FAST ONBOARD TEXTURE ANALYSIS FOR PLANETARY EXPLORATION

David R. Thompson[1] **and Nathalie A. Cabrol**[2]

[1] *Jet Propulsion Laboratory, California Institute of Technology*
[2] *SETI Institute / NASA Ames Research Center*

ABSTRACT

Texture analysis holds significant promise for improving the science return of remote planetary exploration. Communications with exploration spacecraft suffer significant latency and bandwidth constraints; onboard image understanding can summarize large datasets and select representative images for transmission. We present image texture descriptors that satisfy the strict computational requirements of flight processors. We use the integral image transform to produce pixel-level texture features. A decision tree using integer arithmetic computes pixel classifications far faster than texture analysis techniques requiring convolutions of steerable filter banks. A case study characterizes sedimentology in Mars Exploration Rover microscopic images. Here, the decision trees produce image descriptors that correlate strongly with geologists' classifications of sediment types in the Gusev crater region. This in turn permits a kernelized regression model that predicts soil compositions using image texture without explicitly identifying individual particles.

Key words: Autonomous Science, Computer Vision, Image Texture, Mars Sedimentology.

1. INTRODUCTION

Planetary science is entering a new era where onboard image analysis could significantly improve missions' science return [2, 3]. Robot spacecraft suffer from intermittent communications, transmission latency, and bandwidth constraints. Onboard image understanding lets spacecraft collect and interpret science imagery autonomously when human guidance is unavailable. The analysis can generate bandwidth-efficient summaries of data products for downlink. In addition, onboard analysis could recognize different terrain types, triggering opportunistic sensor measurements in response to novel morphology [4]. Finally, autonomous image understanding is useful on Earth to automate statistical analysis of downlinked data. The Mars Exploration Rovers have collected over 300,000 images for manual interpretation, but the relevant geologic analyses are labor-intensive and to date only a small subset has undergone comprehensive study.

This work investigates image texture analysis in the context of these planetary science applications. Previous studies have used texture to classify geologic samples [5], detect science targets such as rocks [6], and produce adaptive image transmission during autonomous site survey [7, 8]. However, space flight processors are far slower than those used in these terrestrial tests due to their strict power and radiation requirements [9]. In general we desire efficient approximate methods for characterizing image texture — a fast terrain characterization that can provide enough rough information to inform on-the-fly decision making.

This paper demonstrates decision tree algorithms that make real time texture analysis feasible under the strict computational requirements of space flight. We exploit the integral image transform [10] to compute fast wavelet-like filters. A binary decision tree based on these features yields a fast "any time" texture classification for each pixel. This can significantly improve run times relative to descriptors requiring explicit convolutions of steerable filters [11]. Next, we apply these descriptors to a case study involving microscopic images from the Gusev Crater region. Here texture descriptors correlate with the principal terrain types identified in a sedimentological survey of Mars Exploration Rover imagery [1]. We approximate manual particle analysis with a kernel regression model that infers particle size counts from image texture.

2. TEXTURE DESCRIPTORS

Many recent efforts to quantify natural image texture favor statistical techniques [12, 13, 14]. Among these, *Texton* analysis has recently appeared in concept studies for autonomous planetary science [6, 8]. Textons are pixelwise texture classifications produced by quantizations of filter bank responses learned from a suitable training set [12]. A common choice is the MR8 filter bank; it contains two rotationally-symmetric filters, a Gaussian and Laplacian of Gaussian, as well as two anisotropic filters, an edge and bar filter. One achieves rotational invariance by using only the maximum response across all edge and bar filter orientations. These filter responses at two scales produce an 8-dimensional feature vector for each pixel [13]. The classical texton algorithm quantizes the

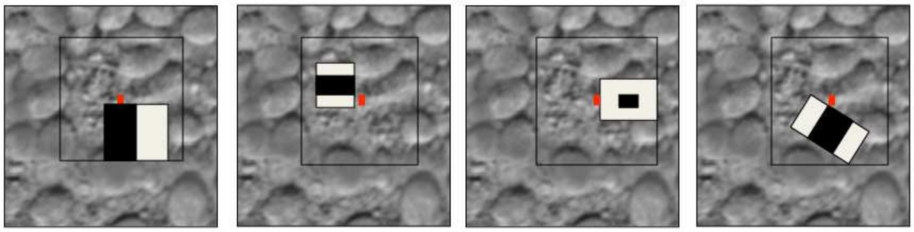

Figure 1. Haar wavelet features.

