Visual Terrain Classification by Flying Robots

Yasir Niaz Khan, Andreas Masselli and Andreas Zell

Abstract— **In this paper we investigate the effectiveness of SURF features for visual terrain classification for outdoor flying robots. A quadrocopter fitted with a single camera is flown over different terrains to take images of the ground below. Each image is divided into a grid and SURF features are calculated at grid intersections. A classifier is then used to learn to differentiate between different terrain types. Classification results of the SURF descriptor are compared with results from other texture descriptors like Local Binary Patterns and Local Ternary Patterns. Six different terrain types are considered in this approach. Random forests are used for classification on each descriptor. It is shown that SURF features perform better than other descriptors at higher resolutions.**

I. Introduction

A. Motivation

For outdoor robots, it is very useful to know about ground surfaces for many purposes. Employed for a variety of outdoor assignments, such as rescue missions or surveillance operations, a flying robot should be able to recognize ground surfaces for successful completion of several outdoor tasks. Terrain classification can be performed by a driving robot [14], but it may not easily reach all areas and some areas may be hazardous for a driving robot. A flying robot is more appropriate for this purpose, since it can fly over almost any area and is not harmed by ground hazards. The classification data gives important information about possible places to land, it can be used to guide a driving robot along save paths, or it can be stored in a map for later use of robots or humans. A robot can rely on the environment's geometry at short and long range acquired using either LADAR sensors [22] or stereo cameras [1] for surface analysis. However, terrain classification based solely on geometrical reasoning gives rise to ambiguities which cannot be resolved in some situations: For example, tall grass and a small bush may have similar geometrical features. Stereo cameras only yield little information at long range. This information, however, is important for generating maps of the environment or to help navigation on ground.

Hence, in this paper, we consider another type of terrain classification which relies on texture descriptors acquired from monocular cameras. Compared to geometrical features, these texture features provide meaningful information about the ground surface even at long-range distances. We extract visual features from image patches as in [7], [13] which are periodically sampled from a grid drawn across the acquired image. We apply a Random Forests based approach on the extracted visual descriptors to solve the problem of terrain classification, i.e., after learning a model which establishes the relation between a visual feature and its corresponding terrain class, this model is then used to predict the ground surface of a respective visual clue. We perform terrain classification on a patch-wise basis rather than on a pixel-wise basis because the latter produces some noisy estimations which complicate the recognition of homogeneous ground surfaces [6].

The main motivation of our paper is to find out an excellent texture descriptor for representing different terrain types on a flying robot. Some approaches are present in literature, yet it remains unclear which approach is suited best for an application on a real outdoor robot in terms of prediction accuracy. Our data originates from a real robot flight whose camera images contain artifacts such as noise and motion blur. In addition to Local Binary Patterns [17], we investigate two further texture descriptors, the Local Ternary Patterns descriptor (LTP) [21] and the SURF descriptor [2], which, to our knowledge, have not been applied to the domain of terrain identification for flying robots before.

B. Related Work

Hudjakov et al. [11] used an artificial neural network to classify terrain from static aerial images, not from a moving UAV. Patterns of 29×29 pixel size were taken from these images and fed into a big neural network containing three hidden layers, to be classified either as houses, roads, grass or debris. This approach does not extract any features, rather processes raw patches of the image. Sofman et al. [20] used camera and laser range data to classify terrain from a UAV into road, grass, tree, and building areas. Laser scanners are generally much more expensive than cameras and have heavy resource consumption especially when they return reflectance data, thus requiring heavy duty UAVs.

