



PERGAMON

Pattern Recognition 35 (2002) 1959–1972

PATTERN
RECOGNITION

THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

K-means Iterative Fisher (KIF) unsupervised clustering algorithm applied to image texture segmentation

D.A. Clausi*

Department of Systems Design Engineering, University of Waterloo, 200 University Avenue West, Waterloo, Ont, Canada N2L 3G1

Received 16 August 2000; received in revised form 21 May 2001; accepted 22 June 2001

Abstract

The K-means Iterative Fisher (KIF) algorithm is a robust, unsupervised clustering algorithm applied here to the problem of image texture segmentation. The KIF algorithm involves two steps. First, K-means is applied. Second, the K-means class assignments are used to estimate parameters required for a Fisher linear discriminant (FLD). The FLD is applied iteratively to improve the solution. This combined K-means and iterative FLD is referred to as the KIF algorithm. Two KIF implementations are presented: a mixture resolving approach is extended to an unsupervised binary hierarchical approach. The same binary hierarchical KIF algorithm is used to properly segment images even though the number of classes, the class spatial boundaries, and the number of samples per class vary. The binary hierarchical KIF algorithm is fully unsupervised, requires no a priori knowledge of the number of classes, is a non-parametric solution, and is computationally efficient compared to other methods used for clustering in image texture segmentation solutions. This unsupervised methodology is demonstrated to be an improvement over other published texture segmentation results using a wide variety of test imagery. Gabor filters and co-occurrence probabilities are used as texture features. © 2002 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Binary hierarchical clustering; Texture segmentation; Computer vision; Image segmentation; K-means; Fisher linear discriminant; Gabor filters; Co-occurrence probabilities

1. Introduction

Cluster identification is an important pattern recognition tool used in diverse fields such as biology, psychology, computer vision, remote sensing, and stock market analysis. Many different clustering approaches have been devised and these can be generally divided into two types: hierarchical and partitional [1]. Hierarchical methods group data in a sequential partitional sense, either in an agglomerative or divisive fashion. Partitional

methods group all the data using only a single partition. The K-means algorithm is a popular partitional clustering method [2]. K-means is an appropriate tool under the assumption that the clusters are hyperspheroidal since Euclidean distance measures are employed. However, real data sets seldom approach this hyperspheroidal idealization and the number of classes is not always known.

Clustering methods are defined to be completely unsupervised, however, in practice, this is a lofty goal. Many algorithms require some form of user control to produce satisfactory results. For example, ISODATA [3] requires the user to set many parameters to determine maximum and minimum sizes of clusters, to split

* Tel.: +519-888-4567x2604; fax: +519-746-4791.

E-mail address: dclausi@engmail.uwaterloo.ca (D.A. Clausi).

clusters with “high” spreads, and to merge “close” clusters. It is preferable from a user perspective to minimize the number of parameters that must be set.

Clustering methods are commonly incorporated in texture analysis algorithms. Texture analysis is used to identify objects in a scene, to provide depth cues within an image, to perform motion estimation, and for other computer vision tasks. Many different methods exist for generating texture features in such circumstances. Popular methods include co-occurrence probabilities [4], model based approaches such as Markov modelling [5–7], and multi-channel filtering techniques [8–10].

The identification of scene texture is challenging for a number of reasons. These include the ability to separate similar textures, identify poorly defined textures, and indicate the exact number of texture classes within the scene. One of the more difficult aspects of texture recognition is providing an all-purpose texture definition that could assist texture categorization. Wright [11] provides a generic definition: “Texture is the spatial distribution of intensities in image regions perceived by normal human observers to be *homogeneous* throughout the region”. The meaningful provision of a definition for homogeneity in computer texture implementations is challenging since textural homogeneity tends to be a learned characteristic.

This paper will demonstrate the design and implementation of a novel technique (the binary hierarchical KIF clustering algorithm) that is able to generate successful clustering of texture feature data based on a wide variety of test imagery. As a first step, existing clustering strategies for image texture segmentation using Gabor filters are presented (Section 2). Then, the design and implementation of the binary hierarchical KIF algorithm is described (Section 3). The technique is applied to seven diverse images to demonstrate the robustness of the technique (Sections 4 and 5). Discussions and conclusions complete the final section (Section 6).

2. Background

To perform image texture segmentation, texture features can be collected on a pixel by pixel basis. Pattern recognition methods are then used to group these features into appropriate classes to achieve the segmentation. Multi-channel filters have been used quite successfully to generate features from digital imagery for texture analysis [8,9,12,13]. These methods generally involve applying a particular filter bank to an image. Since each pixel has a different response to each of the N filters, then an N -dimensional feature vector that can be classified using pattern recognition techniques can represent each pixel. In this section, examples using Gabor filters for the purpose of performing texture segmentation are provided.

Then, based on this information, the contributions of this paper are described.

Dunn et al. [14,15] perform image segmentation in a restricted manner using Gabor filtered outputs. Here, only two distinct textures are found in the image, each texture takes up one-half the image, the textures are separated by a single straight boundary, training samples are required (fully supervised), and only one filter is used to distinguish the two textures (although textures often have more than one dominant frequency/orientation pair). Dunn’s concept that only a limited number of filters is required to segment an image is quite important from a computational and human visual system perspective.

Jain and Farrokhnia have published a well-referenced texture segmentation algorithm [8]. This algorithm has a few restrictions.

- The methodology (use of the modified Hubert index) for determining the number of classes does not work effectively for most of the provided examples [16].
- The methodology for determining the number of classes involves an incremental strategy that requires the entire system be solved at least as many times as the number of classes, which is computationally expensive.
- The spatial coordinates of the pixels are used as features for the displayed segmentation results. Their example images have individual textures placed in a block-like fashion and/or have textures located in only one region of the image. Using spatial coordinates as features for such images provides a tremendous advantage for segmentation. If the same textures are located in more than one region of the image, the spatial coordinates will only confuse the classification. Thus, inclusion of the spatial coordinates in the feature set is not appropriate for solving the generic texture segmentation problem. Note that the spatial coordinates were not used by one of the same authors in a subsequent publication [17].

Mao and Jain [17] implemented a self-organizing neural network for hyperellipsoidal clustering algorithm. They demonstrate the algorithm by segmenting textured images based on Gabor features. Their results are promising, however, they only display the best result of ten trials based on training a random selection of 1000 feature vectors (without class labels). The reduced number of feature vectors was required to minimize the computational time (noted to be approximately 1 h on a Sun Sparc 10).

Randen and Husoy [18] include Gabor filtered image texture segmentation in their comparison study. They comment that, although Gabor filters are commonly employed for image texture segmentation, their results provide no evidence that the Gabor filter should be preferred. Their segmentation experiments involve the use of training data in contrast to the emphasis here on

unsupervised segmentation. Panda and Chatterji [19] perform texture segmentation using tuned Gabor filters. They use the K-means algorithm to perform the segmentation. This is not an appropriate tool for the generic texture segmentation clustering problem since fully separable and/or hyperspheroidal clusters are not expected for all such problems and the number of classes may not be known beforehand.

This paper describes the K-means Iterative Fisher (KIF) binary hierarchical clustering algorithm. This algorithm is appropriate for situations where each class is represented by many features. Image texture segmentation is such a situation and is used for evaluating this algorithm. The algorithm properly groups image texture features, which lead to accurate image segmentation. This algorithm is an improvement on existing techniques for clustering texture features. No other algorithm is known that is capable of robustly performing the same task. Meritorious characteristics of the KIF binary hierarchical clustering algorithm include the following.