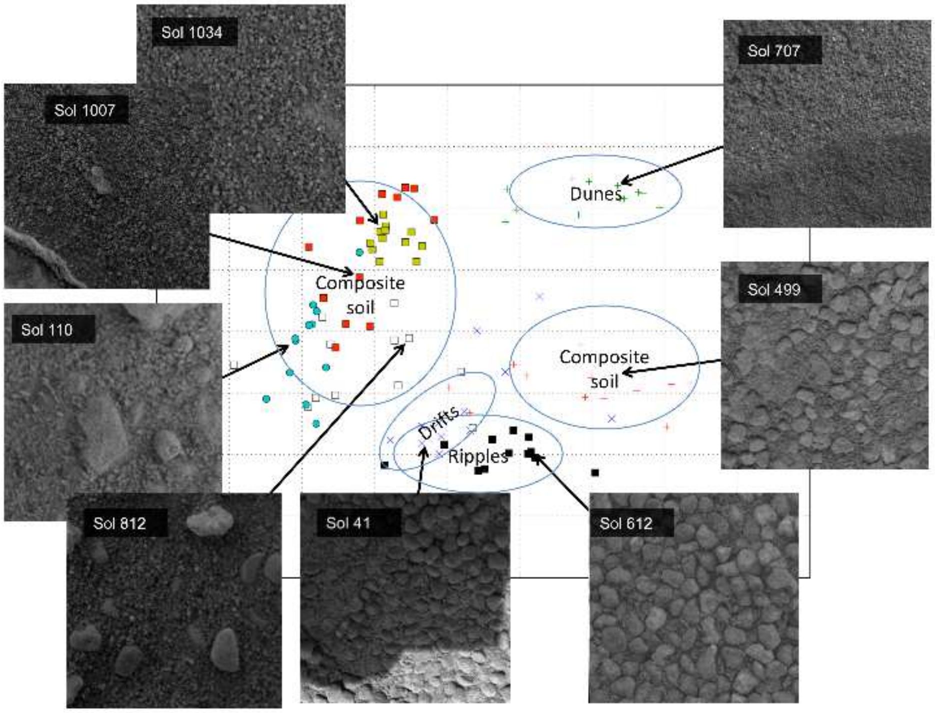

Figure 2. Automatic texture descriptors of Mars Exploration Rover Microscopic Image subframes from Gusev Crater. The descriptors are histograms of pixel classifications derived from a decision tree classifier. The classifier considers a feature set based on Haar-like filters computed quickly using the integral image. Each point in the scatter plot corresponds to a 128×128 *pixel subframe, with subframes from the same image receiving a common colored symbol. Here we visualize inter-class similarities by projecting subframe descriptors onto a 2D manifold with Kernel PCA. Ellipses show general sedimentological classes from Cabrol et al [1]. The manifold projection evidences significant clustering of subframes from similar images and soil types.*

MR8 responses over a training set with K-means clustering [15] to yield a dictionary of "universal textons" representing the canonical textures [13].

