Technical Report TR05-009

Simplifying Texture Classification

Robert E. Broadhurst
Department of Computer Science
University of North Carolina at Chapel Hill
reb@cs.unc.edu

Abstract

In texture classification modeling the full joint probability distribution of features is of questionable value. This paper demonstrates that marginal distributions of filter responses and marginal conditional distributions of intensity values over small neighborhoods are adequate to classify textures and can outperform methods using the joint distribution.

The use of the Earth Mover's Distance for marginal distributions is extended using PCA to build a Gaussian probability model for each class that captures the dependence between feature histograms. This framework is then generalized to include marginal conditional distributions for MRF models.

These methods are demonstrated on the Columbia-Utrecht database by classifying over 2800 images in all 61 texture classes. Results surpass those of Varma & Zisserman (CVPR '03) and Hayman (ECCV '04).

1. Introduction

Texture classification is one of the major problem areas in texture analysis. The four main stages of a texture classification algorithm are (1) feature selection; (2) probability distribution representation; (3) probability distribution distance measures; and (4) the classifier. A recent trend in texture classification has been to simplify different stages of this process. The work reported here extends the representation and distance measure developed in [7] to produce a simpler and more accurate texture classification algorithm. The resulting method is defined for a broader class of features and classifiers.

The MR8 classification algorithm of Varma and Zisserman uses a rotationally invariant filter bank, and clustering to estimate the full joint probability distribution [10]. Representative cluster centers define a texton dictionary, yielding a texton histogram representation for each image. The χ^2 distance measure and a 1-Nearest Neighbor (NN) classifier are stages (3) and (4) of this algorithm. A Support Vector Machine (SVM) classifier was recently used by [3].

Varma and Zisserman developed an alternative method, using different features based on a Markov Random Field (MRF) model, yielding slightly better results than their original algorithm [11]. The probability of a pixel's intensity in an MRF model is conditioned on the intensities from pixels in a local neighborhood. This probability was measured using the same texton approach, where each texton represents a configuration of neighboring intensities and the probability distribution of pixel intensities for that configuration. The original distance measure and classifier were used.

The use of an MRF model simplifies the texture classification process by eliminating the need for filter bank design and response collection. Furthermore, MRF models require smaller support (e.g. 7x7 vs. 49x49 [11]). However, more features may be required, which complicates clustering in the joint conditional probability space.

Another possible simplification when using filter banks is to measure marginal distributions instead of the joint distribution. This is equivalent to assuming the features are independent. Levina [7] developed a simple and effective framework for texture classification in this case by using the Earth Mover's Distance (EMD) [6, 9] and a 1-NN classifier. The use of marginal distributions greatly reduces algorithm complexity, by removing in particular the need to cluster in a high dimensional space.

In this paper Levina's filter bank based framework is extended by building a Gaussian probability model of each class for classification. This model can be built assuming independent or dependent feature histograms. The variation of each histogram in a class

can be jointly modeled while assuming the features at each pixel in an image are independent. A similar framework is then constructed for MRF models where the joint probability of a neighborhood is approximated by computing the probability of a pixel's intensity conditioned independently on each neighboring pixel. Results are reported for both frameworks developed here.

In section 2, we first review the EMD and marginal distribution representation as used in [7]. Then, section 3 describes our test methodology, database, and base framework. This framework is extended in section 4 to use Gaussian probability models, and generalized in section 5 for pixel intensities.

2. The EMD and Marginal Distributions

Rubner et al. introduced the EMD as a distance measure between distributions, which is based on the minimal cost to transform one distribution into the other [9]. Each distribution is represented either by a histogram or by a set of weighted cluster centers. This optimal transformation from one distribution to another relies on the existence of a dissimilarity measure between bins and can be thought of as a correspondence (not necessarily 1-1) between bins of the two distributions.

The EMD is a metric, if the dissimilarity between bins is a metric and the two distributions have the same total weight. Levina showed that, in this case, the EMD is equivalent to the Mallows distance between two probability distributions [6]. This case applies for the rest of the paper, thus the EMD and the Mallows distance are used interchangeably.

The Mallows distance between any two distributions P and Q in R^d, with finite p-th moments is defined as

$$M_p(P,Q) = \min_F \{(E_F \|X-Y\|^p)^{1/p} : (X,Y) \sim F, \; X \sim P, Y \sim Q\},$$

where X and Y are random variables corresponding to P and Q, respectively. That is, $M_p(P,Q)$ is the minimum expected difference between X and Y over all joint probability distributions F for (X,Y) such that the marginal distribution of X is P and of Y is Q.

