

# A FAST METHOD FOR CLASSIFYING SURFACE TEXTURES

*Muntaseer Salahuddin, Mark S. Drew, Ze-Nian Li*

School of Computing Science, Simon Fraser University  
Burnaby, British Columbia, Canada, V5A 1S6  
{msalahud, mark, li}@cs.sfu.ca

## ABSTRACT

Surface texture classification is an important aspect of Computer Vision and a well studied problem. In this paper, we greatly increase speed for texture classification while maintaining accuracy. We take inspiration from past work and propose a new method for texture classification which is extremely fast due to the low dimensionality of our feature space. We extract distinctive features at a very early stage, thus removing the dependency on expensive and sensitive operations such as k-Means clustering which is used by much work in this field of research. We present experimental results on the Columbia-Utrecht Reflectance and Texture Database (CURET), to date the most challenging dataset for texture classification, and show that our method achieves comparable classification accuracy in comparison with the state-of-the-art, but at a 10-fold increased speed.

*Index Terms*— Image Classification, Image Texture Analysis

## 1. INTRODUCTION

Surface texture classification is the process of determining the category of an unknown material from a set of known categories. For example, many image retrieval algorithms try to compute local variations of intensity to select the correct image containing a particular object(s) from a given database of images. An important confounding problem is that in real life textured surfaces occur under variations of illumination and orientation, among other visual differences. These changes may make us perceive the same texture as different under different conditions. Until recently, most algorithms that tried to classify texture suffered from the effect of these variations of illumination and viewpoints.

In this paper we propose a novel approach for classifying texture under varying conditions of illumination and viewpoint, whereby we represent texture in a Weibull space. We then learn the information stored in each training image of a particular texture class by measuring its information entropy in this space. In the classification stage we choose for a test image its nearest neighbour in the Weibull space, i.e. – the training image which has the closest amount of information as in the test image. The result is a much faster algorithm which we compare with the state-of-the-art in terms of speed and accuracy. We perform all our experiments on the CURET database ([1, 2]), to date the most challenging database for texture images capturing variations in illumination and viewpoint.

The rest of the paper is organized as follows: in Section 2 we briefly review the literature in this line of research, in Section 3 we describe our approach, Section 4 is dedicated to our experiments, we analyze our results in Section 5, and finally conclude in Section 6 with some future directions for our work.

## 2. BACKGROUND

Over the past 30 years texture analysis has been widely studied and numerous methods have been proposed for describing image texture. Texture analysis methods were divided by [3] into four categories: statistical, geometrical, model-based and signal processing. For detailed study we point the reader to the following surveys on texture analysis methods: [4], [5], and [3].

Most of the earlier work assumed constant imaging conditions and therefore are limited in terms of performance when such variations are added to the image. An excellent example of a database of textures that incorporate variation in lighting and viewpoint is CURET [2]. This is to date the most challenging and largest database for texture [1].

Originally proposed by [6], Leung and Malik [7] provided the first working version of textons, “elementary particles” that constitute texture. The work of [7] (denoted LM) produced notable classification results on the CURET dataset. In the texton approach, filter responses are first generated by convolving an image with a bank of 48 filters (48 are used in LM) that include first and second derivatives of Gaussians at multiple scales and orientations, Laplacian of Gaussians, and Gaussians. A 2D texton is defined as the cluster centers in the filter response space, where each (sampled) pixel has a 48-vector of responses. However, images had to be carefully registered during the learning stage and then mapped to a 48 dimensional filter response space.

Rotationally invariant set of Gabor-like filters were proposed by [8] (Schmid denoted “S”) that also achieved good classification performance on the CURET dataset.

To date, the state of the art in terms of classification accuracy is provided by the work of [9]. They introduce the idea of Maximum Response Filters (MR8 and MR4) which are a collapsed sub-set of the “Root Filter Set” (RFS). The RFS filters are similar to the LM filters, but there are 38 of these instead of 48 (as in LM). Further, the filter responses from the 38 filters are collapsed by keeping the maximum response across orientations, thus reducing the number of filter responses to 8 and 4 for MR8 and MR4 respectively, for each image. This is done to achieve higher classification rates on directionally dependent textures that are hard to classify using just the S set. Moreover, they reduce the dimensionality of their feature space, which makes their clustering process simpler. Finally, they propose a greedy algorithm which

tries to reduce the number of models required to represent a class of texture without affecting classification accuracy.

