

High Resolution Visual Terrain Classification for Outdoor Robots

Yasir Niaz Khan

Philippe Komma

Andreas Zell

Chair of Cognitive Systems, Computer Science Department, University of Tübingen
Sand 1, D-72076 Tübingen, Germany

<http://www.ra.cs.uni-tuebingen.de>

Abstract

In this paper we investigate SURF features for visual terrain classification for outdoor mobile robots. The image is divided into a grid and SURF features are calculated on the intersections of this grid. These features are then used to train a classifier that can differentiate between different terrain classes. Images of five different terrain types are taken using a single camera mounted on a mobile outdoor robot. We further introduce another descriptor, which is a modified form of the dense Daisy descriptor. Random forests are used for classification on each descriptor. Classification results of SURF and Daisy descriptors are compared with the results from traditional texture descriptors like LBP, LTP and LATP. It is shown that SURF features perform better than other descriptors at higher resolutions. Daisy features, although not better than SURF features, also perform better than the three texture descriptors at high resolution.

1. Introduction

The estimation of the ground surface is essential for a safe outdoor traversal of an autonomous robot. The robot must be aware of ground surface hazards induced by the presence of slippery and bumpy surfaces when employed for a variety of outdoor assignments, such as rescue missions or surveillance operations. These hazards are known as non-geometric hazards [28].

Terrain identification techniques can be classified into at least two different groups: retrospective and prospective terrain identification. Whereas retrospective techniques predict the traversed ground surface from data recorded during robot traversal [15], prospective techniques classify terrain sections, which will be traversed in the near future, i.e. located in front of the robot. The latter approaches can rely on the environment's geometry at short and long range acquired using either LADAR sensors [25] or stereo cameras [3]. However, classifying terrain based on geometrical reasoning alone gives rise to ambiguities which cannot be re-

solved in some situations: for example, tall grass and a short fence may provide similar geometrical features. Furthermore, stereo cameras yield very little information at long range. However, this information is important for generating paths which can safely be traversed by the robot while moving towards distant targets. Hence, in this paper, we consider another class of prospective terrain classification techniques which relies on texture features acquired from monocular cameras. Compared to the geometric features, these texture features provide more meaningful information about the ground surface even at long-range distances. Using the extracted visual cues we then apply a Random Forests [16] based approach to the problem of terrain classification: i.e. after training a model which establishes the assignment between visual inputs and corresponding terrain classes, this model is then employed to predict the ground surface of a respective visual clue. As in [14], texture features are extracted from image patches which are regularly sampled from an image grid. We perform terrain classification on a patch-wise basis rather than on a pixel-wise basis because the latter tends to produce noisy estimations which complicates the detection of homogeneous ground surface regions [9].

Several authors have addressed the problem of representing texture information in terms of co-occurrence matrices [12], Markov modeling [18, 27], Local Binary Patterns (LBP) [20], and texton-based approaches [26, 2] to name a few. Yet, it remains unclear which approach is suited best for an online application on a real outdoor robot in terms of prediction accuracy. Hence, the main motivation of our paper is a thorough comparison of different texture descriptors for representing different terrain types and to devise new methods by adapting descriptors from other problem domains. Here, the data originates from a real robot traversal whose camera images contain artifacts such as noise and motion blur. These data differ from the ones included in the Brodatz data set [8] or Calibrated Colour Image Database [21], where the images have been acquired under controlled conditions lacking dark shadows and overexposure, artifacts



Figure 1. Outdoor robot used for experiments

common in images taken outdoors. In addition, we investigate four new texture descriptors: Local Ternary Patterns descriptor (LTP) [23], Local Adaptive Ternary Patterns descriptor (LATP) [1], SURF descriptor [4] and Daisy descriptor [24], which, to our knowledge, have not been tested in this domain by anyone else. These descriptors have been tested for this purpose [13], but at a very low resolution.

The remainder of this paper is organized as follows: In Sect. 2, we provide details of our classification experiments. Sect. 3 briefly summarizes the different texture descriptors used. Results are presented and discussed in Sect. 4. Finally, Sect. 5 gives conclusions.