Many authors have addressed the problem of representing texture information in terms of texton-based approaches [23], co-occurrence matrices [10], Local Binary Patterns (LBP) [17] and Markov modeling [16], [24] to name a few. Our dataset differ from the ones included in the Mcgill Calibrated Colour Image Database or Brodatz dataset [5]. There, the images have been captured under controlled conditions lacking dark shadows and overexposure. Note that these sources of noise are often present in images taken outdoors. A mobile phone mounted on a quadrocopter is used in [8] to perform visual localization using GPS data as ground truth. [12] used different visual descriptors to classify six different terrain

Y. N. Khan, A. Masselli and A. Zell are with the Chair of Cognitive Systems, Computer Science Department, University of Tübingen, Sand 1, D-72076 Tübingen, Germany {yasir.khan, andreas.masselli, andreas.zell}@uni-tuebingen.de

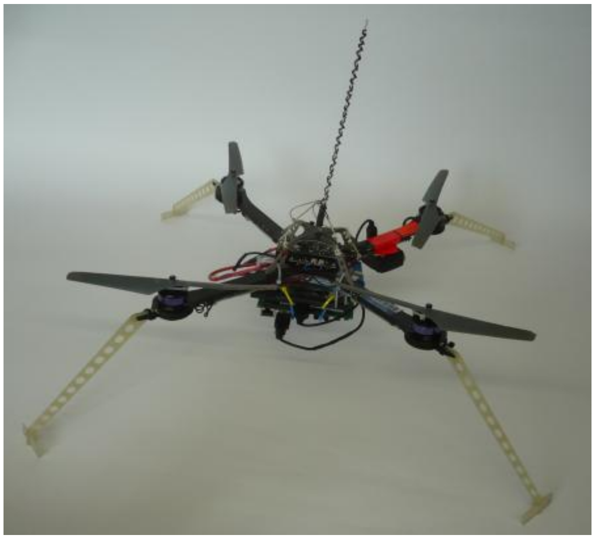


Fig. 1. Flying robot used for experiments

types, but on a driving robot. They obtain good results, but no similarly good approach has been tested for flying robots.

The remainder of this paper is organized as follows: In Sect. II, we provide details of our experiments. Sect. III summarizes the adopted techniques for representing acquired terrain patches in terms of meaningful texture descriptors. These texture descriptors constitute the basis on which the terrain classifier works. Results of our experiments are presented and discussed in Sect. V. Finally, Sect. VI gives conclusions.

II. Experimental Setup

A. The UAV Platform

The images were acquired using an *AscTec X3D-BL Hummingbird* quadrocopter which has a diameter of 53 cm and weighs 0.5 kg. It has been equipped with a *PointGrey FireFly USB* color camera with VGA resolution and a *Gumstix Overo Fire* single-board computer with a 600 MHz *ARM* processor. Pictures were taken during manually controlled flight with a framerate of 1 Hz and stored on a MicroSD card for later offline processing.

By varying the speed of the four motors, the aircraft can tilt, roll, yaw and change its altitude. The quadrotor and its peripheral devices are powered by a 2000 mAh lithium-polymer battery and allow a flight time of up to 15 minutes, depending on the flight maneuvers. The *X3D-BL Hummingbird* platform comes with a circuit board, including two 60 Mhz 32 bit ARM microcontrollers, a three-axis gyroscope, an accelerometer, a compass module, a GPS sensor and pressure sensor.

B. Terrain Types

We flew the robot outdoors in our detached campus in the “Sand“ area at Tübingen and observed the terrain types below the robot through the camera. The outdoor area of the campus consists of roads, meadows, bushes and some parking areas covered with gravel or tiles as shown in Fig 2.