If we consider an image comprised of n pixels, and associate each pixel with a surrounding $m \times m$ image subwindow x, then the texton classification is a composition of an MR8 filter response function $\mathcal{F}(x)$ followed by a one-of-n nearest-neighbor classification $\mathcal{G}$:

$$\mathcal{G}(\mathcal{F}(x)) : \mathbb{R}^{m \times m} \mapsto \mathbb{R}^8 \mapsto \mathbb{I}^n \tag{1}$$

This result does not require a filter convolution. More generally we can represent the texture analysis with a single function $\mathcal{T}(x)$ that may or may not employ an arbitrary, intermediate feature space:

$$\mathcal{T}(x) : \mathbb{R}^{m \times m} \mapsto \mathbb{I}^n \tag{2}$$

One can approximate a known texture descriptor with any other general classification method by means of supervised learning, such as the MRF classification method of Varma and Zisserman [16]. The designer can choose an appropriate speed/accuracy tradeoff for the application. In practice filters reacting to larger-scale features (such as blobs and lines within each subwindow) can be more discriminative than individual pixel values. We combine filter responses into an intermediate feature space $\phi(x) = [\phi_1(x), \ldots, \phi_k(x)]$ for each image patch x.

2.1. Integral Image Filters

Our sedimentology case study computes subwindow features based the *integral image* caching technique. A data structure in the form of an image-sized matrix stores the sum of intensities of those pixels above and to the left [10]. Producing the integral image is a computation that scales linearly with the number of image pixels. The resulting data structure facilitates fast summations of pixel values over any rectangular region of the image using just three integer addition operations.

In particular, the integral image allows efficient computation of "Haar wavelet-like" filters [10]. These consist of oriented edge, bar, and center-surround filters. An extension by Leinhart *et al* includes diagonally oriented response regions [17]. We parameterize the filters according to their aspect ratio, type, scale, and location in the subwindow. Our bank consists of 15 filter styles, including boxcar, center-surround, bar, gradient, and edge filters at different orientations. For each filter style we consider five height-to-width aspect ratios ($\frac{1}{3}, \frac{1}{2}, 1, 2, 3$) at two different scales (3 and 6 pixel widths). They appear centered at some location on a local 5×5 grid. This yields up to 3750 distinct filters, amounting to a highly overcomplete feature vector $\phi(x)$ that describes each subwindow. For our domain a well-chosen subset of filters is a sufficient to separate the texture classes.

Here a decision tree [18, 19] produces fast classification decisions. Decision trees have been used previously for both modelling local texture and whole images [20]. Here we use a binary decision tree, splitting the training dataset at each tree branch by thresholding on a single Haar filter response $\phi_i(x)$. By computing filter responses as needed while traversing the tree the system need only evaluate a small subset of the entire filter bank. The distribution of training classes that reaches a particular branch describes a probability distribution over pixel classifications. This yields an "any-time" classification that can be refined as required by the application by traversing deeper into the decision tree. Generally the decision tree classifier requires just a few branches before matches traditional textons in efficiently recovering sedimentology types. However, it improves run times over these methods by an order of magnitude. Moreover, the Haar-like filters use integer arithmetic so they are appropriate for those flight processors and Field Programmable Gate Arrays that lack dedicated floating point hardware. We will consider both supervised and unsupervised learning methods for constructing the tree; the following sections consider each in turn.

2.2. Pixel-Level Descriptors

Our first strategy is a supervised learning approach. Here we hope to construct the decision tree that best approximates a known texture classification as in the texture trees of Bosch *et al* [21]. Specifically, we aim to reproduce the original MR8 textons, and our training labels are the traditional MR8 texton classifications. The training algorithm starts at the root node of the tree and introduces a single new binary split using the ID3 learning rule [19]. It then expands each resulting leaf recursively until it perfectly classifies the training set or achieves some desired maximum tree size. For the training subset reaching each leaf, the training algorithm considers all possible splits, i.e. splits at many thresholds θ for all possible filter responses $\phi_i(x)$. The class of the subwindow is a random variable given by $\mathcal{T}(X), X \in \mathbb{R}^{m \times m}$. Its entropy is given by an evaluation over all texton classes t:

$$H(\mathcal{T}(X)) = -\sum_t p(\mathcal{T}(X) = t) \log p(\mathcal{T}(X) = t) \tag{3}$$

At each branch, we define a cost function for supervised learning, C_S, to be the expected posterior entropy over possible textons given a binary-valued answer to whether the filter $\phi_i(x)$ response lies above or below the threshold θ:

$$C_S(\phi_i, \theta) = E[H(\mathcal{T}(X) \mid \phi_i(X) \geq \theta?))] \tag{4}$$