2. Experimental Setup

2.1. Testing Platform

The testing platform we used is our outdoor robot (Fig 1), which is a modified RC-model truck whose body was removed and replaced by a dual-core PC, a 32-bit microcontroller and different sensors attached to the vehicle. This includes a Point-Grey Firefly color camera with a 6mm lens to capture images at a resolution of 640x480 pixels. For our experiments, we ran the robot at a speed of about 1 m/s while capturing images from the camera, hence not all of the acquired images are sharp due to motion blur artifacts. The height of the camera is approximately 50 cm from the ground. The camera is tilted down so as to capture the terrain directly in front of the robot. Hence, the camera captures images starting from a distance of 30 cm with respect to the robot's front. The robot is equipped with tractor tires to be able to run on rough outdoor terrain. However, these tires produce an increased amount of vibration while traversing even a smooth surface.

2.2. Terrain Classes

To capture terrain data, we drove the robot outdoors on our campus and captured the terrain types visible to the

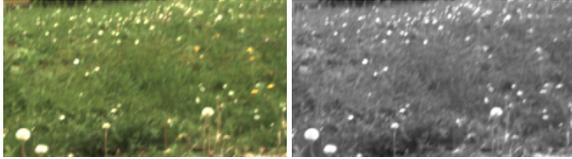
robot through the camera. The outdoor area of the campus consists of asphalt roads, meadows and some parking areas covered with gravel or tiles. We were able to identify five different classes: asphalt, gravel, grass, big tiles and small tiles. The robot was navigated multiple times over different routes at varying times of the day. One of these experiments was carried out under a heavily clouded sky after rainfall. Some of the terrain types contained wet and dry patches, e.g. asphalt, gravel, etc. In this case, a single terrain type contained different colors.

The second experiment took place at a time when the sun was about to set which resulted in over-exposure of the camera. In this case, the image colors were extremely distorted. The third scenario was at noon on a sunny day. Note that not all terrain types were captured in each scenario.

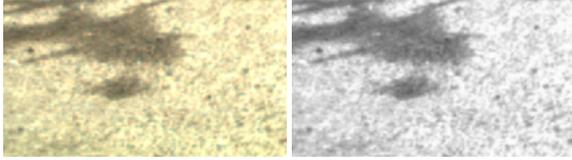
While driving on the campus we found that all terrain types contained many different features depending on the location and time at which the pictures were acquired. Fig. 2 shows different terrain types in color and grayscale to indicate the artifacts introduced under different scenarios. For example, Fig. 2(a) shows a blurred image of the grass terrain type along with small plants and their flowers. Fig. 2(b) shows the asphalt terrain type with a wet patch after rain. In fig. 2(c) the gravel terrain type is depicted after rainfall. Here, water was gathered in a bigger amount. Fig. 2(d) shows a sample image from the big-tiles terrain type. Since parts of the terrain are shadowed, its intensity changes a lot and the boundary also becomes difficult to classify. Similarly, Fig. 2(e) shows an image from the small-tiles terrain type. It is also noticeable that the shadow of a tree induces texture artifacts of its own.

Fig. 3 shows the grass terrain type under two different weather conditions. The image on the left is taken in winter (middle of March) one hour before sunset. The sun was looking into the camera at that time. The image on the right is taken in spring (middle of June) on a cloudy afternoon. Moreover, the image on the left also has a patch of snow among the grass. This clearly shows that color based descriptors will not work in all cases. Also note that under similar conditions, color based descriptors can misjudge the wet and dry or shaded and open parts of the same terrain type. Other than that, color will only accurately distinguish grass from other terrain types as is obvious from the sample terrain images. Finally, Fig 4(a) shows the small-tiles terrain type with blur induced due to robot motion and fig. 4(b) shows the same terrain type with over-exposure due to sun.

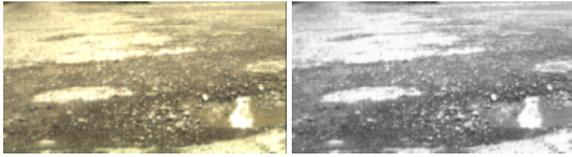
Most of the images are characterized by the presence of not only one but multiple terrain types. These images were labeled manually to generate training images for each class. The boundaries between any two terrain types are mostly diagonal or irregular. Hence, even after clipping, most of the images contained other terrain types at the borders.