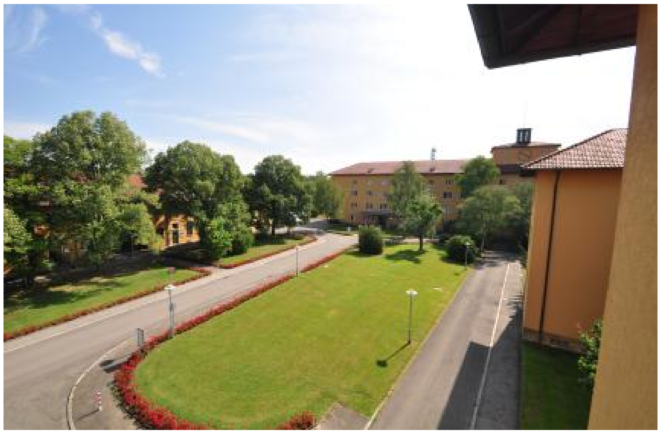
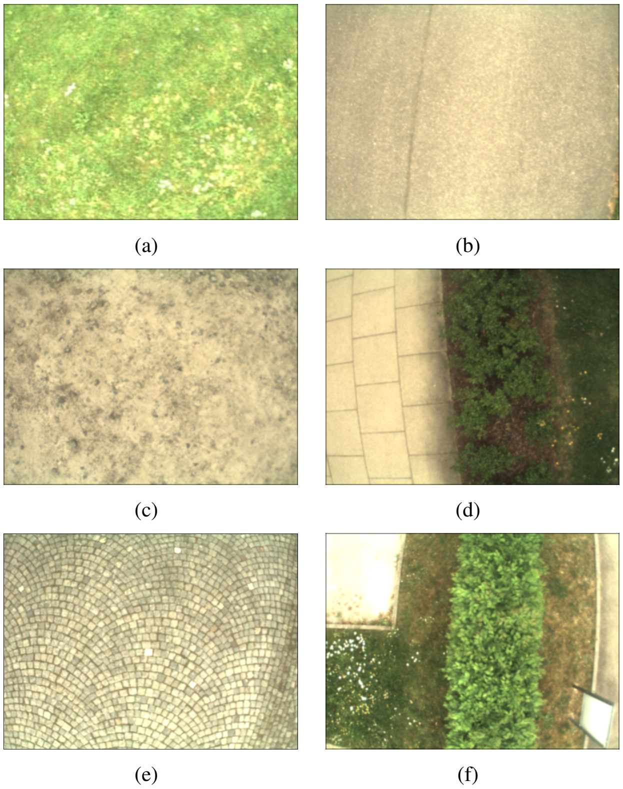


Fig. 2. Campus area for experiments



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 3. Images of various terrain types: (a) grass, (b) asphalt, (c) gravel, (d) big-tiles (left), (e) small-tiles and (f) bushes

We were able to identify six different classes: asphalt, gravel, grass, bushes, big-tiles and small-tiles. The robot was flown at varying heights between 1m to 5m. Images were captured at a resolution of 640 × 480.

While flying in the campus we found that all terrain types contained many different features depending on the location and time at which the pictures were acquired. Figure 3 shows different terrain types to indicate the artifacts introduced under different scenarios. For example, Fig. 3(a) shows an image of the grass terrain type along with small plants and their flowers. Figure 3(b) shows the asphalt terrain type with a line formed from a recent construction. In Fig. 3(c) the

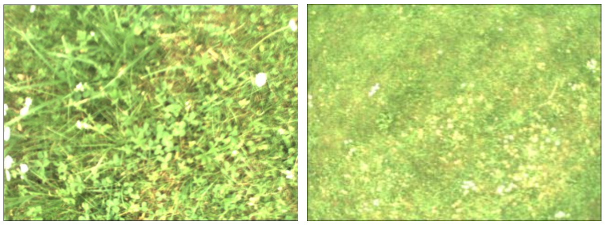


Fig. 4. Difference of scale for grass

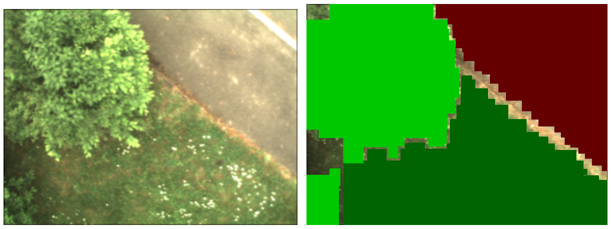


Fig. 5. An image and the corresponding ground truth containing three terrain types: bushes, grass, asphalt

gravel terrain type is depicted which contains a lot of soil as well. Figure 3(d) shows a sample image from the big-tiles terrain type on the left side along with some other terrains on the right. Similarly, Fig. 3(e) shows an image from the small-tiles terrain type. Finally Fig. 3(f) shows one type of bush that we observed.