- *It is a wholly unsupervised algorithm.* The unsupervised solution is much more difficult to achieve than the supervised case. A single parameter, τ , is used to gauge the level of texture separability. This parameter seems to be constant for similarly derived imagery.
- *No a priori knowledge of the number of classes is required.* This is recognized as being one of the most difficult aspects when developing a clustering algorithm, yet the KIF algorithm consistently determines the correct number of classes. Also, no assumption about the a priori class probabilities is required.
- *The algorithm is tested using a wide variety of imagery.* For the provided test set, the number of textures per image ranges from two to seven, many of the textural boundaries are not straight, the number of pixels per class varies, disjoint regions of the same texture exist, and textures are sometimes visually similar. All of these aspects make the texture segmentation problem far more challenging, and the KIF algorithm works successfully in all cases.
- *The algorithm is an improvement over published texture segmentations.* Most of the images used have been included in previous publications. The binary hierarchical KIF algorithm is consistently an improvement over these published images in terms of percent accuracy and computational performance.
- *The identical algorithm is used for all test cases.* Absolutely no tweaking of the algorithm is performed on an image to image basis.
- *It is relatively computationally efficient.* The KIF method, implemented in a binary hierarchical framework, is able to subdivide the feature set into appropriate classes. It is an improvement on using a Hubert modified index since it does not have to operate on the full set of feature vectors for as many classes in

the image. It is an improvement on neural network methods that cluster a feature vector subset to meet appropriate computational requirements.

- *It provides a non-parametric solution.* The KIF algorithm assumes no a priori assumptions about the probabilistic nature of the clusters. This is an advantage over algorithms (e.g. Autoclass) that require such a priori assumptions.

3. Design of binary hierarchical KIF algorithm

Here, the design of the binary hierarchical KIF algorithm will be described. To motivate the basis of the design, the KIF algorithm will first be described (Section 3.1). KIF alone provides a mixture resolving [1] solution for the clustering problem i.e. number of classes are known. Then, KIF is expanded into a binary hierarchical framework to define a wholly unsupervised clustering solution (Section 3.2). Three constraints will be imposed on the clustering algorithm. Simultaneously achieving all of these constraints in the context of image texture segmentation is quite difficult. However, to address generic texture segmentation, achievement of these constraints is important. These constraints are:

- (i) No a priori knowledge of the class cluster distribution is known. In fact, each cluster may be represented by a different distribution.
- (ii) No a priori assumptions can be made about the number of samples that belong to each class.
- (iii) No a priori knowledge of the number of classes is available.

3.1. Mixture resolving KIF implementation

The following three steps are used to perform the mixture resolving KIF algorithm.

Step 1: The K-means algorithm is used to find the centroids of each of the classes (number of classes known). If the clusters are fairly well represented then the centroids should be found and a general description of the segmentation generated.

Step 2: The estimates of the class parameters (mean, covariance) derived from the K-means class assignment estimates are used to produce a Fisher linear discriminant solution (FLD) [2]. The FLD will correct the class variance estimates, and produce an improved class representation.

Step 3: Since it is not certain that the covariances have been fully corrected following only one pass of the FLD, the clusters are improved incrementally by applying the FLD iteratively (iterative FLD or iFLD).

This approach of applying K-means first followed by iFLD will be referred to as the KIF (K-means Iterative Fisher) algorithm. The advantage of KIF is illustrated in

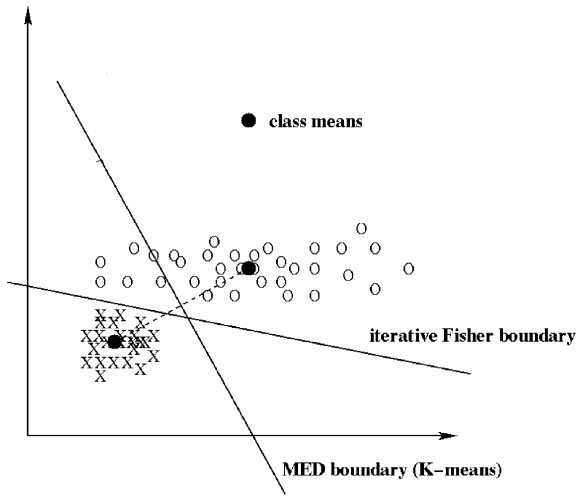


Fig. 1. Unsupervised classification of classes with different covariance matrices. Iterative FLD is able to improve on the classification generated by the K-means algorithm.

Fig. 1. Here, the K-means algorithm finds the means of the two class clusters, but the minimum Euclidean distance classifier erroneously assigns samples. The iterative Fisher linear discriminant (iFLD) generates a more appropriate separation of the classes using the estimated class covariances.

The FLD is determined by optimizing the Fisher criterion:

$$\tau(\omega) = \tau = \frac{\omega^T S_B \omega}{\omega^T S_W \omega}, \tag{1}$$

where S_B and S_W are the between-class and within-class scatter matrices. The greater the Fisher criterion, the larger the weighted distance between the class cluster pair. To project onto a one-dimensional space, S_B and S_W are defined as

$$S_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T$$

and

$$S_W = S_1 + S_2,$$

where

$$S_i = \frac{1}{N_i - 1} \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T.$$

The numerical subscripts represent one of the two classes under analysis. Unlike Duda and Hart, here the within-class scatter matrices (S_i) are weighted by the number of class samples to correct for unequal class sizes. The optimal ω is

$$\omega = S_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2).$$

The FLD is an appealing separability index since it is non-parametric in nature i.e. it does not assume a certain distribution, unlike other common measures which assume a normal class distribution.

The FLD can be implemented to generate a projection of a pair of n -dimensional clusters onto a one-dimensional vector. The projection provides an optimal separation of the two clusters. Then, a maximum likelihood classifier can be used where, in this case, there are sufficient samples to describe each one-dimensional Gaussian distribution. Only two classes can be compared at a time using the one-dimensional projection, so, to determine whether a sample belongs to one of C classes, $\binom{C}{2}$ comparisons are made. Using all comparisons, a sample assigned to a particular class the greatest number of times is the class to which the sample is assigned. Sometimes, two (or more) classes are assigned an equal number of times. Unresolved pixels are typically found on the boundaries separating textures. These pixels are appropriately assigned to the class according to the assignment of its spatially nearest pixel.

When do the iterations (Step 3) of the iFLD cease? As in any clustering problem, ascertaining when the optimal separation occurs is difficult. Here, the average of all class-pairwise Fisher distances is used as an indicator. The iterations stop when the average decreases. However, experience indicates that only a few iterations are required to correct for the class variances, so the maximum number of iterations is set to five.

Two methods were considered for generating the seeds for the K-means algorithm. Given no other information about the image, a suitable starting point is the first samples found in the image (beginning at the top left hand corner). Due to the assumed image periodicity created with discrete time forward and inverse FFTs used to generate the Gabor features, these seeds have the potential to lie near a boundary between numerous distinct textures (from each corner of the image). Another suitable starting point is to evenly space the seeds throughout the feature set. Given that textures are locally based, spreading out the seed points can provide appropriate starting points for identifying clusters. Since the second method tended to produce better results, it is used for all testing presented here.

