An Unsupervised, Agglomerative, Spatially Aware Texture Segmentation Technique

V. Lakshmanan*, V. DeBrunner, R. Rabin

Abstract

A novel method of performing multiscale, hierarchical segmentation of images using texture properties is introduced. The images are first requantized using contiguity-enhanced K-Means clustering. Morphological operations and region growing based on textural properties are used to arrive at the most detailed segmentation. Successively coarser segmentations are achieved by the use of inter-cluster distances in a dyadic, agglomerative technique. An objective way of quantitatively measuring the performance of texture segmentation algorithms independent of texture classification or training is also introduced. The method described in this paper is compared with some unsupervised texture segmentation algorithms reported in the literature. Our method performs better than other unsupervised and untrained texture segmentation algorithms on certain kinds of textured images. Results are presented on natural textures and on real-world scenes.

I. Introduction

Texture has long been an active area of research in the image processing community. In 1979, Haralick wrote a survey of the main approaches to texture [1], in which he cited papers written as early as 1955 [2]. Although researchers [3] have often noted the lack of a common definition, the literature in the field has long reached a consensus on what texture is, how it can be analyzed within an image and when texture analysis is useful in image processing. By consensus, texture in the literature always refers to properties at a scale larger than that of a pixel. Most commonly, texture analysis is employed when there is significant variation between the intensities of adjacent pixels even in the absence of an "edge" between the pixels. It is recognized that there is a difference in the meaning of texture depending on the nature of the images themselves. For

Corresponding author address: lakshman@nssl.noaa.gov. V Lakshmanan is with the National Severe Storms Laboratory (NSSL), 1313 Halley Circle, Norman, OK 73069, USA and the University of Oklahoma (OU). Victor DeBrunner is with OU. Robert Rabin is with NSSL and the University of Wisconsin, Madison.

example, it is recognized that natural textures tend to be random but artificial textures tend to be regular and periodic [4].

Image texture analysis methods use different descriptors of texture. The descriptors used capture a different part of the intuitive understanding of texture. Suggested descriptors include hidden Markov models [5], Markov Random Fields [6], image moments [7], coöccurrence [1] and correlation matrices [8] and filtering methods [9], [10], [11]. There exists no consensus as to which of these approaches provides the optimal texture vector. Havlicek [12] points out that several approaches ([13], [14], [15], for example) developed as a way to emulate the human visual system. The statistical approach, also referred to as the stochastic approach, assumes that texture is characterized by the gray value pattern in a neighborhood surrounding the pixel [16]. Local coherence and orientation estimates [17], Gabor filters banks [18], statistics of Gabor coefficients [19], [20], amplitude envelopes of band-pass filters [21], [22] and multiple components' frequency estimates [12], [23] have been used successfully.

Texture segmentation is simply image segmentation where the labeling of pixels is based on some measure of texture. Texture segmentation algorithms, then, differ on the actual form of texture used in the measurement. A vector of features is computed at each pixel. The segmentation is done based on this vector.

A. Motivation

Identifying and tracking storms is very useful for meteorological algorithms [24]. Storms thus identified and tracked may be used for visualization in a "storm-relative" sense, to study the evolution of storms, project storm movement in a short time period [25] and as inputs to feature detection algorithms [26].

There are numerous pattern recognition algorithms that have been developed on weather images, such as for rainfall estimates [27] and cloud classification [28]. On the other hand, segmentation of weather images is only now being addressed. We compared the method introduced in this paper with amplitude thresholding, a histogram-based method, a watershed approach, a Markov Random Field approach, Kolmogorov-Smirnov tests, a density-gradient method and MPEG-4 methods [29], [30] in the longer form of our work [56]. Many of these approaches could not deal with textured images. In this paper, we will show comparisons only with

explicitly texture segmentation methods.

In the meteorological community, the importance of multiscale segmentation has been often noted [24], [31], [32]. However, an electronic search of American Meteorological Society (AMS) journals from 1988 to the time of writing identified only one paper dealing with the segmentation of weather images. In that paper [33], the authors detail the difficulties that traditional segmentation algorithms have with satellite weather images because of the textural nature of clouds.