Figure 4 shows the grass terrain type at two different heights of the quadrocopter. The image on the left is taken at a height of about 1m, whereas the image on the right is taken at several meters height. Both of these scenarios were included in the dataset for all terrain types. Also note that under similar conditions, color based descriptors can misjudge the shaded and open parts of the same terrain type. Color will only accurately distinguish grass and bushes from other terrain types as is obvious from the sample terrain images, but not within similarly colored terrain types.

All 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. Almost all of the images contained diagonal or irregular boundaries between two terrain types. Hence many images contained multiple terrain types. Note, that this interferes with the terrain descriptors which are based on a rectangular 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.

Ground truth was generated by labelling the images based on the terrain type. We chose different labelling colors for each terrain type. When visual descriptors for a patch of the image were generated, the corresponding patch in the ground truth image was observed to find out the current terrain type. If the patch contained more that 40 % of a specific terrain type, the patch was labelled with that terrain type for training and testing. So the images were not clipped and many images had multiple terrain types. Figure 5 shows an image with multiple terrains and the corresponding ground truth image.

In our approach we divide each image in a grid and use the generated patches or sub-windows to calculate the descriptors. Each image patch is then classified individually. For our experiments we tried different grid sizes for each descriptor to determine the best descriptor for each resolution. The patch size was varied between 10×10 to 100×100. Note that high resolution means that the image is divided into more patches, meaning that each grid patch is very small and so we get a lot of patches. For example, a 640×480 image divided into a 20×20 grid gives $32 \times 24 = 768$ patches. On the other hand, low resolution means that the image is divided into lesser number of patches and that the size of each patch is large. So in this case, a 640×480 image divided into an 80×80 grid gives just $8 \times 6 = 48$ patches.

At a resolution of 10×10, each terrain type had about 30 thousand sample points. At 100×100 resolution, we had 290 sample points for most of the descriptors.

III. Texture Descriptors

A. *SURF*

Speeded Up Robust Features (SURF) [2] are a modification of the famous SIFT features .In this paper, we used SURF features for a new application: texture classification. SURF is used to detect interest points in a grayscale image and represent them using a 64- or 128-dimensional feature vector. In SURF, first interest points are detected across the image using the determinant of the Hessian matrix. We, however, do not use the key-point extraction capabilities of SURF in our application. This is because the interest points detected by SURF are usually concentrated around areas containing sharp gradients, which are likely not present within most homogeneous terrain patches. Instead we fix the interest point locations and scales from which the SURF descriptors are determined. This renders our approach much faster. The interest points are calculated on the intersections of the grid lines that we draw across the image.

The computed SURF descriptor describes how the pixel intensities are distributed within a scale dependent neighborhood of each fixed interest point. Haar wavelets are used to increase robustness and speed.First, a square window of size 20σ is constructed around each interest point, where σ is the scale of the descriptor. The descriptor window is divided into 4×4 regular subregions. Within each resulting subregion, Haar wavelets of size 2σ are calculated for 25 regularly distributed sample points. If 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. Thus, each subregion contributes four values to the descriptor vector resulting in a final vector of length 64 ($4 \times 4 \times 4$).

In our approach, 64-dimensional Upright-SURF (U-SURF) descriptors have been used, where the rotation invariance factor is removed. We don't need rotation invariance, since the robot flies over the terrain in multiple directions and thus captures the terrain at multiple angles. U-SURF is still rotation invariant up to +/- 15 degrees. Furthermore, we consider a single scale for descriptor extraction in the whole image, which was determined experimentally using a grid-search approach. We call this modified approach TSURF or Terrain-SURF.

B. Local Binary Patterns

Local Binary Patterns (LBP) [17] are very simple, yet powerful texture descriptors. A 3×3 window is placed over each pixel of a grayscale image and the neighbors are thresholded based on the center pixel. Then the thresholded neighbors are concatenated in any direction to create a binary code which defines the texture at the center pixel.