3.2. Unsupervised clustering using a divisive hierarchical tree

Determining the total number of classes in any given feature set is a difficult problem [20]. In this section, the mixture resolving KIF algorithm will be extended into a fully unsupervised algorithm, the binary hierarchical KIF algorithm. When performing unsupervised clustering, an inter-cluster distance measure is required to determine whether a cluster is an independent class or is a subset

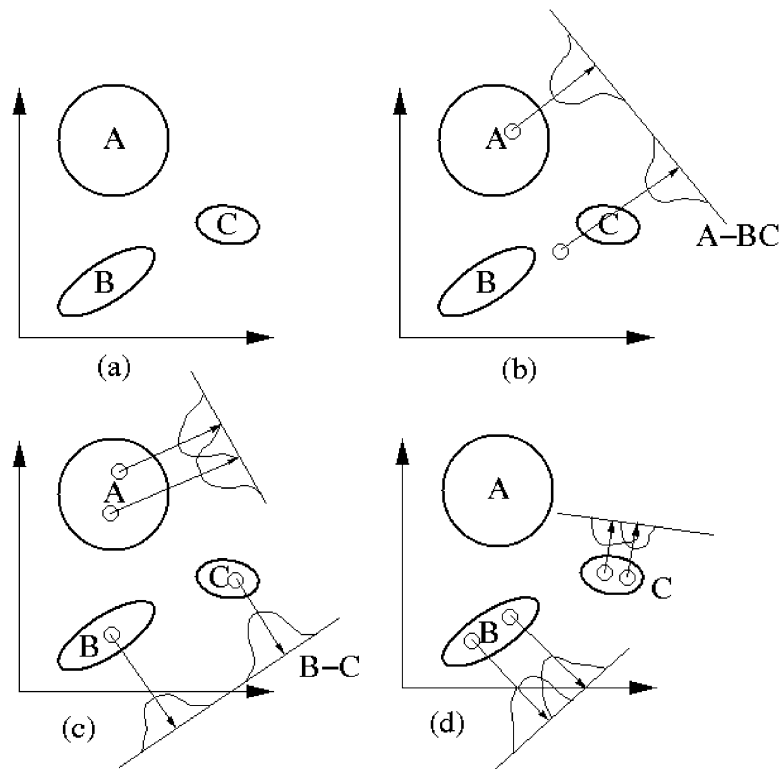


Fig. 2. Demonstration of binary hierarchical clustering. (a) Initial state of unclustered system. (b) Splitting of clusters into two distinct classes. (c) Class A cannot be split. Cluster BC is split into two distinct classes. (d) Classes B and C cannot be further subdivided.

of another class. The Fisher criterion provides such a measure (Eq. (1)).

A means to determine when the correct number of classes has been reached is required. A first attempt to determine the correct number of classes was performed by incrementing the number of classes from two and solving the entire set of feature vectors for each increment. Some measure then indicates when the correct number of classes has been reached. For example, Jain and Farrokhnia [8] implemented the modified Hubert index for this task. The average class pairwise inter-cluster Fisher criterion could also be used to perform the same task. With each increment of the number of classes, one of the following should occur: (a) a distinct class is identified from a larger grouping of clusters (desirable), (b) a cluster of classes is split into a pair of subclusters (desirable), or (c) a single class is split (undesirable). This method proved awkward because the most separable classes are not always the first classes identified by the K-means algorithm, so to determine when the algorithm should stop is difficult. In addition, single complex textures might be split before two distinct yet similar classes are separated and a method to determine when this would happen is not known. When multiple classes are considered, the average inter-class Fisher distance is

not an appropriate method to indicate when the iterations should stop since, although the average inter-class distance may drop, the best segmentation may not have been achieved. Another problem with this approach is that the entire system must be solved as many times as the number of determined classes, which is computationally intensive.

A more appropriate framework to determine the number of classes using KIF is a binary divisive hierarchical approach. Consider the case of three distinct classes in a two-dimensional feature space (Fig. 2). At the top node of the binary hierarchical tree, two classes are assumed. Applying KIF to the entire feature set generates clusters A and BC. The Fisher discriminant combines clusters B and C so that their joint projection resembles a single class on the A-BC discriminant. This projection generates a large Fisher criterion, which will allow the algorithm to proceed by assuming that A and BC are distinct. Then, KIF is applied in order to each of the two clusters, A and BC, to attempt to split them up. The splitting of the single cluster A generates a low Fisher criterion so cluster A is left as an individual class. The BC cluster is easily separated due to a large Fisher criterion. Here, the tree can proceed to the next level and try to separate the two clusters B and C. As when A was considered,

the splitting of B and C generates low Fisher distances and they are left as individual classes. The final clustering identifies A, B, and C as individual classes. Interestingly, this method is also able to determine if there is only one texture class in the image.

If the clusters are well separated in the feature space, there will exist a range of acceptable Fisher distances that can be used to generate an accurate unsupervised segmentation. Let $\hat{\tau}$ denote the threshold value. If $\hat{\tau}$ is set below the minimum of the range, then classes that should not be split will be split. If $\hat{\tau}$ is set above the maximum of the range, then clusters that should be split into classes or subclusters will not be split. By setting $\hat{\tau}$ to *any* value in the correct range, the segmentation will proceed properly. If all textures in the image are highly separable and well clustered then $\hat{\tau}$ has a large range. Setting $\hat{\tau}$ is a function of the quality of the feature vectors and the type of imagery being analyzed.

Fig. 3 summarizes the steps involved with performing the binary hierarchical KIF clustering algorithm. A stack data structure is used to store clusters. First, the feature extraction is performed. Then, the first cluster (represented by all the feature vectors) is placed on the stack. This item is popped and the KIF algorithm is used to try to split all the clusters into two. If the resulting Fisher criterion is larger than $\hat{\tau}$, then the cluster is split into two clusters as defined by the FLD. These clusters are placed on the stack. If $\tau < \hat{\tau}$, then the current cluster is deemed to be a single class. This process is continued until there are no longer any clusters on the stack.

The binary hierarchical KIF satisfies all three constraints proposed at the start of this section. Since K-means and FLD are both non-parametric in nature, no a priori assumptions of the class cluster distributions are made. The algorithm is able to perform without considering the expected number of samples per class. The binary hierarchical component of the algorithm is able to determine the number of distinct classes.

Recall the definition for texture indicated in the introduction. The term “homogeneous” is based on the relative nature of the textures as perceived by the viewer. This aspect is something that is learned by humans. However, we must indicate to the computer some parameter or threshold that provides an indication to the computer about what identifies unique textures. This homogeneity parameter is indicated by $\hat{\tau}$. This is the only parameter that must be set by the user to implement the binary hierarchical KIF algorithm.