The textural nature of weather imagery makes robust segmentation for storm tracking purposes very difficult. For storm tracking to be useful, the identification and tracking algorithm should be completely automated. The identification algorithm should not require training, i.e. the algorithm should not expect to see examples of all the "objects" it must identify. The algorithm needs to be completely unsupervised because a weather forecaster cannot provide any sort of supervision in real-time operations during a severe weather outbreak. The data arrive once a minute in the case of composited radar data, and once every 15 minutes in the case of satellite imagery. The remotely sensed data change too fast (and are too voluminous) for sporadic human intervention to work. Storm "cells" (small scale features) should be capable of being identified. Because the notion of scale is natural in the storm tracking context, we would like to add the requirement that storms at various scales be identified, with their hierarchical structure intact. A multiscale tracking algorithm would be a significant improvement over current tracking schemes which concentrate either on small scales [24] or on large scales [31].

In trajectory and motion estimation problems on image sequences, the lack of robustness of the results of texture segmentation has often been noted [34]. Thus, robust segmentation of weather images is a problem that should be addressed by the electrical and computer engineering community. An extensive electronic search of IEEE journals and conference proceedings from 1988 to the time of writing found only two papers dealing with the segmentation of weather images. The first of those, [35], was content to identify pixels of the image that corresponded to precipitation, i.e. a binary segmentation. This would correspond to the coarsest level of segmentation in our hierarchical technique – a level of segmentation that is not very useful. The second paper [36] was a report of the performance of a Markov Random Field-based texture segmentation

approach [37] to the problem of segmenting satellite images. The segmentation was found to be not very robust across frames and extremely sensitive to initial conditions [36].

The standard techniques in optical flow were developed for quasi-rigid motions with salient and unchanging features [38], [39]. In meteorological images, the displacements are large, the structures morph and thus, there is a great deal of what the optical flow literature terms "spatial and temporal distortion". For sequences such as that of satellite weather images, a fluid-based approach that is compatible with larger displacements has been recently proposed [39]. It is possible that incorporating a fluid-mechanical term of the sort used by [39] would make the segmentation problem easier.

B. Organization of paper

The rest of this paper is organized as follows. In the remainder of this Section, existing approaches to K-Means clustering and multiscale segmentation are described. In Section II, our algorithm for performing unsupervised, agglomerative texture segmentation is described. In Section III, our method is contrasted with traditional multiscale segmentation methods and with traditional K-Means segmentation methods. In Section IV, a quantitative way of evaluating texture segmentation algorithms is introduced. Our method is then compared to other unsupervised, training-free, texture segmentation algorithms on Brodatz textures as well as on real-world images in Section IV and conclusions are made in Section V.

C. K-Means Clustering

K-Means clustering [40], [41], [42] is a clustering technique where the clustering proceeds by computing the affinity of each entity with each of K clusters that already exist. The entity is assigned to the cluster to which it is closest in the measurement space. The means are updated, either after a full pass through the data or immediately after a change in assignment, and the process is repeated for the next entity. The entities are cycled through until no entities are moved between clusters. K-Means clustering may be thought of as a vector quantization technique where the vectors are chosen adaptively from the image itself.

The traditional K-means clustering method uses either the Hamming or the Euclidean distance between the cluster mean and texture vector to determine the distance [43]. However, the traditional method does

not take into account the spatial characteristics of the data. Pixels that are spatially contiguous are likely to belong to the same class. K-Means clustering of images is improved if the choice of the vectors, as well as the association of the pixels to a cluster is done after incorporating a contiguity constraint [44], so as to minimize the global energy: $E = (1 - \lambda)D + \lambda V$ where D is the number of contiguous pixel-pairs in the entire image that have different labels, λ is a number between $(0, 1)$ that provides the relative weights of the two criteria and V is the normalized cluster variance given by:

$$V = \frac{\sum_{xy} \parallel T_{xy} - \mu_{S_{xy}} \parallel^2}{\sum_{xy} \parallel T_{xy} - \overline{T_{xy}} \parallel^2} \tag{1}$$

T_{xy} is the texture vector at pixel (x, y), S_{xy} the label at the pixel and μ_i the cluster mean of the ith label. Thus, $\mu_{S_{xy}}$ is the mean of all the texture vectors that belong to the same cluster as the pixel (x, y).

Instead of using global cluster compactness as a measure of distance, we used a local distance measure and incorporated a spatial contiguity term to form the update rule (See Equation 2). When the global cluster compactness is replaced by a local measure, it becomes possible to agglomeratively cluster within an image, to achieve a multiscale segmentation consisting of nested partitions.