We choose the filter and threshold that results in the largest reduction in the cost function, using empirical probabilities based on the set of training data reaching that branch.

$$\begin{aligned} C_S(\phi_i, \theta) \quad &= \quad p(\phi_i(X) > \theta)\; H(\mathcal{T}(X) | \phi_i(X) > \theta) + \\ & \quad\quad p(\phi_i(X) \leq \theta)\; H(\mathcal{T}(X) | \phi_i(X) \leq \theta) \end{aligned} \tag{5}$$

During training we maintain a sorted list of feature responses for each data point; this allows us to quickly count population sizes for any feature and threshold and permits an efficient linear search. The run time classification generally requires only a small subset of filters and a handful of integer operations per filter.

This supervised algorithm seeks to reproduce MR8 textons, which are themselves just quantizations of continuous MR8 filter bank responses. Thus the final classification result lies several steps from the training image data. An alternative and perhaps more efficient approach simply quantizes the responses of the fast Haar-like filters on an unlabeled training set. Quantizing Haar-like responses directly obviates the need for an MR8 intermediary.

As in the supervised case, our clustering algorithm produces a hierarchical classification with binary splits at each decision branch. At each level we choose a split over features $\phi(x)$ and thresholds θ that minimize the inter-cluster variance of the resulting split. This is the same cost function used in k-means variational clustering strategies [22]. For A the set of subwindows whose feature $\phi_i(x)$ falls above the threshold and B the set of subwindows that fall below, we can express total cost $C_{\mathrm{US}}(\phi_i, \theta)$ for the unsupervised learning algorithm can be expressed as the sum of inter-class variances:

$$C_{US}(\phi_i, \theta) = \sum_i \mathrm{Var}(\phi_i(A)) + \sum_i \mathrm{Var}(\phi_i(B)) \quad (6)$$

We calculate the inter-class variances as follows:

$$\mathrm{Var}(\phi_i(A)) = \sum_{x_j \in A} \left[\phi_i(x_j) - \frac{1}{|A|} \sum_{x_k \in A} \phi_i(x_k) \right]^2 \quad (7)$$

Caching supports a fast a linear search over candidate thresholds. We maintaining a list of image subwindows sorted by the responses of the candidate filter. The variance decomposes as:

$$\mathrm{Var}(\phi_i(A)) = \frac{1}{|A|} \sum_{x_j \in A} [\phi_i(x_j)]^2 - \left[\frac{1}{|A|} \sum_{x_j \in A} \phi_i(x_j) \right]^2 \quad (8)$$

We cache both left and right terms of the subtraction and update them with datapoints that switch classes as we incrementally move the threshold. Thus the overall threshold search scales with $O(kn \log n)$ where k is the number of filters and n the number of datapoints.

2.3. Patch- and Image-Level Descriptors

We consider two methods for aggregating individual pixels' texture classes to characterize larger image regions. The simplest method simply counts the appearances of each class that occur in the image to produce a single histogram of textons. An alternative, the bag of words method [23, 24], describes images using unordered sets of local interest point descriptors or "words." Here our words are 20×20-pixel patches arranged in a non-overlapping grid that completely covers the image. Each patch is associated with a histogram of interior pixel texture classes $\mathcal{T}(X)$. We quantize a representative training set of patch histograms using k-means clustering; the result is a dictionary of 16 canonical patch classes. Then we generate an image-level histogram for the entire scene using the accumulated counts of each patch class.

Appropriate distance measures to compare patch- and image-level histogram descriptors include the χ^2 distance metric [13] or the earth mover's distance [25]. These can be used directly for nearest-neighbor classification strategies. Alternatively they define Mercer Kernels that permit a wide range of kernel pattern analysis techniques. Our case study analyzes image descriptors with the χ^2 kernel function. For two images i and j, with $h_i(w)$ and $h_j(w)$ representing $L1$-normalized counts of patch classifications w appearing in each, we can write the kernel function as follows:

$$k(h_i, h_j) = \exp\{-\frac{1}{b} \sum_w \frac{(h_i(w) - h_j(w))^2}{h_i(w) + h_j(w)}\} \quad (9)$$

Here b is a free bandwidth parameter that can be set manually or by cross-validation.