4. Testing methodology

To solve the general texture segmentation problem, discriminating texture features are required. Gabor filters are used here to produce such features. Many papers demonstrate the usefulness of Gabor filters for the

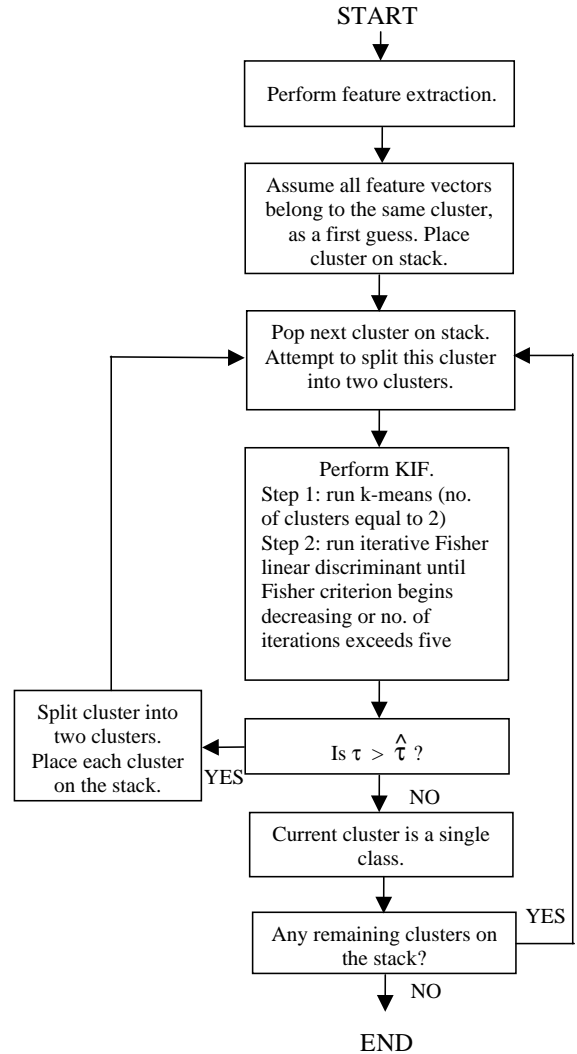


Fig. 3. Binary hierarchical KIF clustering algorithm. See text for details.

purpose of texture segmentation (see Section 2). Here, an appropriate Gabor feature set is determined by following an experimentally determined preferred filter set [21]. A preferred set of features is generated by using a filter bank with 30 degree filter spacing and bandwidth and octave spacing and bandwidth. Magnitude outputs are generated and Gaussian smoothing of the texture feature maps is performed. For a 256×256 image, only four higher octave bands are used ($8\sqrt{2}$, $16\sqrt{2}$, $32\sqrt{2}$, and $64\sqrt{2}$ cycles per image). The DC gain is set to zero to prevent contribution of the average grey level to the texture discrimination [8].

Seven test images are used to demonstrate the robustness of the KIF algorithm. Five of the images are pulled from the existing literature and the other two are created to demonstrate certain aspects of texture segmentation. The first six images are dimensioned to 256×256 pixels

Table 1

Completion times (minutes and seconds) and classification accuracies for using the mixture resolving KIF and binary hierarchical KIF algorithms for each image. Acceptable ranges of $\hat{\tau}$ are included for the KIF case. Completion times and classification accuracies are provided from the original publication for each image. This information was not always provided in the original paper

	Mixture resolving KIF		Binary hierarchical KIF			Published results	
	Total time	Accuracy (%)	Total time	Accuracy (%)	Range of $\hat{\tau}$	Total time	Accuracy (%)
Image class (sinusoidal boundary)	3:59	96.4	7:48	96.4	7.46–13.1		New image
Image class (straight boundaries)	3:46	95.3	9:19	94.8	9.51–12.3		New image
Image #3—Bigun & du Buf [23]	5:57	94.7	11:39	93.9	9.80–14.9	N.P.	N.P. ^a
Image #4—Mao & Jain [17]	3:08	95.1	9:37	97.3	10.5–16.3	> 1 h	94.2 ^b
Image #5—Jain & Farrokhnia [8,17,24]	11:27	94.7	12:08	94.8	11.4–13.9	54:24 [24]	^c
Image #6—Krishnamachari & Chellappa [25]	3:41	90.6	9:12	93.3	9.41–9.91	N.P.	85.5
Image #7—Unser [9]	0:18	94.2	0:53	95.7	6.49–12.8	N.P.	N.P.

^aN.P. indicates “not provided”.

^bIndicates best result of ten based on training 1000 randomly selected feature vectors.

^cResults for this example are not comparable since the published results identified only five classes instead of six. See text for details.

and the final image is 128×128 pixels. Both the mixture resolving KIF and binary hierarchical KIF segmentations are presented for each image. Mixture resolving is performed by giving the correct number of classes to the KIF algorithm. Wholly unsupervised segmentation uses the binary hierarchical KIF algorithm by setting the $\hat{\tau}$ parameter to an appropriate value.

5. Segmentation of test images and results

A summary of the results for each test case is presented in Table 1. This table provides an indication of the computational duration for each test, its segmentation accuracy, the appropriate range of $\hat{\tau}$ necessary to produce an unsupervised segmentation (for the binary hierarchical KIF), and a comparison to the results in the paper where the image was originally published. Such information was not always provided in the original publication (denoted by N.P. in Table 1). The algorithm duration is dependent on the number of iterations required to achieve a stable solution (for both K-means and iFLD) and the number of feature vectors processed. Tests were performed on a Sun Sparc Ultra 1 200E (200 MHz, 128 Mbytes RAM, 322 SPECint, 462 SPECfp). Total times provide a relative assessment of the mixture analysis and binary hierarchical KIF algorithms.

5.1. Image 1—bipartite image with sinusoidal boundary

The first test image (Fig. 4a) is a bipartite Brodatz [22] image containing the textures cork (D4) and cotton

(D77) separated by a sinusoidal boundary. Fig. 4b represents the segmentation obtained using only K-means and Fig. 4c illustrates the unsupervised segmentation using KIF. Since only two textures are considered, the mixture resolving KIF and binary hierarchical KIF approaches produce the same segmentation (both represented by Fig. 4c). Classification accuracies and completion times (in brackets) are 96.4% (3:59) and 96.4% (7:48) for the mixture analysis KIF and binary hierarchical KIF, respectively.

Comparing Fig. 4b and c illustrates the ability of the iFLD to improve the segmentation obtained by K-means alone. The tips of sinusoids are not accurately identified (Fig. 4b) when only K-means is used. The erroneously assigned pixels are found close to the boundary separating the two clusters in the feature space. K-means, using Euclidean distances alone, is unable to properly identify such pixels, however, the iterative FLD is able to account for the hyperellipsoidal shapes and produce an accurate cluster separation. The methodology readily segments the image accurately. For the unsupervised approach, the allowable range of τ is [7.46–13.1]. Any value of τ in this range is able to perform wholly unsupervised and accurate image segmentation.

5.2. Image 2—four class image

Here, the complexity of the texture segmentation is increased compared to the first image by using an image with an increased number of textures, a pair of similar textures, a pair of vague textures, and textures with varying textural resolutions. The test image (Fig. 5a) contains four Brodatz textures (corK-D4, cotton-D77, paper-D57, and raffia-D84) each with different textural

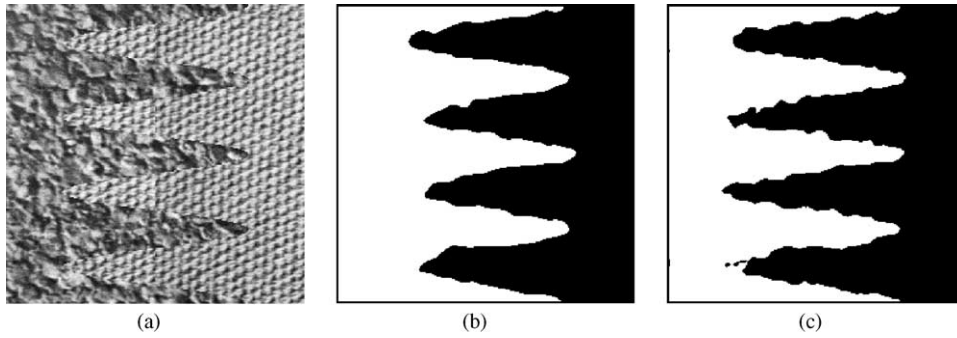


Fig. 4. (a) Image 1—two class Brodatz image with textures separated by a sinusoidal boundary. (b) K-means generates a result with an imperfect boundary. (c) KIF clearly improves this result.