We divide the image into a grid of patches and calculate a histogram of binary patterns of all pixels within a patch. Thus each patch yields a histogram which is then used to assign a terrain class to the respective patch. Since the 8-bit binary pattern can have 256 possible values, we get a histogram containing 256 dimensions for each patch.

Below is an example of a 3×3 pixel pattern of an image.

44	28	52
31	**48**	91
27	75	19

0	0	1
0		1
0	1	0

Binary Pattern $=$ 00110100

C. Local Ternary Patterns

Local Ternary Patterns (LTP) [21] are a generalization of Local Binary Patterns. A ternary pattern is calculated by using a threshold k along with the value c of the center pixel to threshold the neighboring pixels. The threshold is applied as follows:

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

where c is the intensity of the center pixel.

Instead of using a ternary code to represent the 3×3 matrix, the pattern is split into two separate matrices. The first matrix contains the positive values from the ternary pattern and the second contains the negative values. From each matrix an LBP is determined resulting in two individual matrices of LBP codes. Using these codes two separate histograms are computed.

Here, we also divide the image into a grid of patches and calculate histograms for each patch. The two histogram parts are concatenated to form a histogram of 512 dimensions.

IV. Classifiers

We performed the classification task using several classifiers. We used the machine learning software Weka [9] to train and test these classifiers. The tested classifiers were Random Forests, Support Vector Machine (SVM) using the Sequential Minimal Optimization (SMO) training algorithm, the Multilayer Perceptron (MLP), LIBLINEAR, J48 Decision Tree, Naïve Bayes and k-Nearest Neighbour. From this set, Random Forest gave the best overall performance. So we only provide results of the Random Forest classifier due to lack of space.

A. Random Forest

Decision Trees [19] have shown their applicability in various classification tasks [3]. Random forests [15], [18] try to reduce their problem of over-fitting 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 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. 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.The splitting procedure is recursively adopted until a maximum tree depth is reached. Random Forests classifiers grow trees of maximum depth without performing subsequent pruning stepsAfter tree generation, each leaf node stores several instances along with their respective class membership. The latter can be adopted to assess the posterior distribution $p(c = k^*|x_i)$:

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

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. In this work the approaches of [4] have been followed which suggest to choose the estimation examples to be identical to the original set of training examples. 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.

Using a larger number of trees reduces the generalization error for random 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 50 trees gave adequate accuracy without a significant loss in speed. We adopted a 10-fold cross-validation scheme to verify the accuracy of the results.

Grid-size	LBP	LTP	TSURF
10	33.6%	38.9%	**99.6%**
20	48.4%	55.7%	**96.8%**
30	59.4%	67.3%	**88.7%**
40	66.7%	73.6%	**81.5%**
50	71.0%	**78.1%**	76.6%
60	74.8%	**80.3%**	71.9%
70	77.3%	**81.7%**	71.2%
80	78.2%	**83.3%**	66.0%
90	79.6%	**84.7%**	67.7%
100	79.5%	**84.4%**	63.3%