3. CASE STUDY: MARS EXPLORATION ROVER MICROSCOPIC IMAGERY

The Mars Exploration Rovers' Microscopic Imager, or *MI*, [26] is roughly analogous to the hand lens of the field geologist. It can characterize rock textures or provide a close view of soils for sedimentology surveys. The distribution of grain sizes and shapes yields important information about weathering and transport [27, 1]. This permits geologists to characterize the physical processes such as wind strength, saltation, and erosion that shape the planetary surface. We focus on recent work by Cabrol et al, [1] in which scientists characterize the particle size, angularity, and roundness of the archetypal soil types in the Gusev crater area explored by the Mars Exploration Rover "Spirit." Here, scientists label individual particles and compute statistics across these microscopic images. Statistics such as mean size, angularity, and roundness reveal several distinctive terrain types. Sols 1-156 traversed a geologic region known as the *Plains Unit*; here, aeolian processes appear to continually redistribute homogeneous mixture of small rounded particles. After Sol 156, Spirit moved into the *Colombia Hills* where soil mixtures are more heterogeneous with respect to shape, size, and sorting.

We evaluate texture descriptors with the 22 MI images from the soil survey. An initial preprocessing step reduces unwanted artifacts such as shadows and inconsistent focus. We use the local contrast normalization described in Lienhart *et al* [17]. We assign each pixel a value using the linear transformation which would map its subwindow to a constant mean and standard deviation. This removes global artifacts due to lighting changes (Figure **??** Left). In addition to shadows MI image frames exhibit inconsistent focus due to the large relative depth of field. Such changes can significantly affect quantitative measures of image texture. We compensate with a 5×5 pixel boxcar filter as a low-pass filter to slightly defocus each image. Base images have size 1024×1024; we divide each image into sixteen subframes for a complete dataset of size 352. We compute pixel-level texture descriptors using each of four methods: (1) a decision tree trained with the supervised technique, (2) a decision tree trained with the unsupervised technique, (3) a classic MR8 texton classification, and (4) a control group that simply uses the gray scale pixel intensity as the texture classification. Descriptors are learned using a reserved training set of 12 MI images not included in Cabrol et al.'s survey. For each method we generate image-level descriptors using both histogram and bag of words approaches.

3.1. Soil Type Classification

Cabrol *et al* provide four general sedimentological "soil types" for the images: Sol 41 exemplifies "drifts," Sol 612 "ripples," Sol 707 "dunes," and the others a general "composite" category. Here we evaluate each texture result for recovering these terrain type categories. We evaluate image descriptors by classifying frames with nearest-neighbor classification [15] using the χ^2 distance metric. Figure 3.3 shows classification accuracies for different pixel- and image-level texture descriptors. The horizontal axis shows the run time to produce descriptors for the entire megapixel image. We can force faster approximations by limiting the decision tree to a maximum depth; each point on the curve corresponds to an increase of 1 in this maximum. For example, the leftmost point corresponds to a single binary decision producing just two texture classes whose distributions are used to compute the nearest-neighbor classification.

After a single branch, the texture descriptors computed with the unsupervised approach already yield over 70% classification accuracy. The unsupervised algorithm is constructed to produce maximal separation of the training set at each branch so it yields a well-balanced tree whose nodes discriminate well after relatively few decisions. However, the supervised method eventually matches its performance as it emulates the complete set of MR8-based textons.

Note that the run time of the unsupervised method is generally longer for the same maximum depth; this reflects the increased balance of the unsupervised tree. While the most time-intensive operations of feature extraction are all coded in low-level C, these implementations could benefit from further optimization and we advise caution in comparing run times across algorithms. Nevertheless these tests support our intuition that the decision tree offers a considerable speed increase over the classic texton approach, which required over 30 seconds to filter each megapixel image.