Two special cases of distributions are considered for which the EMD can be further simplified. In the first case, all bins of P and Q have the same weight (equi-count histograms with the same number of bins) and so the EMD generates a 1-1 correspondence between the bins of P and those of Q. In the second case, the distributions are one-dimensional and an explicit solution that sorts the bins can be computed.

If P and Q satisfy both conditions, then they each have the same number of bins, n, and can be represented as n-dimensional vectors $P = (p_1, p_2, \ldots, p_n)$, $Q = (q_1, q_2, \ldots, q_n)$ with $p_1 \le p_2 \le \ldots \le p_n$ and $q_1 \le q_2 \le \ldots \le q_n$. Each vector element is a feature value representing 1/nth of the distribution. The 1-1 correspondence between the bins of P and Q is between matching vector elements, making the Mallows distance the L^p vector norm of P and Q:

$$M_p(P,Q) = \left(\frac{1}{n} \sum_{i=1}^{n} (p_i - q_i)^p \right)^{1/p}$$

Now a distance measure between two images can be constructed using this special case of the Mallows distance, as follows [7]. First represent each image by a set of features for every pixel. Each feature can be considered a random variable with an estimated probability distribution given by the image. The features are assumed to be independent and the distance between corresponding features in two images is defined as the Mallows distance of their estimated marginal probability distributions. Next, the distance between two images is defined as the product of the distances between corresponding features. Thus the final distance measure between two images is the product of the L^p vector norms of corresponding features. Levina demonstrated this framework for texture classification on the MeasTex and Brodatz databases using empirical distributions (keeping every pixel's value) and a 1-NN classifier [7].

3. Preliminaries

The classification problem examined in this paper, as in [3, 10, 11], is that of classifying a single image into one class from a pre-known set, whose descriptions are learned during training. The images considered are of a single texture, obtained under unknown illumination and viewpoint conditions.

Our test methodology is further defined in section 3.1, and section 3.2 introduces the filter bank used in section 3.3 and our first classification algorithm in section 4.

3.1. The CUReT Database

The CUReT database contains 61 texture classes consisting of materials imaged under 205 viewing and illumination conditions [2]. Each class contains images from one material that experience 3D effects such as specularities, inter-reflections, and shadowing, as shown in Figure 1. This causes large intra-class variability, making correct classification of the

database a challenging task. The limitations of this database are the lack of significant scale change and the limited in-plane rotation.

The experimental setup of [3, 10, 11] is followed in this paper. Of the 205 viewing and illumination configurations, 92 are used that have the largest minimum number of valid pixels across the samples. These 92 configurations are then equally split into test and training cases, yielding a total of 61 x 46 = 2806 training images and 2806 test images. Each image is then converted to grey scale and processed to have a zero mean and unit variance pixel intensity distribution.

One difference in our setup vs. that of [3, 10, 11] is the selection of valid pixels. For each image all pixels inside a cropped region, supplied with the database, are used, whereas [3, 10, 11] use an unspecified 200x200 pixel region. Our approach yields more data but has more pixels with inaccurate texture measurements due to boundary effects.

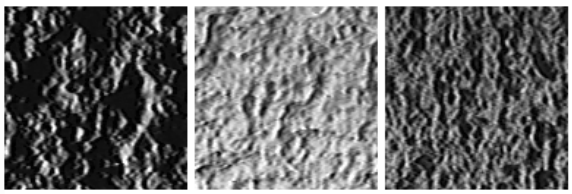

Figure 1: Three images from the "Plaster B" sample in the CUReT database, illustrating the large intra-class variability.

3.2. The MR8 Filter Bank

The MR8 filter bank consists of 38 filters and 8 filter responses [10]. There are two isotropic filters, a Gaussian and a Laplacian of a Gaussian (LOG), both at scale $\sigma = 10$. The 36 other filters include an edge (first derivative) filter at 6 orientations and 3 scales, and a bar (second derivative) filter at the same 6 orientations and 3 scales $(\sigma_x, \sigma_y) = \{(1, 3), (2, 6), (4, 12)\}$. Rotational invariance is achieved by storing only the maximum response over all orientations of a given filter type and scale.