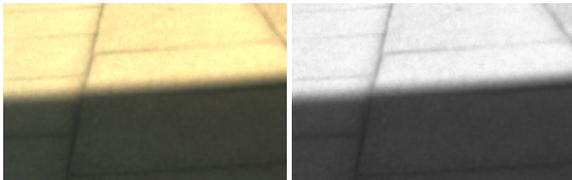
(a)



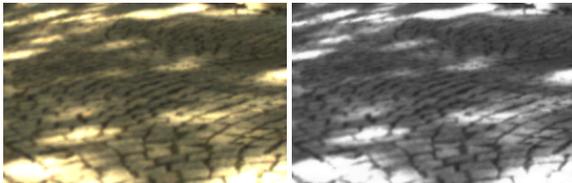
(b)



(c)



(d)



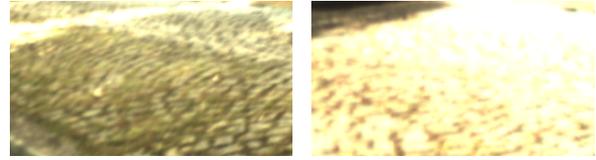
(e)

Figure 2. Sample images of different terrain types: (a) grass, (b) asphalt, (c) gravel, (d) big-tiles and (e) small-tiles, both in color and in grayscale



Figure 3. Difference of grass color under different sun angles

Note that this interferes with the terrain descriptors which are based on a square grid and hence results in a decrease in classification accuracy. Images containing blur were not filtered out, except in extreme cases where the blur artifacts were too dominant. For our experiments we tried different grid sizes for each descriptor to determine the best descrip-



(a)

(b)

Figure 4. Samples from small-tiles terrain type under: (a) blur, (b) over-exposure

tor for each resolution.

3. Texture Descriptors

3.1. SURF

Speeded Up Robust Features (SURF) [4] are an improvement of the famous SIFT features [17]. SURF is used to detect interest points in a grayscale image and to represent them using a 64- or 128-dimensional feature vector. These features can then be used to track the interest points across images and thus prove suitable for localization tasks. In this paper, we considered SURF features for a new application: texture classification. In SURF interest points are detected across the image using the determinant of the Hessian matrix. Box filters of varying sizes are applied to the image to extract the scale space. Then the local maxima are searched over space and scale to determine the interest points at the best scales. The key-point detection capabilities of SURF, however, have been omitted here. This is because the interest points detected by SURF are usually concentrated around sharp gradients, which are likely not present within homogeneous terrain patches. Instead we fix the interest point location and scale from which the SURF descriptor is determined.

In our approach we divide the image in a grid and use the generated patches or sub-windows to calculate the descriptors. Each image patch is then classified individually. We use 64-dimensional Upright-SURF (U-SURF) descriptors, in which the rotation invariance factor is removed. Still they are rotation invariant up to ± 15 degrees. Furthermore, we only consider a single scale for descriptor extraction which was determined experimentally using a grid-search approach. We call this modified approach TSURF (where T denotes Terrain). The SURF descriptor describes how the pixel intensities are distributed within a scale dependent neighborhood of each interest point. Haar wavelets are used to increase robustness and speed over SIFT features. First, a square window of size 20σ is constructed around the interest point, where σ is the scale of the descriptor. The descriptor window is then divided into 4×4 regular subregions. Within each subregion, Haar wavelets of size 2σ are calculated for 25 regularly distributed sample points. If the x and y wavelet responses are referred by dx and dy respectively,

then for the 25 sample points,

$$v_{subregion} = \left[\sum dx, \sum dy, \sum |dx|, \sum |dy| \right]$$

are collected. Hence, each subregion contributes four values to the descriptor vector, resulting in a final vector of length 64 (4 x 4 x 4).