Depth 7 decision trees using the image histogram descriptor produce the best overall result, with 99% leave-one-out classification accuracy. Both methods' performance drops beyond this depth, presumably due to overfitting. The two control groups (the actual MR8 textons and the intensity histogram) overfit and underfit respectively to yield classification errors of 0.8 and 0.68.

3.2. Particle Size Prediction

Sedimentology studies often characterize samples using the distribution of particle sizes. When only images (and not sieves) are available this profile requires manual labeling of each visible particle, a laborious process that requires hours per image. Here we extrapolate from training data to predict pixel size counts in seconds, based solely on image-level texture descriptors. The Cabrol study produces a *Wentworth scale* profile by grouping individual particles into discrete size bins. The study does not track particle counts within subframes so we use the same reported Wentworth profile for all sixteen subframes in each megapixel image. We aim to recover several scalar quantities - the particle count for each Wentworth bin, and from this the mean and median particle sizes in each subframe.

We employ kernel ridge regression [28] with the χ^2 kernel to map the image descriptors onto the logarithm of particle counts. We again use a leave-one-out strategy, reserving a single subframe and computing the regression result from the remaining subimages from that image and each other image in the training set. The regression accurately recovers the general profile of particle sizes. Figure 3.3 shows the average log error for different decision tree depths. Here again the unsupervised training strategy outperforms at lower depths. Both methods generalize best at depth 7. The supervised method utilizing image histograms produces the best overall accuracy.

Figure 3.3 illustrates the size count profiling for part of an MI Image collected on Sol 499; it exemplifies the "composite soil" class. The distribution of sizes is bimodal with grains of coarse sand ($\sim 1.2mm$) present on a background of very fine sand particles ($\sim 200\mu m$). This is reflected in the bimodal profile produced by manual analysis of the entire image (Chart A). Charts B and C are automatic profiles estimated using texture descriptors for each of the two subframes indicated by blue rectangles. Subframe B shows a preponderance of coarse grains; the automatic analysis predicts this accurately in the Wentworth profile. Conversely, subframe C shows fewer coarse grains and its profile estimate favors the smaller mode.

3.3. Manifold Learning

Good texture measures yield histograms that are consistent across images with similar sedimentology. We can visualize these relationships using dimensionality reduction by mapping the high dimensional histogram descrip-

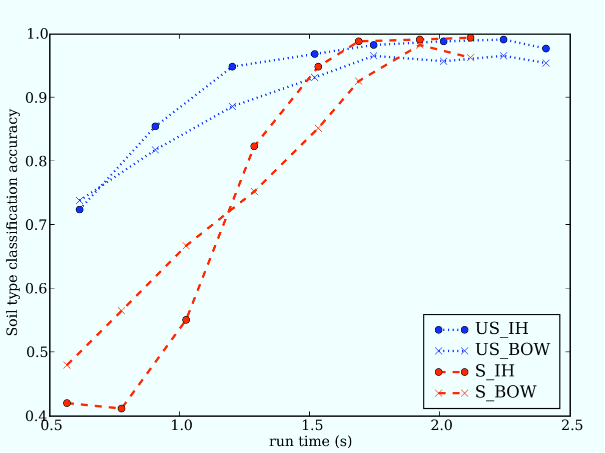

Figure 4. Soil classification accuracy rate for various pixel descriptors, image texture descriptors, and run times. US: Unsupervised tree learning rule. S: Supervised tree learning rule. IH: Histogram image descriptor. BOW: Bag of Words image model. Points on the curve show the decision tree evaluated to a maximum depths ranging from 1 *to* 8*.*

tors to a 2D manifold. We employ Kernel Principal Component Analysis [15] with the χ^2 kernel. Figure 2 shows the result applied to the unsupervised decision tree descriptors using image histograms. There is significant clustering within sample subframes from the same FI image. This suggests that texture descriptors correlate with physical contents of the scene. The manifold places ripple and drift classes in close proximity. The method flags Sol 499 as an outlier distinct from the other "composite soil" classes in the dataset.