3.3. Marginal Distributions and 1-NN Results

The marginal EMD framework described in section 2, using the MR8 filter bank and a 1-NN classifier is implemented here. The only simplification is to represent each filter response marginal with an equi-count histogram, instead of its empirical distribution. Bins of size 10, 100, or 1,000 reduce each marginal from the order of 100,000 values. With this setup results depend on the number of bins and the L^p norm (Table 1). An accuracy of 96.54% is achieved using 1000 bins and the L^6 norm, with degrading accuracy for higher p values. Results for the L^2 norm are included since this case will be used in section 4.

The algorithm of Varma and Zisserman, using the MR8 filter bank with a 1-NN classifier and a joint distribution estimate, achieves an accuracy of 96.93% and 97.43% with 610 textons and 2440 textons, respectively [10]. The SVM extension of this algorithm gives the best-known accuracy (98.46%) on this database [3]. Although these algorithms outperform the marginal implementation, they are computationally more expensive.

Improved results and reduced computational complexity are achieved in the next section by extending the marginal framework to other classifiers.

# Bins	L^2 Norm	L^6 Norm
10	94.44%	94.01%
100	95.69%	95.72%
1000	96.04%	96.54%

Table 1: Classification accuracy for the marginal EMD algorithm using the MR8 filter bank and a 1-NN classifier.

4. The Extended Gaussian Framework

One of the major drawbacks of a 1-NN classifier is its computational complexity, since for each test image the distance to every training image must be computed. Building a parametric model for the variability of each class is computationally more efficient and can be constructed using the framework discussed in section 2. In this section, two methods of constructing parametric models are presented.

4.1. Local Gaussian Models

One of the simplest parametric models is the multivariate Gaussian distribution. In this section models for each feature are built independently, so first consider a single feature. Then each image is a single distribution represented as a point in a high dimensional space, where the dimension is the number of bins used to represent each distribution. Define the Mallows distance between two distributions as the L^2 norm, making this a Euclidean space. Next, the dimensionality of the space is reduced to obtain a full rank covariance matrix for each class. Principal Component Analysis (PCA) is applied to all the

training images, generating the common directions of variation. A small number (5-15) of these modes are used and a class-specific Gaussian model is estimated.

The final probability of a class is given as the product of each feature's Gaussian model. Using this classifier with the marginal MR8 algorithm from the previous section gives the results in Table 2. Along with a significant gain in speed, accuracy is increased to 98.86% when 1000 bins and 10 eigenmodes are used. As the number of eigenvectors is increased, the classifier quickly spikes at its maximum accuracy, presented in Table 2, and then slowly drops off until one of the class covariance matrices is no longer invertible. One possible interpretation of this drop off is that higher modes contain unwanted intra-class variability, which is projected out when these modes are not included.

4.2. A Global Gaussian Model

The above approach assumes the distribution of each feature is independent, as well as the features themselves. Alternatively, the joint intra-class variation of feature marginals can be computed. One approach is to combine each image's marginal histograms into one vector and to apply PCA as above, forming a single multivariate Gaussian model for each class. While promising results were achieved, they were inconsistent across the number of bins, and accuracy degraded rapidly once the optimal number of eigenmodes was passed.

This lead to the final method: compute PCA modes as in section 4.1, combine these reduced marginal representations into one vector, and then apply PCA again forming once more a single multivariate Gaussian model for each class. The most remarkable accuracy of 99.54% is achieved when only 10 bins and 3 eigenmodes are used per feature (local modes) and 14 eigenmodes for the final Gaussian model (global modes). In addition, an accuracy of over 99% is achieved for all combinations of numbers of bins, local modes, and global modes, where at least 97% of the total variance is captured at the local level, the local model is not extremely overfit, and at least 99% is captured at the global level. For 10, 100, or 1000 bins at least 3 local modes and 14 global modes are necessary to meet this requirement. Table 3 shows results for the valid range of global modes using 10 bins and 3 local modes per marginal.

This algorithm is very computationally efficient and generates a compact representation of each image. To classify this database, it is sufficient to save only 80 values per image (10 bins for each feature), which can be further reduced to 14 values.

# Bins	# Eigenmodes	Local Gaussian
10	5	97.29%
100	10	98.61%
1000	10	98.86%

Table 2: Classification accuracy for the marginal MR8 algorithm with independent multivariate Gaussian models per feature.