A different line of research, such as in [10], is concerned with other properties of textured surfaces. In [11], they provide the notion of a sequential fragmentation process. Here a textured surface is perceived to be the result of magnifying into a large structure resulting in the structure resolving into smaller structures. The fragmentation process is stochastic in nature for almost all textures, especially those present in the CURET database. They propose the Weibull distribution as suitable (by performing the Anderson-Darling test and measuring goodness-of-fit) to measure the distribution of such textures as a function of orientation. As a result, the two Weibull parameters (shape of the distribution and scale of the distribution) that characterize a probability distribution function are capable of characterizing the spatial layout of stochastically ergodic textures. In [10] they move on to extract properties of texture (such as regularity, coarseness) based on the Weibull parameters. We take insight from their work and, using Weibull parameters, define our own feature space which has a much reduced dimensionality than other texture classification methods discussed in this section.

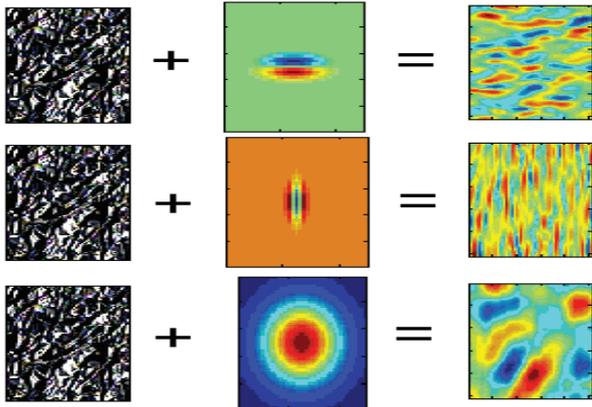
### 3. PROPOSED APPROACH

#### 3.1. Preprocessing Steps

The following pre-processing steps are applied before going ahead with any learning or classification.

We use the modified version of the CURET dataset which can be found at [12]. All processing is done on the cropped regions in this dataset and they are converted to grey scale and intensity-normalized to have zero mean and unit standard deviation. This normalization gives invariance to global affine transformations in the illumination intensity.

Second, filter banks (see Section 3.2) are  $L_1$  normalized, so that the responses of each filter lie roughly in the same range. In more detail, each filter  $F_i$  in the filter bank is divided by  $\|F_i\|_1$  so that the filter has unit  $L_1$  norm. This is to make the scaling for each of the filter response axes the same [13].



**Fig. 1.** Sample Filter Responses. An Example Image (1<sup>st</sup> column), Medium Scale Gaussian First Derivative at 0° (2<sup>nd</sup> column, 1<sup>st</sup> row), Medium Scale Gaussian Second Derivative at 90° (2<sup>nd</sup> column, 2<sup>nd</sup> row), Rotationally Symmetric Gaussian at Scale = 10 (2<sup>nd</sup> column, 3<sup>rd</sup> row), Corresponding Filter Responses (3<sup>rd</sup> column)

#### 3.2. Root Filter Set (RFS)

RFS consists of 38 filters, partitioned as follows: first and second derivatives of Gaussians at 6 orientations and 3 scales making a total of 36, and 1 Gaussian and 1 Laplacian of Gaussian filter. The Gaussian and Laplacian of Gaussian both have scale  $\sigma = 10$  pixels (these filters have rotational symmetry). The bar (first derivative) and edge (second derivative) filters both include 3 scales:  $(\sigma_x, \sigma_y) = \{(1,3), (2,6), (4,12)\}$ . These filters are oriented at 6 orientations: (0°, 30°, 60°, 90°, 120°, 150°). Sample filters and their corresponding filter responses on a textured surface are displayed in Figure 1.

#### 3.3. Histogramming

Our first contribution is at this stage where after applying RFS to each training image, we obtain a set of 38 filter responses for each image. We histogram each of these filter responses to speed up the process of Weibull parameter estimation for a particular filter response (see Section 3.4). Properties of the histogram (number of bins/bin size) will be analyzed in Section 5 and some interesting insights will be revealed.