3.2. Daisy

The Daisy descriptor [24] is inspired from earlier ones such as the Scale Invariant Feature Transformation (SIFT) [17] and the Gradient Location-Orientation Histogram (GLOH) descriptor [19], but can be computed much faster. It does not introduce artifacts that degrade the matching performance when used densely, unlike SURF, which can also be computed efficiently. For each image, first H orientation maps, $G_i, 1 \leq i \leq H$, are computed, one for each quantized direction, where $G_o(x,y)$ equals the image gradient norm at location (x,y) for direction o if it is bigger than zero, else it is equal to zero. Each orientation map is then convolved several times with Gaussian kernels of different \sum values to obtain convolved orientation maps for different sized regions. Daisy uses a Gaussian kernel, whereas SIFT and GLOH use a triangular shaped kernel.

Originally, Daisy features are calculated as dense features on the entire image. We instead divide the image into a grid of a specific size and calculate Daisy features on this grid, like our TSURF approach. We call this approach TDaisy (where T denotes Terrain). Then classification is performed on these local features. Each local feature is a 200-dimensional vector.

3.3. Local Binary Patterns

Local Binary Patterns (LBP) [20] are very simple, yet powerful texture descriptors. A 3x3 window is placed over each pixel of a grayscale image and the neighbors are thresholded based on the center pixel. Neighbors greater than the center pixel are assigned a value of 1, otherwise 0. Then the thresholded neighbors are concatenated to create a binary code which defines the texture at the considered pixel. Since the 8-bit binary pattern can have 256 values, we have a histogram containing 256 dimensions for classification. Below is an example of a 3x3 pixel pattern of an image. Thresholding is performed to obtain a binary pattern:

94	38	54
23	50	78
47	66	12

1	0	1
0		1
0	1	0

Binary Pattern = 10110100

3.4. Local Ternary Patterns

Local Ternary Patterns (LTP) [23] are a generalization of Local Binary Patterns. Here, instead of a binary pattern, a ternary pattern is generated by using a threshold k around the value c of the center pixel. Neighboring pixels greater than $c+k$ are assigned a value of 1, smaller than $c-k$ are assigned -1, and values between them are assigned 0.

$$T = \begin{cases} 1 & T \geq (c + k) \\ 0 & T < (c + k) \text{ and } T > (c - k) \\ -1 & T \leq (c - k) \end{cases}$$

where c is the intensity of the center pixel.

Instead of using a ternary code to represent the 3x3 matrix, the pattern is divided into two separate binary codes, part1 and part2. The first part contains the positive values from the ternary pattern, and the second contains the negative values. From both separately calculated 3x3 matrices an LBP is determined resulting in two individual matrices of LBP codes. Using these codes two separate histograms are calculated. The two histogram parts are concatenated to form a histogram of 512 dimensions.

Below is an example of a 3x3 pixel pattern of an image. A threshold parameter ($k=5$) is used to obtain a ternary pattern, which is then divided into two binary patterns:

94	38	54
23	50	78
47	66	12

1	-1	0
-1		1
0	1	-1

Ternary Pattern ($k=5$): 1(-1)01(-1)10(-1)

Part1=10010100, Part2=01001001

3.5. Local Adaptive Ternary Patterns

Local Adaptive Ternary Patterns (LATP) are based on the Local Ternary Patterns. Unlike LTP, they use simple local statistics to compute the pixel threshold. This makes them less sensitive to noise and illumination changes. LATP have been shown to work in face recognition problems [1]. We test this operator in the domain of texture classification. The basic procedure is the same as LTP. Instead of a constant threshold, the threshold (T) is calculated for each local window using local statistics as given in the equation:

$$T = \begin{cases} 1 & T \geq (\mu + k\sigma) \\ 0 & T < (\mu + k\sigma) \text{ and } T > (\mu - k\sigma) \\ -1 & T \leq (\mu - k\sigma) \end{cases}$$

where μ and σ are mean and standard deviation of the local region, respectively, and k is a constant. The resulting ternary pattern is divided into two binary patterns, part1 and part2, like LTP and separate histograms are calculated and concatenated for classification forming a 512 dimensional vector. Below is an example of LATP calculation:

94	38	54
23	50	78
47	66	12

1	0	0
-1		1
0	0	-1

$\mu=51.33, \sigma=25.74, \mu+k\sigma=77.07, \mu-k\sigma=25.59$

Ternary Pattern ($k=1$): 1001(-1)00(-1)

Part1=10010000, Part2=00001001

3.6. Classifier

We performed the classification task using several classifiers. Therefore, we used the machine learning software Weka [11] to train and test these classifiers. The classifiers tested were Random Forests, Support Vector Machine (SVM), Multilayer Perceptron (MLP), LIBLINEAR, J48 Decision Tree, Naive Bayes and k-Nearest Neighbor. From this set, Random Forests gave the best overall performance.