II. Method

Clustering algorithms try to find structure in data, to find a convenient and valid organization of the data. A cluster is commonly understood to mean a set of entities that are alike, such that entities from different clusters are not alike [45], [46].

A hierarchical clustering method is a procedure for transforming an affinity matrix into a sequence of partitions that are nested [45]. In clustering, a set of objects is partitioned based on how similar the objects are to one another (given by the affinity matrix). Hierarchical clustering is a multi-step process where the set of partitions from the previous step is used to form the partitions at the current step. The end result is a sequence of partitions such that each partition is nested into the next partition in the sequence. This can be done in two ways: (a) *agglomerative* where the clustering algorithm at each stage merges two or more trivial clusters, thus nesting the earlier partition into a smaller number of clusters and (b) *divisive* where the

clustering algorithm at each stage divides up the current clusters into smaller clusters. The result of clustering is often represented by a dendrogram where layers of nodes each represent a cluster. The method proposed in this paper is an agglomerative hierarchical clustering algorithm.

A. Hierarchical K-Means Clustering of Texture Vectors

Images are segmented using an iterative texture segmentation method that yields a hierarchical representation of the regions at different scales. We drew on existing segmentation approaches that used K-means clustering on texture vectors [40], [41], [44], [42] but our use of inter-cluster distances leads naturally to a hierarchically arranged tree of identified regions. We follow the K-means clustering stage with a region growing step where connected pixels belonging to the same cluster are labeled the same. This ensures that we need not know the number of textures before hand, since K is only an intermediate step to finding the number of regions in the image. A final step uses the Euclidean distance between the mean texture vectors of statistically unsound regions and their statistically sound neighbors to yield a robust segmentation at the finest level of detail.

Having obtained the most detailed segmentation, we use the available inter-cluster distance measurements between every pair of adjoining regions in a dyadic manner to successively merge regions. Thus, the merge is done based purely on texture characteristics.

A vector of measurements taken in the neighborhood of a pixel is associated with that pixel. There is little consensus on which set of measurements are optimal, or what the standards for the choice of measurements should be. Descriptors based on statistical, structural and spectral properties of images have been utilized to form sets of discriminant features. Suggested descriptors include neighborhood statistics [47], [36], hidden Markov models [5], Markov Random Fields [6], image moments [7], coöccurrence and correlation matrices [8] and filtering methods [9]. Since there exists no consensus of which of these approaches provides the optimal texture vector, iterative feature extraction algorithms [48] have been devised to choose the basis and to reduce the dimensionality of the texture vectors. The suggested approach, then, is to include as many characteristics as possible and to utilize a dimensionality reduction algorithm to choose the characteristics that will actually

be used in the segmentation.

Since we are utilizing the texture vectors to separate the various clusters, we could utilize only those components that simultaneously maximize inter-cluster distance and minimize the within-cluster distance. Following [49], we can define the inter-cluster distance as $d = \sum_{i=1}^{K} \frac{N_i}{N} \parallel \mu_i - \mu \parallel$ and the within-cluster distance of the i^{th} cluster as: $d_i = \sum_{j=1}^{N_i} \parallel \mu_i - T_j \parallel$ and define a positive cost function that combines d and d_i, for example, $J = d/(\sum_{i=1}^{K} \frac{N_i}{N} d_i)$ where N_i is the number of pixels in the i^{th} cluster, and μ_i is the mean texture vector of that cluster. The value T_j is the texture value at the j^{th} pixel. The values N and μ represent the total number of pixels and the mean texture value over the entire image. Among the choices possible for the texture vectors, we should choose the texture vector set that has the highest J.

Texture is defined, not at the pixel level, but over the neighborhood of a pixel. The choice of the size and shape of the neighborhood is dictated by the properties of the texture under question (isotropicity, periodicity, variance, etc.). Within an image, there may be a variety of textures with different properties. One common solution to this problem is to use kernels of different sizes, shapes and orientations to compute the texture statistics in all the neighborhoods, thus encompassing a variety of textures [50]. In Section IV, results are demonstrated with texture vectors computed within a fixed neighborhood, a $7x7$ one, of the pixel.