#### 3.4. Mapping to the Weibull Space

At this point we observe the nature of textured surfaces proposed in [11], and therefore move to fit a 2-parameter Weibull distribution to each of the histograms we generated in the previous step. Therefore, we map a filter response to its corresponding location in the Weibull space. This is our second and most significant contribution. The Weibull distribution has the following probability density function (pdf):

$$f(x; k, \lambda) = \frac{k}{\lambda} (x/\lambda)^{k-1} e^{-(x/\lambda)^k} \quad (1)$$

for  $x > 0$ , and  $f(x; k, \lambda) = 0$  for  $x \leq 0$ , where  $k > 0$  is the *shape parameter* and  $\lambda > 0$  is the *scale parameter* of the distribution.

To estimate the parameters of such 2-parameter Weibull distribution we take the partial derivatives of the pdf with respect to the parameters. The two equations for shape and scale, respectively, are as follows:

$$\frac{\sum \log(x) x^k}{\sum x^k} - \frac{1}{n} \sum \log(x) - \frac{1}{k} = 0, \quad (2)$$

$$\lambda = \left( \frac{\sum x^k}{n} \right)^{1/k} \quad (3)$$

Equation 2 (estimating shape of the distribution) is solved using the Newton-Raphson method.

#### 3.5. The Final Model

We store the shape and scale parameters for each filter response for each image. This is our model for an image; i.e., every image is represented by a 76-vector (38 values for scale and 38 values for shape). And that is the complete, very simple model used here.

## 4. CLASSIFICATION METHOD

### 4.1. Distance in the Weibull Space

We note that the information entropy for a Weibull distribution is defined as follows:

$$H = \gamma \left(1 - \frac{1}{k}\right) + \log \left(\frac{\lambda}{k}\right) + 1 \quad (4)$$

where  $k > 0$  is the shape parameter and  $\lambda > 0$  is the scale parameter of the distribution, and  $\gamma$  is the Euler–Mascheroni constant with numerical value 0.577 (to 3 decimal places) [14].

The information entropy measurement captures the information stored in a Weibull distribution represented by a pair of  $\{shape, scale\}$  parameter. Therefore, we use equation 4 as basis of a distance measure between two images in our Weibull space. I.e., for every image we have a 38-vector of entropies (one for each shape, scale pair). Now for every test image we measure the  $L_2$  distance between its entropy vector and that of a training image. We classify the test image according to the class of its nearest neighbor amongst the training images. Further details of the experimental setup follow in Section 4.2. This entropy based distance measure is indeed new to this paper and has produced superior classification results.

### 4.2. Experimental Setup

We follow the experimental setup of [9] in order to compare our results with theirs and other previous results of texture classification (S,LM).

We perform three experiments to assess texture classification rates over 92 images for each of 20, 40 and 61 texture classes respectively. The first experiment, where we classify images from 20 textures, corresponds to the setup employed by [15] which is also used by [9]. The second experiment, where 40 textures are classified, is modeled on the setup of [7] also used by [9]. In the third experiment, we classify all 61 textures present in the Columbia-Utrecht database corresponds to the setup employed by [9]. The selected images result from the modified CURET dataset which can be found at [12].

Each experiment consists of two stages: generating a model for the class (using 46 images per texture class) and classification of novel images (the remaining 46 images per texture class).

To compare the run-time of our algorithm with that of [9] we conducted our experiments on a Windows based system with Intel 2.2GHz processor, 2GB of RAM running Matlab 7.1. We selected a set of 480 training and test samples and ran the classification procedure multiple times under consistent experimental environment to generate the average run times per texture for the algorithms (see Table 2).

## 5. RESULTS AND DISCUSSION

The results (percentage accuracy of classifying test images) of all three experiments are presented in Table 1. The first point we note from Table 1 is that in case of 20 texture classes our method, achieves classification accuracy rates very close to that of S, and LM, notwithstanding its simplicity and much faster speed (see Table 2). It is better than the MR4 approach in all cases and only slightly (~2%) worse than MR8 for the 20 class case.