Decision Trees [22] have shown their applicability in various classification tasks [29, 5]. Yet, predictive models which have been generated with this approach tend to overfit the data and hence do not generalize well. Random forests [16] try to overcome these problems by injecting randomness into the tree generation procedure and combining the output of multiple randomized trees into a single classifier.

The trees are established by recursively bisecting the data set into smaller subsets at each inner node R_i . As splitting criterion the Gini-index [10] is employed which is defined by:

$$I_G(i) = \sum_{j=1}^k \hat{p}_{ij}(1 - \hat{p}_{ij}),$$

where k is the number of classes to discriminate and \hat{p}_{ij} denotes the probability of observing a measurement of class j with respect to all instances provided for node R_i . Formally, this probability is defined as $\hat{p}_{ij} = \frac{N_j}{N_i}$ with N_j denoting the number of measurements which belong to class j and N_i is the total number of observations in node R_i . At each splitting step, the remaining data is separated into two distinct subspaces or subnodes, R_{c_1} and R_{c_2} , using a random feature subset of size m , where m is typically chosen by $m = \sqrt{d}$. The best split is defined as the subdivision which maximizes the decrease in the Gini-index:

$$\Delta I_G(i) = I_G(i) - \hat{p}_{c_1} I_G(c_1) - \hat{p}_{c_2} I_G(c_2).$$

The splitting procedure is recursively adopted until a maximum tree depth is reached. For decision trees, pruning or recursion depth limitation techniques have to be applied to prevent overfitting. Random Forests classifiers, however, grow trees of maximum depth without performing subsequent pruning steps.

After tree generation, each leaf node stores several instances along with their respective class membership. The

latter can be adopted to assess the posterior distribution $p(c = k^* | x_i)$:

$$\begin{aligned} p(c = k^* | x_i) &= F(\{t_1, \dots, t_N\}, x_i, k^*) \\ &= \frac{1}{N} \cdot \sum_{j=1}^N \frac{N_r + f(t_j, x_i, k^*)}{K \cdot N_r + \sum_{l=1}^K f(t_j, x_i, k_l)}, \end{aligned}$$

where $f(t_j, x_i, k_l)$ denotes the number of estimation examples which belong to class k_l and which are assigned to the same leaf as instance x_i in t_j . Here, the term *estimation examples* is used to stress that this set is only adopted to the estimation of posterior probabilities and hence does not have to be identical to the training set in general. In this work, however, the approach of [7] has been followed which suggests to choose the estimation examples to be identical to the original set of training examples. Further, N_r represents a regularization term which behaves as a uniform Dirichlet prior [6] over feature values. If an instance assigned to a specific leaf node is not encountered during training, the inclusion of the additional terms assign a non-zero value to the corresponding probability.

During the recall phase, the test pattern traverses each random tree until a leaf node is reached. The posterior distributions assigned to the respective nodes are then averaged over all members of the ensemble. Finally, the class k^* which maximizes $p(c = k^* | x_i)$ is chosen to be the classification result of the test pattern.

Concerning prediction accuracy, using a larger number of trees reduces the generalization error for forests. However, this also increases the run-time complexity of the classification process. Hence, a compromise has to be found between accuracy and speed by varying the number of trees. We found that in our case 100 trees gave adequate accuracy without a significant loss in speed. Further, we adopted a 5-fold cross-validation scheme to verify the accuracy of the results.

4. Results

Classifiers were applied on each descriptor and the true positive ratio (TPR) of the entire dataset was obtained. The TPR is the ratio of the correctly classified instances to the number of all test patterns contained in the data set. Since Random Forests produced the best overall result, we only describe those results here. Table 1 presents a summary of accuracy results of the five approaches on the five terrain types.