Fisher's multiple linear discriminant functions [51] are the optimal solution, under certain constraints, to the problem of choosing the best set of texture vectors and the right size of neighborhood to compute the texture vectors. However, this requires knowledge of the means of the different image regions and a pooled covariance matrix. An iterative technique that reduces the feature set based on matrices computed at different resolutions (neighborhood sizes) can provide a reasonable approximation to the Fisher optima [52].

Thus, the optimal choice of texture vectors and neighborhood sizes is image dependent. The right choice should be made for the application by computing a variety of statistics at various neighborhood sizes. Then, using either the within- and between-cluster statistics of [49] or the iterative techniques of [48], [52], a subset of those features should be chosen.

In this work, we describe an untrained segmentation technique that can be used regardless of which texture vector is optimal. We used a single set of texture vectors for all the results discussed in this paper. With a

set of texture vectors that is tuned to the image being segmented (rather than the general purpose set used here), results may be improved.

B. K-Means Clustering Step

Using the texture vectors associated with every pixel in the image, the images were requantized to a fixed number of levels using K-Means clustering. It should be emphasized that this fixed number of levels ("K" in the K-means clustering) is not the number of regions in the resulting segmentation. It is the number of levels into which the image is initially requantized. The requantization is an iterative process that makes use of K-Means clustering to partition the image values into the K bins.

The measurement space (the gray level of the images) was divided up into K equal intervals and each pixel was initially assigned to the interval in which its gray level value lay. We used a cost function that imposes a Markov property on the pixels – that pixels tend to belong to the same cluster as their immediate neighbors but not on pixels further away. In each iteration, the best label for each pixel in the image was chosen based on a cost factor that incorporated two measures. The first measure is the Euclidean distance, $d_m(k)$, between the texture vector at that pixel and the cluster mean of the candidate k, given by: $d_m(k) =\parallel \mu_k^n - T_{xy} \parallel$ where μ_k^n is the cluster mean of the k^{th} cluster at the n^{th} iteration and T_{xy} the texture vector at the pixel (x, y). The second measure is a contiguity measure, $d_c(k)$, that measures the number of neighbors whose labels differed from the candidate label k. We can formally express the distance $d_c(k)$ as: $d_c(k) = \sum_{ij \epsilon N_{xy}}(1 - \delta(S_{ij}^n - k))$ where S_{ij}^n is the label of the pixel (i, j) at the n^{th} iteration and N_{xy} is the set of 8-neighbors of the pixel (x, y). Then the choice of the label for the pixel (x, y) in the $(n + 1)^{th}$ iteration, S_{xy}^{n+1}, is given by the label $k \epsilon S_{N_x y}^n$ for which the energy, $E(k)$. given by

$$E(k) = \lambda d_m(k) + (1 - \lambda) d_c(k) \qquad 0 \leq \lambda \leq 1 \tag{2}$$

is minimum. We used $\lambda = 0.6$ for all the images, finding that any value of λ between 0.2 and 0.8 gave similar results. At λ less than 0.2, i.e. where contiguity was over-emphasized, the result was that several texturally different, but contiguous, regions were nevertheless segmented as a single region. The candidates that were

considered were the labels at the n^{th} iteration of the pixels within the 8-neighborhood of (x, y). At the end of each iteration, the cluster attributes (the μ_k's) were updated based on all the pixels that were labeled as belonging to the cluster at that time.

C. Hierarchical segmentation step

At this point, the image has been requantized, but the quantization has taken the spatial arrangement of pixel values into account. A region growing algorithm is employed to build a set of connected regions, where each region consists of 8-connected pixels that belong to the same K-Means cluster. If a connected region is too small, then its mean texture (the mean of the texture vectors at each pixel in the region) is compared to the means of the adjoining regions and the small region is merged with the closest mean. This process is repeated until the regions are such that all their means have reliable statistics. In practice, we considered a region too small if it had less than 10 contributing (and independent) textural measurements. When segmenting a single image, the 10 contributing measurements will come from 10 pixels, but if you have two data fields corresponding to the same domain (visible and IR channels in satellite imagery, for example), then the 10 measurements can come from only 5 pixels). For comparison purposes, the results in this paper were all derived using only one image.