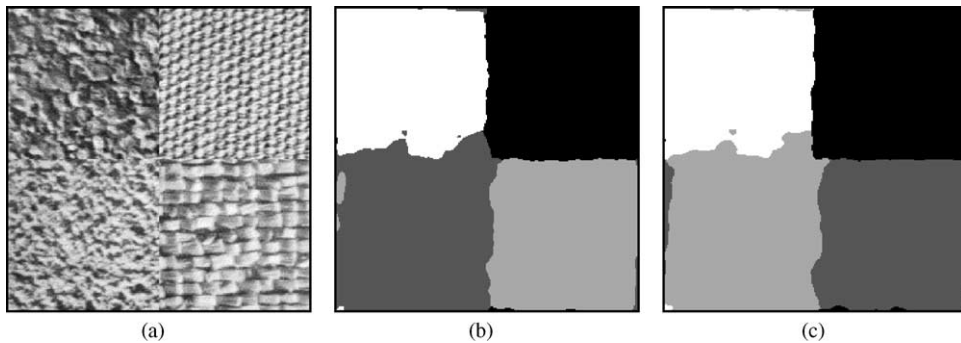


Fig. 5. (a) Image 2—four class Brodatz image. (b) Mixture resolving. (c) Binary hierarchical KIF methods are both successful.

resolutions. The cork and paper (on the left hand side) are poorly defined and quite similar in nature, especially compared to the cotton and raffia. Recall that the DC component of the Gabor filter is set to zero so that the average grey level does not play a role in the segmentation process.

Whether mixture resolving KIF (Fig. 5b) or the hierarchical KIF clustering (Fig. 5c) approach is used, the textures are clearly discriminated. Some error exists between the paper and cork textures, however, upon inspection of the image, the boundary between the textures is noticed to be poorly defined in these erroneous areas. If the observer did not know beforehand that the boundary between these two classes was a horizontal line, it would be difficult to identify. The range of acceptable $\hat{\tau}$ to perform an unsupervised segmentation is [9.7–12.3]. The classification accuracy for each of mixture resolving KIF and hierarchical KIF are approximately the same: 95.3% and 94.8%, respectively. Completion times are 3:46 and 9:19.

Table 2 indicates the ability of τ to parallel human distinction of the textures. This table is a matrix representing all class pairwise τ values produced from the clusters derived by the mixture resolving KIF analysis.

Table 2
Matrix of τ values for mixture resolving KIF segmentation of Image 2 (Fig. 5). The smallest τ represents the two textures that are the most similar (paper and cork). Cotton, a very distinct texture, has large τ 's with respect to the other three textures

	Cotton (top-right)	Paper (bottom-left)	Raffia (bottom-right)	Cork (top-left)
Cotton (top-right)	—	32.8	28.4	28.8
Paper (bottom-left)	—	—	18.9	11.9
Raffia (bottom-right)	—	—	—	42.7
Cork (top-left)	—	—	—	—

Small τ 's should indicate that the textures are similar and large τ 's should indicate that the textures are distinct. Cork and paper are the most visually similar pair of textures and their τ value (11.9) is the lowest in the matrix. Cotton is the most distinct texture and has larger τ

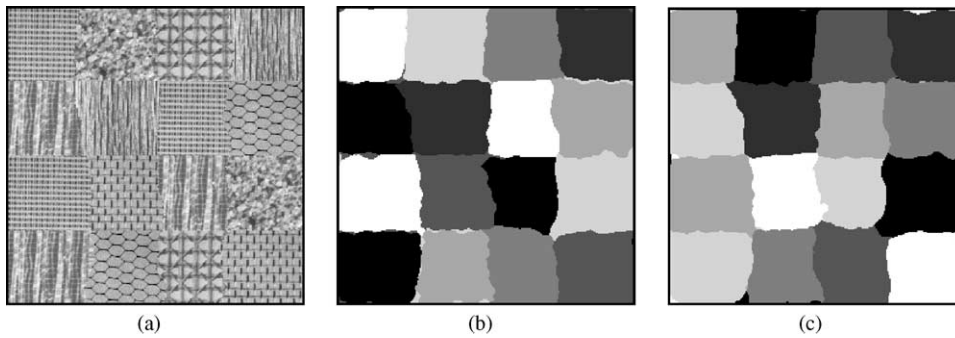


Fig. 6. (a) Image 3—seven class Brodatz image. (b) Mixture resolving has a successful segmentation. (c) Binary hierarchical KIF improves the class distinctions at textural boundaries.

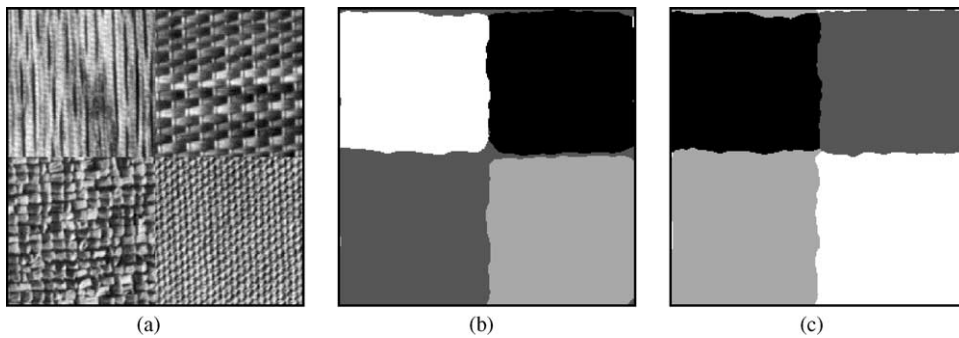


Fig. 7. (a) Image 4—four class Brodatz image. (b) Mixture resolving generates a successful segmentation. (c) Binary hierarchical KIF improves the class distinctions at the textural boundaries.

values with respect to each of the other classes. Similar results were noted for the rest of the test images, indicating that τ does provide a measure of the distinctiveness of textures.

5.3. Image 3—Bigun and du Buf seven class image

Adding more textures to the scene and placing the same texture in different locations provides additional complexity for texture segmentation. Bigun and Du Buf originally published the image represented in Fig. 6a [23]. Seven Brodatz textures are combined in a 16-patch mosaic so that each texture's boundary touches each of the others at least once. One texture is noticeably complex and the rest have regular patterns. Different textural resolutions are apparent. Horizontal boundaries are straight and vertical boundaries are generated by a random walk.

Visually, the binary hierarchical KIF approach (Fig. 6c) generates a segmentation that is marginally better than the mixture resolving KIF approach (Fig. 6b). For the unsupervised approach, $\hat{\tau}$ has a range of [9.9–14.9]. This segmentation compares quite well with

the segmentation produced by Bigun and du Buf (both are successful). Unfortunately, Bigun and Du Buf do not provide classification accuracies in the paper so that quantitative comparisons are not possible. Classification accuracies and completion times (in brackets) for the mixture resolving KIF and binary hierarchical KIF algorithms are 95.1% (5:57) and 93.9 (11:39), respectively.