**Table 1.** Comparison of Classification Accuracy for Varying Number of Texture Classes

Approach	# of Texture Classes		
	20	40	61
S	96.30	95.27	94.62
LM	96.08	93.75	93.44
MR4 (200 Textons)	94.13	92.07	90.73
MR8 (200 Textons)	97.83	96.41	96.40
MR8 (610 Textons)	-	-	96.93
Our Weibull based	95.98	92.28	91.52

However, for the case with all 61 classes in the database our method is some 5% worse than MR8. We will come back to this point but first we present the execution times per texture in Table 2.

**Table 2.** Execution Times Per Texture

Approach	Model Generation	Classification
Our Weibull based	~2.7s	$8 \times 10^{-4}$ s
MR8 (610 Textons)	~26s	$4 \times 10^{-3}$ s
MR8 (200 Textons)	~22s	$1.4 \times 10^{-3}$ s

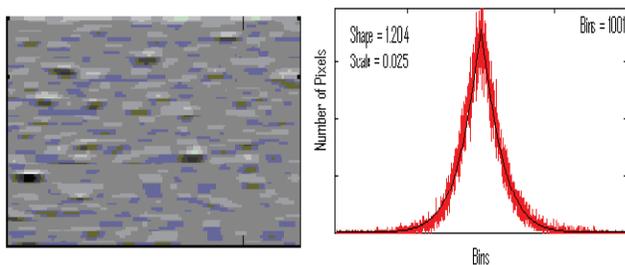
The method proposed in this paper is almost 10 times as fast as that of [9] (the MR8 approach). Although in the worse case (when classifying all 61 texture classes) our method does lose 5% accuracy, it can be argued that the method proposed here makes up for that in run-time. Of course, when using less number of texture classes (for example, 20) our accuracy is very close to that of MR8 but our method is 10 folds faster. We analyze the dimensionality and complexity of MR8 further and compare it to ours.

Up until the point of generating filter responses to the 38 filters from RFS both our approach and MR8 have identical complexity. Let  $\rho$  be the number of pixels in each filter response.

To generate the texton dictionary MR8 clusters the filter responses using a *standard k-Means* technique. There are a number of reliability issues related to this step. A standard k-Means technique is not guaranteed to converge in polynomial time. It has been shown by [16] that with high probability the k-Means algorithm may converge in super-polynomial time. They also go on to prove that the worse case complexity of k-Means on  $n$  data points is  $2^{\Omega(\sqrt{n})}$ . We have that,  $n$  for the MR8 approach is  $13\rho$  (13 sample images are chosen at random from each class). This would generate 10 textons for a particular class and the process has to be repeated for all 20 texture classes and for different samples, further adding to the complexity. Given that the clustering method is not guaranteed to converge (in our experiments the k-Means failed to converge even within 100 iterations), especially under such high dimensions, the ordering and initialization of the data becomes very critical. For example a simple re-ordering of initial neighbours, be they pixels from the filter response or even the column ordering of the MR8 features, will adversely effect the generation process of textons/cluster centres.

In contradistinction to these problems, our method extracts meaningful information (Weibull shape and scale parameters) from each filter response without being dependent on so many parameters. The transformation of a filter response to the Weibull space involves histogramming the data first. This gives rise to the

question that how many bins should there be in the histogram and what should be the size of each of them? Interestingly, in a fairly exhaustive set of experiments we found that it does not matter what the number of bins are as long as they are above a certain threshold (in our case this happens to be 1000). This is primarily due to the information present in the filter responses from specific filters and the Weibull fit process (essentially a least squares type estimate). A sample estimated probability density function is presented in Figure 2. Increasing the number of bins does not improve or reduce classification accuracy. Even decreasing the number of bins to as low as 201 only slightly reduces the classification accuracy (for 61 classes the accuracy drops by 0.04% only). So our algorithm is independent of the number of bins in the histogram.



**Fig. 2.** (Left) A filter response, (Right) Probability density function (the black line) generated from the estimated weibull parameters for the histogram of the filter response on the left

The second important criterion of our proposed approach is the convergence of the Newton-Raphson method while estimating the shape parameter of the Weibull distribution. Although we allow a maximum number of iteration of 30, in practice for 99.73% cases of the 213,256 filter responses (61 classes, 92 images from each class, 38 filter responses for each image) present in the dataset, the Newton-Raphson method converges in 5 iterations or less.