The result of the K-Means segmentation, region growing and region merge steps is the most detailed segmentation of the image. From this point onwards, we work exclusively in the domain of the segmented regions. The inter-cluster distances of all adjacent clusters (or regions) in the image are computed. A threshold is set such that half the pairs fall below this threshold. An iterative region merging is carried out whereby if a pair of clusters differ by less than this threshold, they are merged. More or less than half the clusters in the image may get merged because the region means are updated at the end of each merge, resulting in a different number of pairs which are closer than the threshold. The region merges are stopped when none of the resulting pairs of adjoining regions are closer than the threshold. The segmentation result at this point is the next coarser segmentation.

Because the results of segmentation at the second stage are formed by region merges only, every region in

the coarse segmentation completely contains one or more regions in the detailed segmentation. Thus, there is a hierarchy of containment between the segmented results at these two scales. The inter-cluster distance threshold is relaxed steadily, set at each iteration to be of a value such that half the cluster pairs are closer to each other than the threshold. This process is repeated until the segmented results are stable. The result of the segmentation at each stage gives one level of the hierarchical tree.

III. Comparison to Similar Work

In this paper, we advocate a novel method of performing multiscale segmentation which provides nested partitions even on textured images. The traditional interpretation of multiscale segmentation is in terms of multiscale image inputs. In our interpretation, the multiscale nature is in terms of the outputs (nested partitions). This interpretation of multiscale segmentation, introduced in [53] in terms of explicit spatial interrelationships, has found wide applicability – in image enhancement [54], for robust segmentation in image sequences [34] and in image registration [55].

Our argument is buttressed by the needs of the techniques that often follow the segmentation stage. Since the input to these techniques is the set of components at the various scales, the proposed approach can provide a better quality input. For example, in the tracking problem [56], it is necessary to match segmented components across frames. This matching is, by the nature of the problem, subject to association uncertainties. In a multiscale tracking problem, the matching of segmented components will have to match not only across frames but across scale as well. By ensuring that there is no leeway in the association of components across scale, we reduce the dimensionality of the association problem in multiscale tracking.

The methodology followed by [53], however, requires a mapping of the image gray level values, in essence specifying the "texture" vector to be used. In the method we propose, the choice of texture vector can be made depending on the types of images being segmented.

A considerable body of work in clustering [57], [58], [59], [60] deals with partitioning clustering techniques. Partitioning techniques are dynamic, in that they allow pixels to move from cluster to another at different stages. While the use of partitioning techniques, especially in combination with fuzzy clustering [61], can

improve cluster validity, partitioning clustering algorithms do not satisfy the requirement of nested partitions. Nested partitions are essential to the formation of a hierarchical tree representation of the segmented regions.

A recent paper [62] tries to reconcile the relative advantages of hierarchical and partitioning clustering by determining the "optimum" number of clusters via competitive agglomeration [63]. The method in this paper, like that of [62], uses agglomerative hierarchical partitioning rather than a divisive method. In the method proposed in this paper, global cluster information is incorporated via the K cluster means in the image, subject to Markovian penalties. In [62], global information is incorporated via an *a priori* assumption of shape, for example that all clusters are ellipsoidal. An *a priori* knowledge of cluster shape would be impossible in weather images, one important application of the segmentation method proposed in this paper.

The method of segmentation – K-means clustering of textural feature vectors – is not new. We drew on existing segmentation approaches that used K-means clustering on texture vectors. As in [40], a combination of K-means clustering and morphological operators was used here. As in [41], we iteratively estimate the local means of each region. As in [44], we use a contiguity-enhanced measure and as in [42], we model the cluster regions as random fields.

IV. Segmentation accuracy

Texture classification algorithms can be evaluated and compared quantitatively. Typically [64], [65], various texture representations – Markov Random Field, Gabor filters, fractal dimension measures, logical operators, coöccurrence matrices or statistics – are used to compute a vector of measurements for every pixel. Using the computed textures over a number of textured images, a model of the input classes is formed. This model is then used in tandem with a texture classification technique such as a K-Nearest neighbor classifier or with a Gauss-Markov classifier to classify the texture vectors found in a set of test images. By varying either the texture representation or the texture classification method, the percentage of pixels that are correctly classified may be compared with other known techniques.

Thus, frameworks such as [66] may be used to evaluate texture representations or texture classification algorithms. However, they cannot be used to evaluate texture segmentation algorithms. In the literature, for

example [65], this shortcoming is avoided by casting the segmentation problem into a classification one by requiring that a suitably trained classifier be used for segmentation. Since we are proposing a new technique of doing image segmentation, a metric geared toward only segmentation needs to be utilized.