Note that high resolution means that the image is divided into more parts, meaning that each grid cell is very small and so we get a lot of grid cells. For example, a 640x480 image divided into 10x10 patches gives 64x48=3072 grid cells. This is the highest resolution we tested, i.e. with a grid size of 10x10. On the other hand, low resolution means

Grid-size	LBP	LTP	LATP	TSURF	TDaisy
10	55.0%	67.2%	54.2%	99.2%	79.2%
20	75.7%	83.0%	73.7%	97.8%	74.9%
30	85.2%	90.0%	83.3%	96.6%	72.4%
40	90.2%	93.3%	88.4%	95.7%	70.8%
50	92.6%	94.7%	91.4%	94.9%	70.0%
60	94.4%	95.7%	93.4%	95.2%	69.0%
70	95.5%	96.8%	95.0%	94.5%	69.0%
80	95.8%	96.8%	95.8%	93.1%	67.8%
90	96.9%	97.4%	96.5%	93.6%	68.5%
100	96.9%	98.1%	97.2%	94.0%	69.1%

Table 1. Classification accuracy of the five descriptors

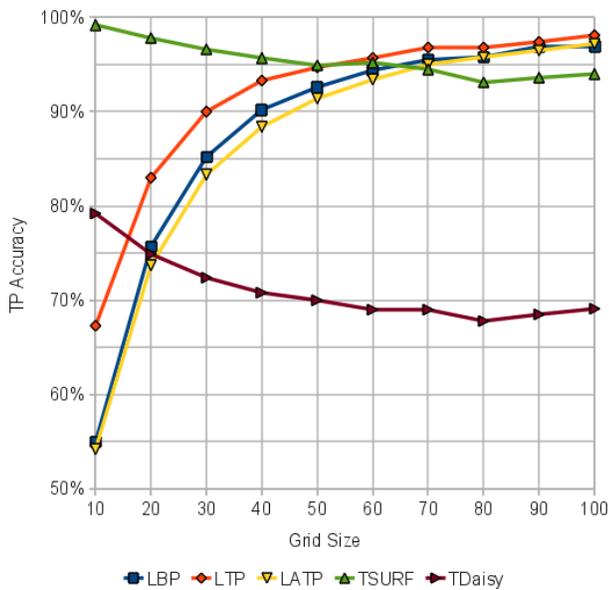


Figure 5. Graph of descriptor accuracies at different grid-sizes

that the image is divided into lesser cells and that the size of each grid cell is large. So in this case, a 640x480 image divided into 80x80 patches gives just 8x6=48 grid cells.

The same data is plotted in Fig. 5 for visualization. Here it is clear that, although at lower resolutions the texture descriptors such as Local Binary Patterns, Local Ternary Patterns and Local Adaptive Ternary Patterns perform the best, at higher resolutions, TSURF and TDaisy features produce much better results. At a grid-size of 50x50, TSURF matches the performance of the best texture descriptors, as both TSURF and LTP have an accuracy of about 95%. For higher resolutions, TSURF performs better than them. At a grid-size of 10x10, TSURF has a performance of 99%, whereas LTP only gives a performance of 67%.

It is to be noted that for grid-sizes lower than 30x30, the performance of the three texture descriptors decreases sharply. This is due to the fact that such a small cell doesn't

Grid-size	LBP	LTP	LATP	TSURF	TDaisy
10	64,963	88,356	93,139	34,714	96,359
20	13,417	16,279	16,654	5,196	17,669
30	3,479	5,448	6,054	1,576	6,946
40	1,599	2,784	2,451	625	2,921
50	808	1,420	1,285	313	1,614
60	532	776	863	178	833
70	388	573	499	119	546
80	190	349	256	81	304
90	169	282	213	43	239
100	121	232	148	31	169

Table 2. time taken in seconds for cross-validation by different descriptors at different grid-sizes

include enough neighboring information for adequate feature description. Even below a grid-size of 40x40, the performance of the three texture descriptors falls below 90% and hence they may not be usable.

The performance of the TDaisy descriptor improves with increasing resolution. Although worse than TSURF, it performs better than the three texture descriptors only at a grid-size of 10x10.

