2, pp. 119–130, 2007.

[25] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, Graz, Austria, 2006, pp. 404–417.

[26] Y. Khan, P. Komma, and A. Zell, "High resolution visual terrain classification for outdoor robots," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Barcelona, Spain, 2011, pp. 1014–1021.

[27] Y. Zou, W. Chen, L. Xie, and X. Wu, "Comparison of different approaches to visual terrain classification for outdoor mobile robots," *Pattern Recognit. Lett.*, vol. 38, pp. 54–62, Mar. 2014.

[28] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Eng. Appl. Artif. Intell.*, vol. 26, no. 4, pp. 1373–1385, 2013.

[29] Y. Sakagami *et al.*, "The intelligent ASIMO: System overview and integration," in *Proc. IEEE/RSJ Conf. Intell. Robots Syst.*, vol. 3. Lausanne, Switzerland, 2002, pp. 2478–2483.

[30] K. Sabe *et al.*, "Obstacle avoidance and path planning for humanoid robots using stereo vision," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 1. New Orleans, LA, USA, 2004, pp. 592–597.

[31] J.-S. Gutmann, M. Fukuchi, and M. Fujita, "3D perception and environment map generation for humanoid robot navigation," *Int. J. Robot. Res.*, vol. 27, no. 10, pp. 1117–1134, 2008.

[32] A. Hornung, K. Wurm, and M. Bennewitz, "Humanoid robot localization in complex indoor environments," in *Proc. IEEE/RSJ Conf. Intell. Robots Syst.*, Taipei, Taiwan, 2010, pp. 1690–1695.

[33] K. Nishiwaki, J. Chestnutt, and S. Kagami, "Autonomous navigation of a humanoid robot over unknown rough terrain using a laser range sensor," *Int. J. Robot. Res.*, vol. 31, no. 11, pp. 1251–1262, 2012.

[34] S.-H. Ryu *et al.*, "Humanoid path planning from HRI perspective: A scalable approach via waypoints with a time index," *IEEE Trans. Cybern.*, vol. 43, no. 1, pp. 217–229, Feb. 2013.

[35] P. Michel *et al.*, "GPU-accelerated real-time 3D tracking for humanoid locomotion and stair climbing," in *Proc. IEEE/RSJ Conf. Intell. Robots Syst.*, San Diego, CA, USA, 2007, pp. 463–469.

[36] B. Wu, T. L. Ooi, and Z. J. He, "Perceiving distance accurately by a directional process of integrating ground information," *Nature*, vol. 428, pp. 73–77, Apr. 2004.

[37] H. Farid and J. B. Woodward, "Video stabilization and enhancement," Dept. Computer Science, Dartmouth College, Hanover, NH, USA, Tech. Rep. TR2007-605, 2007.

[38] I. Selesnick, R. Baraniuk, and N. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Signal Process. Mag.*, vol. 22, no. 6, pp. 123–151, Nov. 2005.

[39] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, Apr. 2011.

[40] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[41] R. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, no. 5, pp. 786–804, May 1979.

[42] S. Grigorescu, N. Petkov, and P. Kruizinga, "Comparison of texture features based on Gabor filters," *IEEE Trans. Image Process.*, vol. 11, no. 10, pp. 1160–1167, Oct. 2002.

[43] W. Hwang, C. Lu, and P. Chung, "Shape from texture: Estimation of planar surface orientation through the ridge surfaces of continuous wavelet transform," *IEEE Trans. Image Process.*, vol. 7, no. 5, pp. 773–780, May 1998.

[44] R. Diao, F. Chao, T. Peng, N. Snooke, and Q. Shen, "Feature selection inspired classifier ensemble reduction," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1259–1268, Aug. 2014.

[45] N. Anantrasirichai, J. Burn, and D. R. Bull, "Projective image restoration using sparsity regularization," in *Proc. IEEE Int. Conf. Image Process.*, Melbourne, VIC, Australia, 2013, pp. 1080–1084.

[46] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1150–1163, Jul. 2006.

[47] N. Anantrasirichai, A. Achim, N. Kingsbury, and D. Bull, "Atmospheric turbulence mitigation using complex wavelet-based fusion," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2398–2408, Jun. 2013.

[48] P. A. Hageman and D. J. Blanke, "Comparison of gait of young women and elderly women," *Phys. Therapy J.*, vol. 66, pp. 1382–1387, Sep. 1986.

[49] N. Anantrasirichai, J. Burn, and D. R. Bull, "Orientation estimation for planar textured surfaces based on complex wavelets," in *Proc. IEEE Int. Conf. Image Process.*, 2014.

Nantheera Anantrasirichai (S'04–M'07) received the B.E. degree in electrical engineering from Chulalongkorn University, Bangkok, Thailand, the M.E. degree in telecommunications from the Asian Institute of Technology, Pathumthani, Thailand, and the Ph.D. degree in electrical and electronic engineering from the University of Bristol, Bristol, U.K., in 2000, 2003, and 2007, respectively.

She is currently a Research Associate with the Visual Information Laboratory, University of Bristol. Her current research interests include image and video coding, image analysis and enhancement, image fusion, medical imaging, and texture-based image analysis.

Jeremy Burn, photograph and biography not available at the time of publication.

David Bull (M'94–SM'07–F'13) received the B.Sc. degree from the University of Exeter, U.K., the M.Sc. degree from the University of Manchester, U.K., and the Ph.D. degree from the University of Wales, U.K., in 1980, 1983, and 1988, respectively.

He is currently the Chair in Signal Processing at the University of Bristol, U.K and prior to that was previously a Lecturer with the University of Wales, U.K., and a Systems Engineer with Rolls Royce. He was the Head of the Electrical and Electronic Engineering Department, University of Bristol, from 2001 to 2006, and is currently the Director of Bristol Vision Institute. In 2001, he co-founded ProVision Communication Technologies, Ltd., Bristol, and was its Director and Chairman until 2011. He has worked widely in the fields of 1-D and 2-D signal processing and has published some 450 academic papers and articles and has written two books and holds numerous patents, several of which have been exploited commercially. His research interests address the problems of image and video communication and analysis for wireless, Internet, surveillance, broadcast, and immersive applications.