# Global Eigenmodes	Global Gaussian
1	14.72%
2	33.50%
4	67.57%
6	90.16%
10	98.68%
14	99.54%
18	99.32%
22	99.39%

Table 3: Classification accuracy for the marginal MR8 algorithm using a single multivariate Gaussian model per class, 10 bins per marginal, and 3 local eigenmodes. At an accuracy of 99.54% only 13 of 2806 images are classified incorrectly.

The local PCA can lead to the projection of points into invalid histograms, the consequences of which have not been examined. PCA could also be used to speed up the 1-NN classifier when the L^2 norm is used; this produced similar results as before.

5. The Marginal MRF Framework

In this section an algorithm is developed using the complete local neighborhood of pixel values instead of filter responses. An approximation to the full joint probability of these intensities is developed and then the preceding framework is used for classification. Results are reported for neighborhoods of size 1x1, 3x3, 5x5, and 7x7, where the 1x1 neighborhood is supplied for comparison. Multi-scale neighborhoods, generated using a Gaussian filter, are then examined.

5.1. Conditional Probability Representation

Let x_c be a valid pixel in an image, with intensity $I(x_c)$ and neighbors $y_1, y_2, \ldots, y_n$. We seek to estimate the joint distribution

$$p(I(x_c), I(y_1), \ldots, I(y_n)).$$

Disregarding boundary effects, the marginal distributions of these intensities are identically distributed. Thus, a useful approximation cannot

assume fully independent intensities. First, the distribution is rewritten as

$$p(I(x_c)) \cdot p(I(y_1),...,I(y_n) \mid I(x_c)).$$

The intensities of the set of neighbors are assumed to be conditionally independent given x_c, yielding

$$p(I(x_c)) \cdot \prod_{i=1}^{n} p(I(y_i) \mid I(x_c)).$$

Finally, by assuming the intensities of the neighbors to be identically distributed and by the spatial symmetry of the set of neighbors, the final probability measure can be rewritten as

$$p(I(x_c)) \cdot \prod_{i=1}^{n} p(I(x_c) \mid I(y_i)) \quad (1).$$

A representation of these conditional distributions akin to that of the marginals is desired. The conditional probability $p(X|Y)$ is represented by measuring $p(X)$ with equi-count histograms for a fixed number of Y-intensity ranges. Y-intensity ranges are determined using an equi-count histogram as well, thus allowing each $p(X)$ to be measured with the same accuracy. To compute the representation of $p(X|Y)$ the values of Y are sorted and binned. Then for each bin the corresponding values of X are found, sorted, and binned.

Now, a distance measure needs to be defined between two conditional probabilities. Let $p(X|Y)$ and $p(Z|W)$ be two conditional probabilities measured at intensity ranges $y_1, y_2, \ldots, y_n$ and $w_1, w_2, \ldots, w_n$, respectively. First note that the distance measure needs only to compare $p(X|y_i)$ to $p(Z|w_i)$ for $i = 1, \ldots, n$ because X and Y, and Z and W, are identically distributed so any differences between Y and W will have corresponding differences between X and Z. Thus the actual intensity ranges of Y and W are ignored. The distance between $p(X|y_i)$ and $p(Z|w_i)$ is defined to be the Mallows Distance. These distances are then combined using the same L^p norm as in the Mallows distance to form the total distance between two conditional distributions. Ordering the distributions into a vector makes this distance correspond to its L^p vector norm.

5.2. Local Neighborhoods

For 1-NN classification, the total distance between two images can be constructed using probability measure (1) by multiplying together the individual conditional terms and the unconditioned term (measured as in section 2). The classification accuracy of this algorithm is shown in Table 4.

Modeling each factor of (1) by a multivariate Gaussian distribution (as in section 4.1) significantly improves accuracy (Table 5). Note that good results were still achieved using fairly coarse histograms (10 intensity ranges each with a 10 bin histogram). These results are comparable to those of an algorithm modeling the full conditional probability of a pixel's intensity given its local neighborhood [11]: 95.87%, 97.22%, and 97.47% accuracy for 610 textons and 3x3, 5x5, and 7x7 neighborhoods, respectively. The algorithm presented here slightly outperforms [11] for 610 textons; however [11] yields a higher accuracy (98.03%) for 2440 textons and a 7x7 neighborhood.

N	L^2 Norm	L^6 Norm
1	68.07%	70.81%
3	87.78%	88.99%
5	90.31%	91.77%
7	92.16%	92.52%

Table 4: Classification accuracy using NxN pixel neighborhoods and the 1-NN classifier. Each conditional probability is represented using 40x40 histograms (40 intensity ranges each with a 40 bin histogram).