There is considerable scope for future work. We will consider alternative unsupervised classification schemes such as the Random Forests method utilizing multiple parallel decision trees [21, 14]. More discriminative decision trees might be possible with splits involving more complex decision boundaries and multiple filter responses. Image-level classification may warrant a more sophisticated and efficient classification method than the nearest-neighbor approach we use here. Finally, a comprehensive validation is necessary before automated sedimentology analysis could supplement human labeling in science-quality data products. Nevertheless, our preliminary analysis suggests that explorer robots may be able to autonomously recover meaningful distinctions in terrain types to facilitate selective sampling and return.

ACKNOWLEDGMENTS

This work was supported by a JPL Strategic University Partnership Grant. Copyright 2009, Jet Propulsion Laboratory, California Institute of Technology. All Rights Reserved. U.S. Government Support Acknowledged.

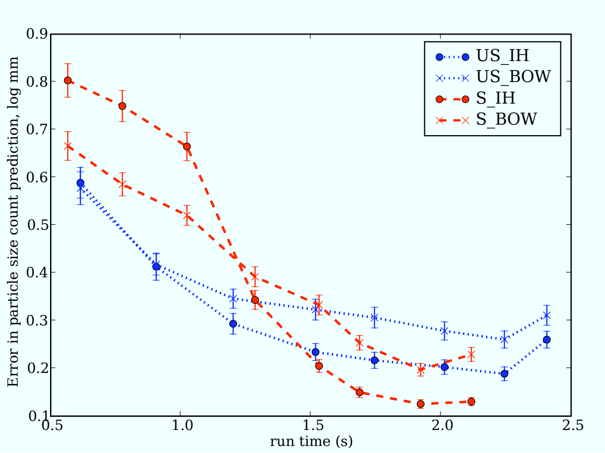

Figure 5. Wentworth profile showing error in log counts. Performance is given for different pixel descriptors, image texture descriptors, and run times as in Figure 3.3 Brackets indicate 95% *confidence intervals (presuming Gaussian error).*

REFERENCES

[1] N. A. Cabrol et. al. Soil Sedimentology at Gusev Crater from Colombia Memorial Station to Winter Haven. *Journal of Geophysics Research*, 113:E06S05, doi:10.1029/2007JE002953, 2008.

[2] V. C. Gulick, R. L. Morris, M. A. Ruzon, and T. L. Roush. Autonomous Image Analysis During the 1999 Marsokhod Rover Field Test. *J. Geophysical Research*, 106(E4):7745–7764, 2001.

[3] Rebecca Castaño, Robert C. Anderson, Tara Estlin, Dennis DeCoste, Forest Fisher, Daniel Gains, Dominic Mazzoni, and Michele Judd. Rover Traverse Science for Increased Mission Science Return. In *IEEE Aerospace*, March 2005.

[4] Trey Smith, David R Thompson, David Wettergreen, Nathalie Cabrol, Kimberly Warren-Rhodes, and Shmuel Weinstein. Life in the Atacama: Autonomous Science for Improved Data Quality. *Journal of Geophysics Research"*, 112:G04S03, 2007.

[5] H. Dunlop. *Automatic Rock Detection and Classifition in Natural Scenes*. Masters Thesis, Carnegie Mellon University, CMU-RI-TR-06-40, 2006.

[6] Heather Dunlop, David R Thompson, and David Wettergreen. *Multi-scale Features for Detection and Segmentation of Rocks in Mars Imagery*. 2007.

[7] David R. Thompson, Trey Smith, and David Wettergreen. Autonomous detection of novel biologic and geologic features in atacama desert rover imagery. In *Lunar and Planetary Sciences Conference*, March 2006.

[8] Thompson, D. R. and F. Calderón P. and D. Wettergreen. Intelligent Maps for Autonomous Kilometer-Scale Science Survey. 2008.