5.4. Image 4—Mao and Jain four class image

This image (Fig. 7a) appeared in Ref. [17]. The following Brodatz textures are found in the image: wood (D68), straw matting (D55), raffia (D84), and cotton (D77). The segmented image in Fig. 7b (mixture resolving KIF) has appropriate segmentation except that the boundary between the cotton and the straw matting is classified as raffia. The same type of error occurs in the segmentation performed by Mao and Jain. However, the binary hierarchical KIF segmentation (Fig. 7c) is an improvement over the mixture analysis KIF (Fig. 7b) and Mao and Jain's solution (their Fig. 10e) since this boundary error is corrected. The acceptable range for $\hat{\tau}$ to perform the binary hierarchical KIF is [10.5–16.3]. The classification

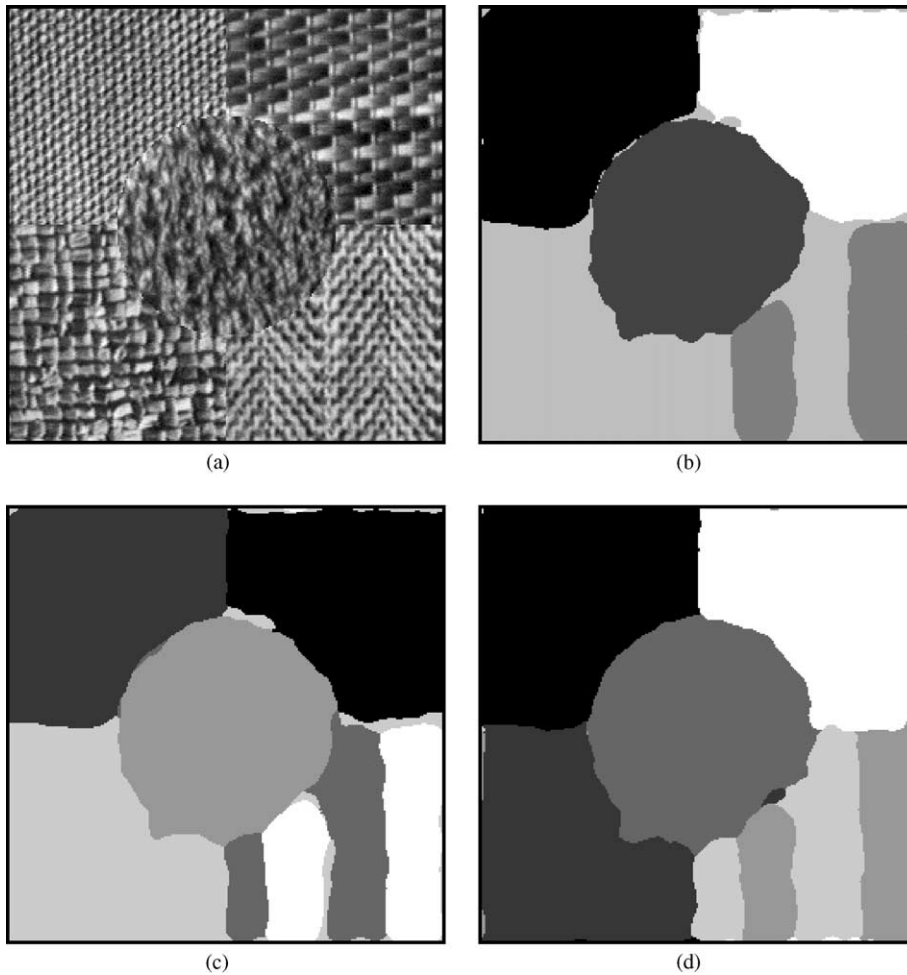


Fig. 8. (a) Image 5—five class Brodatz image. (b) Mixture resolving result if five classes are assumed. (c) Mixture resolving result if six classes are assumed. Herringbone texture will be segmented into two classes because Gabor filtered features are directionally sensitive. (d) Binary hierarchical KIF identifies six classes (unsupervised segmentation). Segmentation in (d) is an improvement over that in (c).

accuracy of the mixture analysis KIF and binary hierarchical KIF algorithms are 95.1% and 97.3%, respectively (with completion times of 3:08 and 9:37). Mao and Jain report classification accuracy of 94.2% for the identical image using the best result of ten trials that require selection of 1000 random feature vectors for training the neural network. Their solution takes approximately 1 h on a Sun Sparc 10 workstation (no clock speed provided) using their neural network designed for hyperellipsoidal clustering.

The binary hierarchical KIF algorithm generally breaks down the image by determining the most distinct texture first, the second most distinct texture next, etc. Here, for example, the scheme identifies cotton and groups the other three classes in the first step in the hierarchical process. Then, wood is identified from the

remaining two textures. This is appropriate since cotton and wood are more distinct relative to the other two textures.

5.5. Image 5—Jain and Farrokhnia image

The fifth image tested (Fig. 8a) was originally published by Jain and Farrokhnia [8] and subsequently published in the texture segmentation realm by Mao and Jain [17] and Chang et al. [24] also using Gabor filters to generate the texture features and an unsupervised clustering scheme. This image contains five Brodatz textures: cotton (D77), straw matting (D55), raffia (D84), herringbone (D17), and calf leather (D24).

Relative to the segmentations generated by the other researchers, an interesting result is obtained. The binary

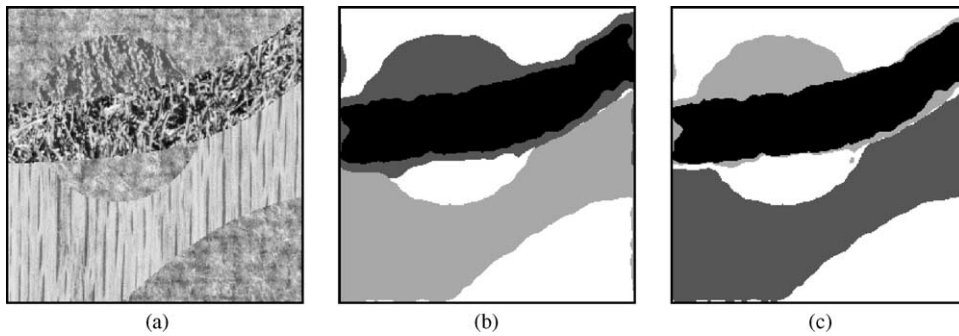


Fig. 9. (a) Image 6—four class Brodatz image with variety of boundary types and a variety of textures. (b) Mixture resolving KIF performs fairly accurate segmentation, however, the lower boundary of the grass texture is confused. (c) Binary hierarchical KIF is an improvement over (b) with reduced confusion of the lower grass boundary.

hierarchical KIF (Fig. 8d) automatically splits the herringbone texture (located in the bottom right hand corner) into two segmented regions: one region for the 45-degree grating and a separate region for the 135-degree grating. Due to symmetry and our higher understanding, we view the herringbone texture as one uniform texture. However, if using Gabor filters that are sensitive to orientation, one should expect a wholly unsupervised algorithm to split the herringbone texture into two separate textures. A 45-degree Gabor filter at the proper frequency responds strongly to the grating in the 45-degree direction. Similarly, a 135-degree Gabor filter set at the proper frequency is tuned to the 135-degree grating. As a result of filters having orientation sensitivity, separate clusters appear in the feature space for these two regions within the herringbone texture. In the unsupervised clustering algorithm presented here, the expected segmentation is obtained (Fig. 8d). Texture segmentations in Jain and Farrokhnia [8], Mao and Jain [17] and Chang et al. [24] do not subdivide the herringbone texture, as would be expected from an unsupervised algorithm and orientation sensitive features.