N	10x10 Histograms	40x40 Histograms
1	77.51%	82.39%
3	96.36%	97.18%
5	97.26%	97.40%
7	97.58%	97.93%

Table 5: Classification accuracy using NxN pixel neighborhoods and a Gaussian model for each conditional probability with 15 PCA modes.

The algorithm presented in this section demonstrates that conditioning a pixel's intensity independently on each of its neighbors provides adequate information for classification.

5.3. Multi-scale Neighborhoods

This local neighborhood approach uses information from a much smaller spatial extent around each pixel than filter based methods. Larger neighborhoods are expected to improve accuracy, however there is a quadratic increase in the number of features with the scale of the neighborhood. Multi-scale neighborhoods can be used to alleviate this problem.

A pixel's multi-scale neighborhood includes the original 3x3 local neighborhood. A Gaussian filter is then used to generate 3x3 neighborhoods that summarize progressively larger spatial areas. For each level a Gaussian filter with $\sigma = 1.4$ is applied to the image used in the preceding level. The 3x3

neighborhood for a level is defined as the nine pixels twice the distance from the center pixel as in the previous level.

Classification accuracy is improved to 98.93% (Table 6) when four multi-scale neighborhoods are used. These neighbors supply filter responses, however, making this a simple filter bank based approach. This method also demonstrates one way in which the joint distribution of filter-based features can be approximated using conditional independence.

# Levels	10x10 Histograms	40x40 Histograms
2	98.47%	98.54%
3	98.68%	98.90%
4	98.79%	98.93%

Table 6: Classification accuracy using multi-scale 3x3 neighborhoods and a Gaussian model for each conditional probability with 15 PCA modes.

6. Discussion and conclusions

Accurate and computationally efficient methods for texture classification play an important role in texture research. This paper shows that the use of the Mallows (Earth Mover's) distance allows several simplifications, resulting in efficient classification algorithms.

One major simplification when using filter banks is the use of marginal distributions. This paper, as in [7], demonstrates the effectiveness of this framework. Furthermore, modeling the intra-class dependence between marginal histograms is shown to be a powerful tool.

Classifier choice strongly affects computational complexity and accuracy. In this paper, multivariate Gaussian probability models are shown to be superior to 1-NN classifiers often used in texture classification and achieve a classification accuracy of over 99% on the CUReT database.

The use of normalized pixel intensities as features further reduces algorithm complexity. A novel approach is presented here to approximate of the joint distribution of these intensity values using conditional independence. However, in contrast to [11], the most accurate results were achieved by filter bank rather than pixel intensity based methods.

In conclusion, this paper shows that algorithms based on low dimensional equi-count histograms, the Mallows distance between probability distributions, and Gaussian probability models are effective and efficient for classification. Algorithms presented here using pixel intensities and the MR8 filter bank achieve the best-known accuracy to date on the CUReT database.

7. References

[1] O. Cula and K. Dana. Compact representation of bidirectional texture functions. In *Proc.* CVPR, vol. 1, pages 1041-1047, 2001.

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

[3] E. Hayman, B. Caputo, M. Fritz, and J. Eklundh. On the Significance of Real-World Conditions for Material Classification. In *Proc.* ECCV, 2004.

[4] D. J. Heeger and J. R. Bergen. Pyramid-Based texture analysis/synthesis. In Proceedings AMC SIGGRAPH, pages 229-238, 1995.

[5] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. IJCV, pages 29-44, 2001.

[6] E. Levina, P. Bickel. The Earth Mover's Distance is the Mallows Distance: Some insights from statistics. In *Proc.* ICCV, pages 251-256, 2001.

[7] E. Levina. Statistical Issues in Texture Analysis. Ph.D. Dissertation, Department of Statistics, UC Berkley, Spring 2002.

[8] J. Puzicha, Y. Rubner, C. Tomasi, and J. M. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. In *Proc.* CVPR, 1999.

[9] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Proc.* ICCV, pages 59-66, 1998.

[10] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *Proc.* ECCV, vol. 3, pages 255 – 271, 2002.

[11] M. Varma and A. Zisserman. Texture Classification: Are Filter Banks Necessary?, In *Proc.* CVPR, vol. 2, pages 691-698, 2003.