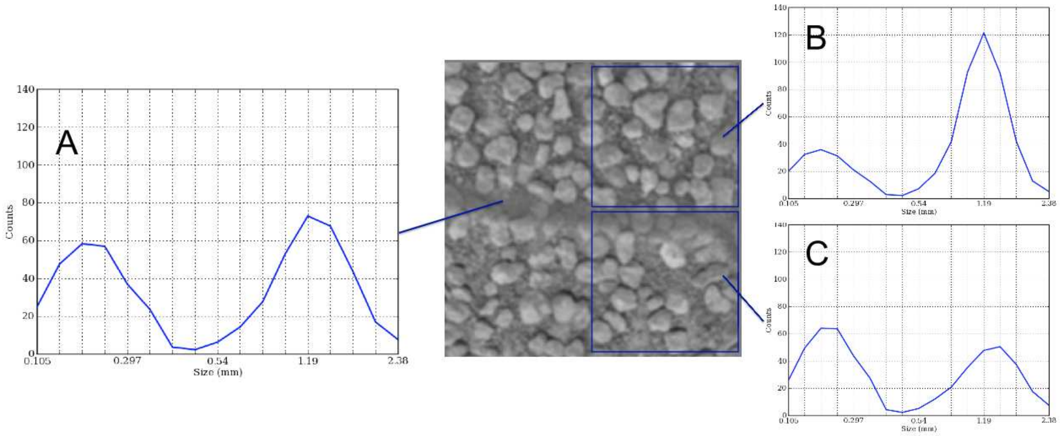

Figure 3. Prediction of Wentworth size profiles by extrapolating from training data. A: The manual labeling of entire image reflects a bimodal profile, with large and small particle sizes present. B: The automatic profile of the subframe at upper right shows a preponderance of coarse particles. C: In the automatic profile of the subframe at lower left, very fine sand predominates.

[9] Brian Cox. Updated avionics electronic trends. Technical Report OPFMS:033, 2007.

[10] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. CVPR*, 1:511–518, 2001.

[11] R. Sandler and M. Lindenbaum. Gabor Filter Analysis for Texture Segmentation. In *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*. IEEE Computer Society Washington, DC, USA, 2006.

[12] D.R. Martin, C.C. Fowlkes, and J. Malik. Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, pages 530–549, 2004.

[13] M. Varma and A. Zisserman. A Statistical Approach to Texture Classification from Single Images. In *Intl. J. Computer Vision: Special Issue on Texture Analysis and Synthesis*, volume 62:1-2, pages 61–81, 2005.

[14] J. Shotton, M. Johnson, R. Cipolla, T.C.R.D. Center, and J. Kawasaki. Semantic Texton Forests for Image Categorization and Segmentation. In *CVPR*, 2008.

[15] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. New York: Springer, 2008.

[16] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*, volume 2, 2003.

[17] Rainer Lienhart and Jochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. *ICIP*, 2002.

[18] T.M. Mitchell. Burr Ridge, IL: McGraw Hill, 1997.

[19] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Kauffman, 1993.

[20] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[21] A. Bosch, A. Zisserman, and X. Munoz. Image Classification using Random Forests and Ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, 2007.

[22] David C. J. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge U. P., Cambridge, UK, 2003.

[23] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[24] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman. Discovering Objects and their Localization in Images. In *ICCV*, volume 1, pages 370–377. IEEE Computer Society Washington, DC, USA, 2005.

[25] Y. Rubner, C. Tomasi, and L.J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[26] K. E. Herkenhoff, S. W. Squyres, et al. Athena Microscopic Imager investigation: Mars exploration rover mission and landing sites. *J. Geophysical Research*, 108(E 12):6–6, 2003.

[27] N. A. Cabrol, J. D. Farmer, E. A. Grin, L. Richter, L. Soderblom, R. Li, K. Herkenhoff, G. A. Landis, and R. E. Arvidson. Aqueous processes at Gusev crater inferred from physical properties of rocks and soils along the Spirit traverse. *J. Geophysics Research*, 111:E02S20, 2006.

[28] John Shawe-Taylor and Nello Cristiani. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.