When five classes are assumed in the mixture resolving KIF case, then Fig. 8b results. Here, one strip of the herringbone texture is identified and the other is grouped with the raffia texture. Indicating six classes (Fig. 8c) properly segments the image, according to the characteristics identified by the Gaborian features. The binary hierarchical KIF approach produces visually better results than the mixture resolving KIF approach.

The classification rates for the six-class segmentations of the mixture resolving KIF and binary hierarchical KIF are 94.7% and 94.8%. The classification accuracies obtained here cannot be directly compared to classification accuracies obtained by the papers that perform segmentation on the same image since their solution does not subdivide the herringbone texture. For the record, here is a summary of their results for a five-class segmentation of

this image. Jain and Farrokhnia obtained a classification rate of 95.13% (when pixel coordinates were not used as features). Chang et al. report accuracies of 85.04% and an average run time of 54.4 min on Sun Ultra Sparc (no clock speed indicated). Mao and Jain reported a classification accuracy of 97.1% for their best result of ten tests using 1000 randomly selected training patterns. Their solution took about an hour on a Sun Sparc 10 workstation (no clock speed indicated) using the neural network clustering algorithm. In comparison, the solutions by the mixture analysis KIF and binary hierarchical KIF take about 12 min each.

5.6. Image 6—Krishnamachari and Chellappa four class image

This image (Fig. 9a), originally published by Krishnamachari and Chellappa [25,26], provides a variety of textures and boundary shapes. Textures used are grass (D9), calf leather (D24), wool (D19), and wood (D68). Note that the wool texture appears in three separate regions. Their segmentation technique uses a multiresolution Gauss-Markov random field (GMRF). The segmented images for mixture analysis KIF and the binary hierarchical KIF are represented in Figs. 9b and c, respectively. Both segmentations perform a fairly accurate segmentation, however, both confuse the lower boundary of the grass texture with the calf leather texture. However, these two segmentations are quite strong relative to the segmentations produced by Krishnamachari and Chellappa. In their work, the lower grass boundary is accurately identified, but the regional approximations within the grass, wool, and calf have local regions that are assigned to the wrong class. The visual segmentation differences between the two methods are probably a result of using different feature extraction methods (Gabor filtering versus GMRF). They report a classification rate of 84.54%. Here, the classification rates and total time required (in brackets) for each of mixture resolving KIF

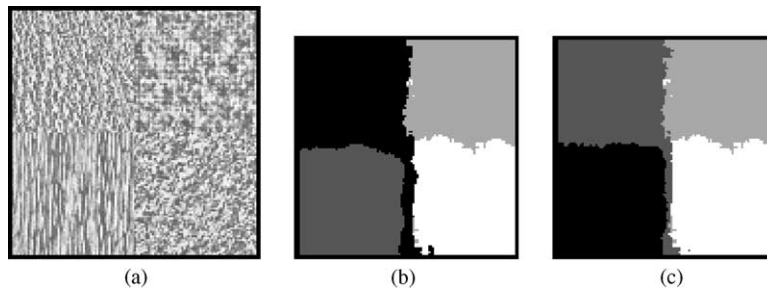


Fig. 10. (a) Image 7—four class MRF image. (b) Mixture resolving KIF and (c) binary hierarchical KIF segmentations.

and binary hierarchical KIF are 90.6% (3:41) and 93.3% (9:12).

5.7. Image 7—Unser four class image

Unser used this image (Fig. 10a) to demonstrate a wavelet-based texture segmentation method [9]. Four Markov random field (MRF) textures are represented. To demonstrate that the binary hierarchical KIF algorithm is able to work with other feature sets, this image is segmented using co-occurrence texture features [4]. Briefly, the co-occurrence method involves determining the probabilities of grey level pairs at certain relative distances and relative orientations within a fixed size image window. Features are determined by applying statistics to the probabilities. Here, the nearest neighbor pixels at orientations 0° , 45° , 90° , and 135° are used to create the probability sets. Then, the statistics contrast, correlation, and entropy are used to generate the texture features, as defined in Ref. [27]. The window size is 16×16 . A linked list technique is used to rapidly extract co-occurrence texture features on a pixel-by-pixel basis for the entire image [27].

The clustering algorithms are easily able to work with the co-occurrence feature set. Mixture analysis KIF (Fig. 10b) and the binary hierarchical KIF (Fig. 10c) are both able to accurately segment the image, except for the boundary separating the lower two textures. This same type of error is noticeable in Unser's results as well, however, the mixture analysis KIF and binary hierarchical KIF techniques are better able to classify other regions in the image. The classification rates are 94.2% and 95.7% for each of mixture analysis KIF and binary hierarchical KIF with completion times of 0:18 and 0:53. Unser does not provide a classification rate.

5.8. Results summary

Table 1 indicates that, for each segmentation, the mixture resolving KIF technique is completed in less time than the binary hierarchical KIF. The trade-off is that the mixture resolving technique requires a priori knowledge

of the number of classes, while the binary hierarchical method does not. Classification accuracies are all strong (in the mid-90s). In four of seven images with segmentation accuracies, the binary hierarchical KIF had a higher classification rate than the mixture resolving KIF method.

The value of $\hat{\tau}$ required to perform wholly unsupervised image segmentation is consistent from image to image. In fact, a $\hat{\tau}$ value of 12 will segment six of seven images (all but Image 6). If a $\hat{\tau}$ value of 12 is used for Image 6, then the grass and calf leather images are considered to be a uniform texture i.e. only three classes are determined for the image. This is not an unusual result since the grass and calf leather are quite similar in appearance. It is expected that, for similar sets of imagery, consistent values of τ can be used to segment the images accurately.

6. Discussion and conclusions

A successful method for segmenting textured images has been developed and demonstrated. This KIF (K-means Iterative Fisher linear discriminant) algorithm is a two step process. K-means is used to estimate the class prototypes indicated by the dense groupings. Class assignments based on the K-means result are used as an estimate for the class covariance. This information is used in an iterative class-pairwise Fisher linear discriminant to improve the classification. This technique can be utilized in two ways. A mixture analysis approach requires knowledge of the number of classes to solve the whole system at once. Also, when the KIF methodology is incorporated in a binary hierarchical clustering algorithm, a robust unsupervised segmentation is performed that does not require knowledge of the number of classes. A Gabor filter bank is used to generate the texture features for the first six images. To demonstrate the versatility of the clustering algorithms, co-occurrence probability texture features are used for the final image.

The mixture resolving KIF and binary hierarchical KIF are advantageous since they are non-parametric, require only a single parameter, and do not require any a priori

assumptions about the number of samples per class. The τ value (i.e. the Fisher criterion) represents the weighted distance between two clusters in the feature space. This, in turn, has a direct relationship on the visual distinctiveness of the textures representing each cluster. Large τ values generally indicate that the textures are visually distinct and small τ values indicate that textures are visually similar.