The TSURF descriptor is the smallest descriptor consisting of only 64 dimensions. The LTP and LATP descriptors are the longest descriptors consisting of a 512 dimensional vector each. LBP and TDaisy are intermediate length descriptors. LBP produces a 256-dimensional descriptor and the TDaisy descriptor consists of 200 dimensions. For TSURF based classification, different scale levels (σ) described in section 3.1 ranging from 2 to 20 were tried. Higher values of this scale parameter for descriptor calculation give the best result in all of the cases. For LTP-based classification, we also tried different values for the threshold value k described in section 3.4 having values between 2 and 20. It was observed that small values of the threshold give better results. Similarly for LATP-based classification, we tried different values of the threshold value k described in the section 3.5 ranging from 0.1 to 1.9. In this case, an intermediate threshold value close to 1.0 gives the best result.

The time taken for 5-fold cross-validation for all of the descriptors is described in Table 2. These times are in seconds and are for the random forests classifier used for validation.

Fig. 6 plots the time taken for 5-fold cross-validation in a graph. Note that the values are plotted on a logarithmic scale. Here we can observe that for all grid-sizes, TSURF takes the least amount of time. The most amount of time is mostly taken by LTP or similar descriptors. This is natural, since TSURF has the smallest descriptor vector.

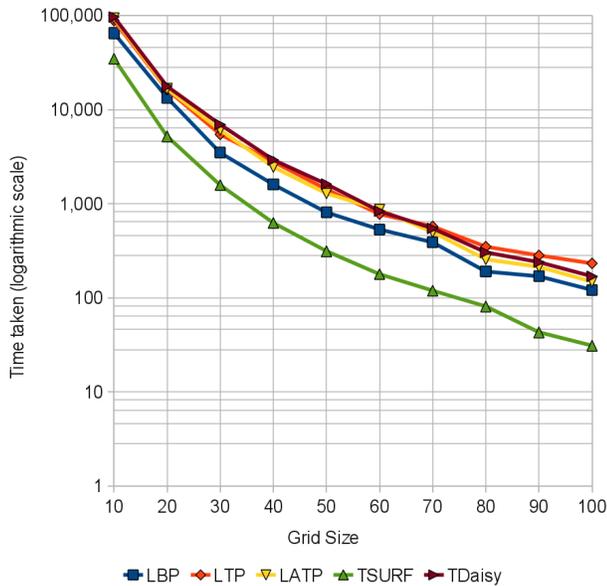


Figure 6. Graph of time taken for cross-validation depicted on logarithmic scale

5. Conclusion

In this paper, we thoroughly investigated the applicability of different local descriptors at varying resolutions for visual terrain classification on outdoor mobile robots. Most current texture classification approaches use sharp images containing a single texture captured from a perpendicular camera angle. Whereas, we used images from real runs of the robot containing blurred images with non-sharp terrain boundaries. Along with three texture-based descriptors, LBP, LTP, and LATP, we have tested modified forms of two other descriptors: SURF and DAISY. SURF and DAISY are modified to be calculated on a grid placed on the image. The texture-based descriptors performed best at low resolutions. LTP gave the best low resolution performance, however, it has one of the largest feature vectors. LATP is the other largest feature vector that also performs well. However, at higher resolutions TSURF performs much better than all other descriptors. In addition TSURF has one of the smallest feature vectors and is fast to train. TDaisy has the second smallest feature vector, but its performance is not satisfactory at almost all of the resolutions we tried. Hence, we have demonstrated that visual terrain classification can be performed from high to low resolution using one of the local descriptors we have described above. At resolutions higher than 50x50, TSURF should be used. For resolutions lower than 50x50 LTP should be used where accuracy is the most important; TSURF should be used where speed is the most important; and LBP should be used where a compromise between speed and accuracy is desired.

Furthermore, it is demonstrated that visual terrain clas-

sification can be successfully performed even under non-optimal conditions, such as motion blur induced by a fast moving robot and its vibrating camera, different weather conditions, both wet and dry ground surfaces and a low camera viewpoint.