TABLE I
CLASSIFICATION ACCURACY OF THE THREE DESCRIPTORS AT DIFFERENT GRID-SIZES

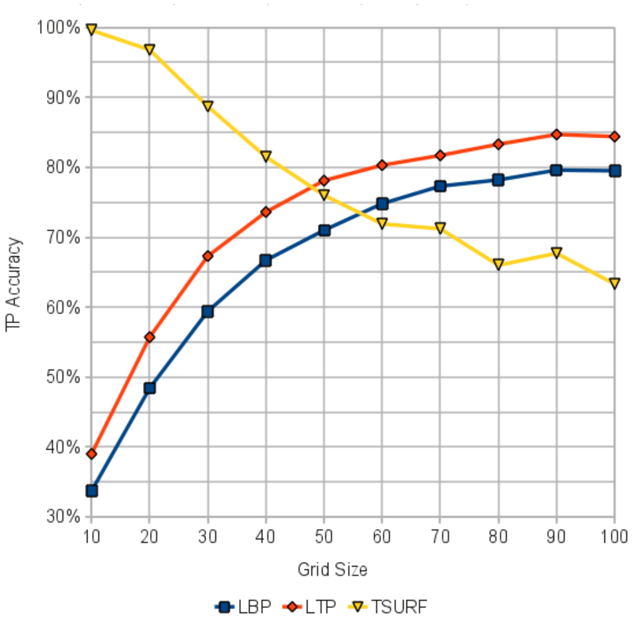


Fig. 6. Graph of descriptor accuracies at different grid-sizes

V. RESULTS

We tested different classifiers on each descriptor and obtained the true positive rate (TPR) of the entire dataset. The TPR is the ratio of the correctly classified instances to the total number of test patterns. Since Random Forests performed the best, only those results are described here. Table I presents accuracy results of the three approaches on the six terrain types with 10-fold cross-validation.

Fig. 6 shows a plot of accuracies for visualization. Here it is clear that, although at lower resolutions (less patches) the texture classifiers such as Local Binary Patterns and Local Ternary Patterns perform the best, at higher resolutions, TSURF features produce much better results. At a grid-size of 50×50, TSURF lags the performance of the best texture descriptor LTP by only 1.5 %. For higher resolutions, TSURF performs better than LTP. At a grid-size of 10×10, TSURF has a performance of 99.6 %, whereas LTP only gives a performance of 38.9 %.

It is to be noted that for grid-sizes lower than 40×40, the performance of the LBP and LTP decreases sharply. This is due to the fact that such a small patch doesn't include enough neighborhood information for adequate feature description.

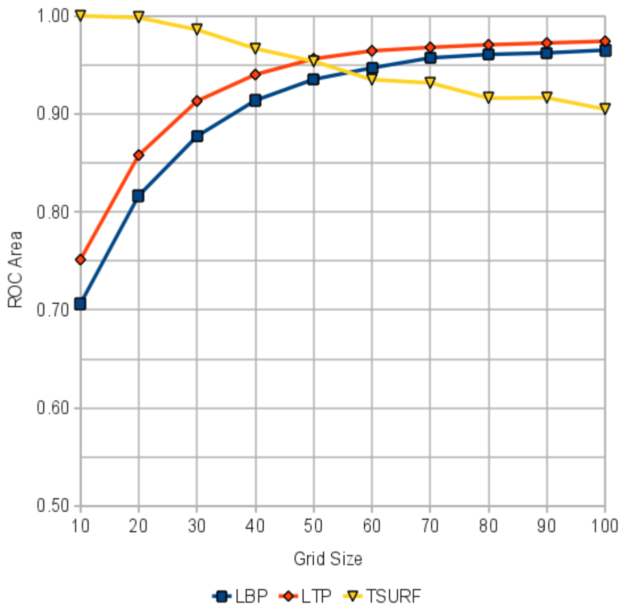


Fig. 7. Graph of ROC area under the curve at different grid-sizes

	gravel	grass	big-tiles	bush	small-tiles	asphalt
gravel	6,497	62	29	103	87	44
grass	29	6,255	223	103	6	206
big-tiles	5	125	6,606	14	0	72
bush	16	13	7	6,750	35	1
small-tiles	20	7	1	11	6,781	2
asphalt	2	70	38	13	0	6,698

TABLE II
CONFUSION MATRIX FOR TSURF AT GRID SIZE 20×20

The ROC area is plotted in Fig. 7. This graph shows the ROC area under the curve plotted for all grid sizes for the three descriptors.

In terms of descriptor size, the TSURF descriptor is the smallest descriptor consisting of only 64 dimensions. The LTP descriptor has the longest descriptor consisting of a 512 dimensional vector. LBP is an intermediate length descriptor with 256 dimensions. This has a big impact on training times for the classifier.

For TSURF based classification, different scale levels (σ) described in section III-A ranging from 2 to 20 were tried. Higher values of this scale parameter for descriptor calculation close to 20 gave 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 III-C having values between 2 and 20. It was observed that small values of the threshold close to 5 gave better results. These parameters were optimized by grid search.