Seven test images that contain a variety of boundary shapes and texture characteristics are used. Accurate segmentations are generated using both the mixture resolving KIF and binary hierarchical KIF algorithm. Note that *identical* unsupervised algorithms are used for the texture segmentation in every case. No tweaking of the algorithm is performed to achieve the successful segmentations—exactly the same feature extraction and segmentation algorithms are used in each case. For wholly unsupervised segmentation, a common value of $\hat{\tau} = 12$ can be used for each case to produce proper segmentations. The segmentations performed using this algorithm are at least as strong and often better than the previously published segmentations.

The KIF algorithms are useful when compared to other algorithms used to cluster texture features for the purpose of image segmentation. KIF is a demonstrated improvement in the clustering over K-means. ISODATA requires the setting of many different parameters whereas the mixture resolving KIF requires setting only the number of classes and the binary hierarchical KIF requires only setting of $\hat{\tau}$. Computationally, neural networks generally require considerable computational overhead compared to the KIF algorithms.

Future work will be performed in the following areas.

- Since K-means is sensitive to the choice of the initial seeds, it may generate inappropriate solutions. A more robust method of determining the true dense regions in the feature space is desirable. The Equitz algorithm will be investigated for this purpose [28].
- The iterative FLD requires the most computational effort within the KIF algorithm. For each iteration, the within-class scatter matrix must be determined and its inverse calculated (S_w^{-1}). However, when each scatter matrix is recalculated, only a small percentage of the class feature vectors are added to or removed from the class. To accommodate this, it is possible to create a closed form solution for updating S_w^{-1} , to reduce the computational demands.
- To improve the computational requirements, it is possible to use features obtained from every fourth pixel (every other pixel in the x and y directions) for the analysis. Once the Fisher discriminants are obtained, they can be used to classify all pixels in the image. For some applications, it may be appropriate to spatially subsample the feature vectors and obtain accurate results to reduce computations.

- Given that texture boundaries are smoothed out when Gaussian smoothing is applied to the feature images, adaptive smoothing techniques might be better able to smooth the image without affecting the high contrast regions. This is appealing to increase the segmentation accuracy, however using adaptive filters would have a significant computational overhead when compared to the current linear smoothing.

References

- [1] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.
- [2] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, Toronto, 1973.
- [3] G.H. Ball, D.J. Hall, Some fundamental concepts and synthesis procedures for pattern recognition preprocessors, in: Proceedings of the International Conference on Microwaves, Circuit Theory, and Information Theory, Tokyo, 1964, pp. 281–297.
- [4] R.M. Haralick, K. Shanmugan, I. Dinstein, Textural features for image classification, IEEE Trans. Syst., Man, Cybern. SMC-3 (6) (1973) 610–621.
- [5] M.G. Bello, A combined Markov random field and wave-packet transform-based approach for image segmentation, IEEE Trans. Image Process. 3 (6) (1994) 834–846.
- [6] C. Kervrann, F. Heitz, A Markov random field model-based approach to unsupervised texture segmentation using local and global spatial statistics, IEEE Trans. Image Process. 4 (6) (1995) 856–862.
- [7] B.S. Manjunath, R. Chellappa, Unsupervised texture segmentation using Markov random field models, IEEE Trans. Patt. Anal. Mach. Int. 13 (5) (1991) 478–482.
- [8] A.K. Jain, F. Farrokhnia, Unsupervised texture segmentation using Gabor filters, Pattern Recognition 24 (12) (1991) 1167–1186.
- [9] M. Unser, Texture classification and segmentation using wavelet frames, IEEE Trans. Image Process. 4 (11) (1995) 1549–1560.
- [10] A.C. Bovik, M. Clark, W.S. Geisler, Multichannel texture analysis using localized spatial filters, IEEE Trans. Patt. Anal. Mach. Int. 12 (1) (1990) 55–73.
- [11] G.A. Wright, Feature Selection for Texture Coding, MASC Thesis, Department of Systems Design Engineering, University of Waterloo, Waterloo, Ont., Canada N2L 3G1, 1986.
- [12] L.L. Chen, A. Kundu, Unsupervised texture segmentation using multichannel decomposition and hidden Markov models, IEEE Trans. Image Process. 4 (5) (1995) 603–619.
- [13] A.C. Bovik, Analysis of multichannel narrow-band filters for image texture segmentation, IEEE Trans. Signal Process. 39 (9) (1991) 202–2043.
- [14] D. Dunn, W.E. Higgins, J. Wakeley, Texture segmentation using 2-D Gabor elementary functions, IEEE Trans. Patt. Anal. Mach. Int. 16 (2) (1994) 130–149.
- [15] D. Dunn, W.E. Higgins, Optimal Gabor filter for texture segmentation, IEEE Trans. Image Process. 4 (7) (1995) 947–964.

- [16] F. Farrokhnia, Multi-Channel Filtering Techniques for Texture Segmentation and Surface Quality Inspection, Ph.D. Thesis, Department of Electrical Engineering, Michigan State University, East Lansing, MI, 1990.
- [17] J. Mao, A.K. Jain, A self-organizing network for hyperellipsoidal clustering (HEC), *IEEE Trans. Neural Networks* 7 (1) (1996) 16–29.
- [18] T. Randen, J.H. Husoy, Filtering for texture classification: A comparative study, *IEEE Trans. Patt. Anal. Mach. Int.* 21 (4) (1999) 291–310.
- [19] R. Panda, B.N. Chatterji, Unsupervised texture segmentation using tuned filters in Gaborian space, *Pattern Recognition Letters* 13 (1997) 445–453.
- [20] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, Inc., Toronto, 1990.
- [21] D.A. Clausi, M.E. Jernigan, Designing Gabor filters for optimal texture separability, *Pattern Recognition* 33 (11) (2000) 1835–1849.
- [22] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover, New York, 1966.
- [23] J. Bigun, J.M. du Buf, N-folded symmetries by complex moments in Gabor space and their application to unsupervised texture segmentation, *IEEE Trans. Patt. Anal. Mach. Int.* 16 (1) (1994) 80–87.
- [24] K.I. Chang, K.W. Bowyer, M. Sivagurunath, Evaluation of texture segmentation algorithms, *Proceedings of CVPR*, 1999, pp. 294–299.
- [25] S. Krishnamachari, R. Chellappa, Multiresolution Gauss-Markov random field models for texture segmentation, *IEEE Trans. Image Process.* 6 (2) (1997) 251–267.
- [26] S. Krishnamachari, R. Chellappa, GMRF models and wavelet decomposition for texture segmentation, *Proceedings IEEE International Conference on Image Processing*, 1995, pp. 568–571.
- [27] D.A. Clausi, M.E. Jernigan, A fast method to determine co-occurrence texture features, *IEEE Trans. Geosci. Remote Sensing* 36 (1) (1998) 298–300.
- [28] W.H. Equitz, A new vector quantization clustering algorithm, *IEEE Trans. Acoust. Speech Signal Process.* 37 (10) (1989) 1568–1575.

About the Author—DAVID CLAUSI earned his B.A.Sc. (1990), M.A.Sc. (1992), and Ph.D. (1996) at the University of Waterloo (Waterloo, Ontario, Canada.) He initiated his career by working in the imaging industry (Mitra Imaging Inc.) and then started his faculty career as an Assistant Professor in the Department of Geomatics Engineering at the University of Calgary (Calgary, Alberta, Canada). In 1999, he returned to the University of Waterloo to his current position as an Assistant Professor in Systems Design Engineering. His research interests include automated image interpretation, digital image processing, pattern recognition with applications in remote sensing and medical imaging.