## 6. CONCLUSION AND FUTURE WORK

We have set out a new texture categorization method that recognizes texture surface distributions are often well represented by the Weibull distribution. Our method can dispense with a good deal of the complexity of the texton approach while maintaining comparable classification accuracy, notwithstanding a substantial speedup in the algorithm. The new entropy-based similarity measure has not been suggested before for judging nearness of distributions.

In future, we intend to improve on our results by incorporating a technique, such as that of [17], to pre-process the images and create their illumination invariant versions. This can be done as we have sufficient number of images representing all change in illumination. Moreover, we believe the images could be phase normalized using the Weibull Entropy measure. This would greatly reduce the effect of view-point variation and thus improve classification accuracy.

## 7. REFERENCES

[1] Columbia-Utrecht Reflectance and Texture Database, Dataset available at <http://www1.cs.columbia.edu/CAVE//software/curet/>, 1999.

[2] K.J. Dana, B. Van-Ginneken, S.K. Nayar, and J.J. Koenderink, "Reflectance and Texture of Real-World Surfaces," *ACM Transactions on Graphics*, ACM, vol. 18, no. 1, pp. 1-34, 1999.

[3] M. Tuceryan and A.K. Jain, *Handbook of Pattern Recognition and Computer Vision*, World Scientific Publishing, 1993.

[4] M. Petrou, *Image Processing: Dealing with Texture*, Wiley, 2006.

[5] M. Mirmehdi, X.H. Xie, and J. Suri, *Handbook of Texture Analysis*, World Scientific, 2008.

[6] B. Julesz, "Textons, the Elements of Texture Perception, and their Interactions," *Nature*, vol. 290, pp. 91-97, 1981.

[7] T. Leung and J. Malik, "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons," *International Journal of Computer Vision*, Springer, vol. 43, no. 1, pp. 29-44, 2001.

[8] C. Schmid, "Constructing Models for Content-based Image Retrieval," *Computer Vision and Pattern Recognition*, IEEE, vol. 2, pp. 39-45, 2001.

[9] M. Varma and A. Zisserman, "A Statistical Approach to Texture Classification from Single Images," *International Journal of Computer Vision: Special Issue on Texture Analysis and Synthesis*, vol. 62, no. 1-2, pp. 61-81, 2005.

[10] J.M. Geusebroek and A.W.M. Smeulders, "A Six-Stimulus Theory for Stochastic Texture," *International Journal of Computer Vision: Special Issue on Texture Analysis and Synthesis*, Springer, vol. 62, no. 1-2, pp. 7-16, 2005.

[11] J.M. Geusebroek and A.W.M. Smeulders, "Fragmentation in the Vision of Scenes," *In Proc. 9th International Conference on Computer Vision*, IEEE, vol. 1, pp. 130-135, 2003.

[12] Visual Geometry Group, Dataset accessible at <http://www.robots.ox.ac.uk/~vgg/research/texclass/index.html>, 2008.

[13] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and Texture Analysis for Image Segmentation," *International Journal of Computer Vision*, Springer, vol. 43, no. 1, pp. 7-27, 2001.

[14] J. Havil, *Gamma: Exploring Euler's Constant*, Princeton University Press, 2003.

[15] O.G. Cula and K.J. Dana, "3D Texture Recognition Using Bidirectional Feature Histograms," *International Journal of Computer Vision*, Springer, vol. 59, no. 1, pp. 33-60, 2004.

[16] D. Arthur and S. Vassilvitskii, "How Slow is the  $k$ -Means Method?," *In Proc. 22nd Annual ACM Symposium on Computational Geometry*, ACM, Arizona, USA, pp. 135-143, June 2006.

[17] M.S. Drew, M. Salahuddin, and A. Fathi, "A Standardized Workflow for Illumination-Invariant Image Extraction," *In Proc. 15th Color Imaging Conference*, IS&T, New Mexico, USA, pp. 36-41, Nov 2007.