Table II shows an example confusion matrix. This matrix is resulted from the validation of TSURF descriptors on a grid of 20×20 and a scale of 20. Here it is evident that

Grid-size	LBP	LTP	TSURF
10	79,156	109,123	**37,981**
20	12,830	19,364	**6,025**
30	4,544	7,038	**2,090**
40	2,328	3,794	**990**
50	1,217	1,763	**464**
60	891	1,219	**336**
70	504	713	**148**
80	451	643	**154**
90	291	427	**92**
100	185	260	**46**

TABLE III

TIME TAKEN IN SECONDS FOR CROSS-VALIDATION

there was some confusion between grass and big-tiles and between grass and asphalt. Other than that, there was not much difficulty.

The time taken for 10-fold cross-validation for all of the image patches is described in Table III. These times are in seconds and are for training and classification through the random forests classifier used for validation. Here we can observe that for all grid-sizes, TSURF takes the least amount of time. The most amount of time is taken by LTP. This is natural, since TSURF has the smallest descriptor vector as described before.

VI. Conclusion

In this paper, we investigated the performance of different image descriptors at varying resolutions in an attempt to find the best descriptor for visual terrain classification on outdoor flying robots. Many of the current texture classification approaches use sharp images containing a single texture captured from a fixed camera angle under controlled conditions. We used images from real flights of the robot containing blurred images and consisting of multiple terrains. Along with two texture-based descriptors, LBP and LTP, we have tested another descriptor SURF. SURF is modified to be calculated on a grid placed on the image. LBP and LTP performed best at low resolutions only. LTP gave the best low resolution performance, however, it has one of the largest feature vectors. At higher resolutions TSURF performs much better than the other two descriptors. In addition TSURF has one of the smallest feature vectors and is fast to train. Hence, we have demonstrated that visual terrain classification can be performed at high resolution using TSURF.

References

[1] 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.

[2] 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.

[3] 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.

[4] Leo Breiman. Random forests. In *Machine Learning*, volume 45, pages 5–32, Hingham, MA, USA, October 2001. Kluwer Academic Publishers.

[5] P. Brodatz. Textures: A photographic album for artists & designers. New York: Dover, New York, NY, 1966.

[6] I.L. 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.

[7] C. Dima, N. Vandapel, and M. Hebert. Classifier fusion for outdoor obstacle detection. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2004)*, pages 665 – 671, New Orleans, LA, USA, 2004.

[8] Sara Erhard, Karl Engelbert Wenzel, and Andreas Zell. Flyphone: Visual Self-Localisation Using a Mobile Phone as Onboard Image Processor on a Quadrocopter. *Journal of Intelligent & Robotic Systems*, 57:451–465, 2009.

[9] 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.

[10] R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(6):610–621, November 1973.

[11] Robert Hudjakov and Mart Tamre. Ortophoto analysis for ugv long-range autonomous navigation. *Estonian Journal of Engineering*, 17(1):17–27, 2011.

[12] Yasir Niaz Khan, Philippe Komma, and Andreas Zell. High resolution visual terrain classification for outdoor robots. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1014 –1021, Barcelona, Spain, nov 2011.

[13] 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.

[14] Philippe Komma and Andreas Zell. Markov random field-based clustering of vibration data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*, pages 1902–1908, Taipei, Taiwan, October 2010. Best Paper Award Nominee.

[15] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 28, pages 1465–1479, 2006.

[16] B.S. 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.

[17] 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.

[18] M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[19] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[20] Boris Sofman, J. Andrew (Drew) Bagnell, Anthony(Tony) Stentz, and Nicolas Vandapel. Terrain classification from aerial data to support ground vehicle navigation. Technical Report CMU-RI-TR-05-39, Robotics Institute, Pittsburgh, PA, January 2006.

[21] 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.

[22] N. Vandapel, D. Huber, A. Kapuria, and M. Hebert. Natural terrain classification using 3-d ladar data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2004)*, pages 5117–5122, New Orleans, LA, April 2004.

[23] 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.

[24] P. Vernaza, B. Taskar, and D.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.