References

- [1] M. Akhloufi and A. H. Bendada. Locally adaptive texture features for multispectral face recognition. In *IEEE International Conference on Systems, Man, and Cybernetics*, Istanbul, Turkey, October 2010. IEEE. 2, 4
- [2] A. Angelova, L. Matthies, D. Helmick, and P. Perona. Learning and prediction of slip from visual information: Research articles. *Journal of Field Robotics*, 24(3):205–231, 2007. 1
- [3] M. Bajracharya, T. Benyang, A. Howard, M. Turmon, and L. Matthies. Learning long-range terrain classification for autonomous navigation. In *IEEE International Conference on Robotics and Automation, 2008 (ICRA 2008)*, pages 4018–4024, Pasadena, CA, 2008. 1
- [4] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proceedings of the European Conference on Computer Vision (ECCV 2006)*, pages 404–417, Graz, Austria, 2006. 2, 3
- [5] A. Birk, T. Stoyanov, Y. Nevatia, R. Ambrus, J. Poppinga, and K. Pathak. Terrain classification for autonomous robot mobility: from safety, security rescue robotics to planetary exploration. In *IEEE International Conference on Robotics and Automation (ICRA), Planetary Rovers Workshop*, pages 1–5, 2008. 5
- [6] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 5
- [7] L. Breiman. Random forests. In *Machine Learning*, volume 45, pages 5–32, Hingham, MA, USA, October 2001. Kluwer Academic Publishers. 5
- [8] P. Brodatz. *Textures: A photographic album for artists & designers*. New York: Dover, New York, NY, 1966. 1
- [9] I. Davis, A. Kelly, A. Stentz, and L. Matthies. Terrain typing for real robots. In *Proceedings of the Intelligent Vehicles '95 Symposium*, pages 400–405, Detroit, MI, 1995. 1
- [10] C. Gini. Variabilita e mutabilita. *Journal of the Royal Statistical Society*, 76(3):326–327, February 1913. 5
- [11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, Volume 11, Issue 1, 2009. 5
- [12] R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, November 1973. 1
- [13] Y. N. 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 (CIVTS 2011)*, Paris, France, 2011. 2
- [14] D. Kim, S. Sun, S. Oh, J. Rehg, and A. Bobick. Traversability classification using unsupervised on-line visual learning

- for outdoor robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2006)*, pages 518–525, Orlando, FL, USA, 2006. 1
- [15] P. Komma, C. Weiss, and A. Zell. Adaptive Bayesian filtering for vibration-based terrain classification. In *IEEE International Conference on Robotics and Automation (ICRA 2009), Kobe, Japan*, pages 3307–3313, May 2009. 1
- [16] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006. 1, 5
- [17] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 3, 4
- [18] B. Manjunath and R. Chellappa. Unsupervised texture segmentation using markov random field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):478–482, 1991. 1
- [19] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(10):1615 – 1630, October 2005. 4
- [20] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 24(7):971–987, 2002. 1, 4
- [21] A. Olmos and F. Kingdom. McGill calibrated colour image database. 1
- [22] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, San Mateo, CA, 1993. 5
- [23] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *Proceedings of the 3rd international conference on Analysis and modeling of faces and gestures (AMFG 07)*, pages 168–182, Berlin, Heidelberg, 2007. Springer-Verlag. 2, 4
- [24] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:815–830, 2010. 2, 4
- [25] N. Vandapel, D. Huber, A. Kapuria, and M. Hebert. Natural terrain classification using 3-d lidar data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2004)*, pages 5117–5122, New Orleans, LA, April 2004. 1
- [26] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1-2):61–81, 2005. 1
- [27] P. Vernaza, B. Taskar, and D. Lee. Online, self-supervised terrain classification via discriminatively trained submodular Markov random fields. In *IEEE International Conference on Robotics and Automation (ICRA 2008)*, pages 2750–2757, 2008. 1
- [28] B. H. Wilcox. Non-geometric hazard detection for a mars microrover. In *Proceedings of the NASA Conference on Intelligent Robots in Field, Factory, Service and Space*, pages 675–684, Houston, TX, 1994. 1
- [29] D. Wilking and T. Röfer. Realtime object recognition using decision tree learning. In *RoboCup 2004: RobotWorld Cup VII*, pages 556–563, Heidelberg, 2005. Springer. 5