The Rand [67] index may be used to compare the partitions yielded by a segmentation algorithm with a true partioning of the image. When pairs of pixels are considered in turn, the Rand index is the fraction of pairs correctly placed. If two pixels belong to the same cluster in the truth, then if they belong to the same cluster in the cluster output, the clustering algorithm has succeeded. Similarly, if the pixels belong to different clusters in the truth, a successful clustering algorithm should put them into different bins. Thus, the Rand index computes clustering efficiency by treating the clustering problem as a binary classification problem for every pair of pixels in the image. This pairwise comparison does not incorporate contiguity. The Rand index, therefore, does not incorporate the notion of a "region" which is important in segmentation problems.

We propose a metric for image segmentation that is computed similar to the way metrics for texture classification are computed. A test image, comprised of different textural swatches, is presented to the segmentation algorithm. At the time of presentation, a labeled image is created with labels $1, 2..., K$ depending on which texture was used for that point in the test image. The segmentation algorithm then performs the segmentation and delivers its output labeled $1, 2, ..M$. If the segmentation algorithm requires prior knowledge of the number of regions, then the value of K is passed into the algorithm (in which case, the resulting M would be equal to K). At this point, we have the correctly labeled image (the "true" segmentation) and the segmented image. We can then measure the performance of the segmentation algorithm.

The segmentation accuracy of the i^{th} object in the scene can be computed as: $a_i = (S_{i1} \vee S_{i2} \vee \ldots \vee S_{iM})/N_i$ where S_{ij} is the number of pixels in the region of the i^{th} object that is occupied by the j^{th} segmented region and $\vee$ is the max operator: $a \vee b = \max(a, b)$ Therefore, $\bigvee_j S_{ij}$ gives the segmented region that covers the largest fraction of the i^{th} object. The fraction of the i^{th} object occupied by this segmented region is also a measure of the accuracy with which the segmentation method has segmented that object.

We could measure the overall segmentation accuracy as a fuzzy measure given by the minimum accuracy with which individual objects, including the "background object", have been identified, i.e. by: $a = \bigwedge_i^K \frac{\bigvee_j S_{ij}}{N_i}$

where $\wedge$ is the min operator: $a \wedge b = \min(a, b)$ However, this measure has a shortcoming in that it does not enforce the separate identification of different regions. This can be enforced by computing the accuracy as:

$$a = \frac{\bigvee_j S_{i_1 j}}{N_{i_1}} \wedge \frac{\bigvee_{j \epsilon E_1} S_{i_2 j}}{N_{i_2}} \wedge \ldots \frac{\bigvee_{j \epsilon E_{k-1}} S_{i_k j}}{N_{i_k}} \tag{3}$$

where the regions from the true segmentation (the i's) are selected such that $N_{i_p} > N_{i_{p+1}}$, i.e. in decreasing order of size. The choice of the regions from the result of the segmentation technique (the j's) is restricted to the set E_p, the set of $j's$ that have not been selected as matching any of the previous p regions.

The minimum and maximum operators used here in tandem form a compensatory fuzzy aggregation [68]. The minimum and maximum operators provide the most conservative measures [69], [68]. Other fuzzy aggregation operators, such as the arithmetic or geometric means [70] or an ordered weighted average [71] may also be used if less extreme characterizations are desired.

A. Brodatz Textures

Following [64], [65], natural textures were chosen from the Brodatz [72] database and arranged within an image in a known manner. Some of the images created from the Brodatz textures and the best resulting segmentations of those images are shown in Figure 1. The hierarchical, agglomerative, K-Means technique described in this paper was compared against two other texture segmentation algorithms reported in the literature – a contiguity-enhanced K-Means method [44] and with a Kolmogorov-Smirnov test-based method [37]. These techniques were chosen because they are completely unsupervised and because they, like the method introduced in this paper, require no training on the textures in the image. For the hierarchical method of this paper, the second most detailed scale of segmentation was chosen for all the measurements – this yields the benefits of both the K-Means requantization and the inter-cluster distance merging steps.

To make meaningful comparisons possible, the same set of statistical measurements (mean, variance, homogeneity, kurtosis, and contrast) were computed in the same neighborhood (7x7) about a pixel for all of the segmentation algorithms. Kurtosis at a pixel (x, y) is computed within the neighborhood of the pixel, $N_x y$,

using:

$$kurt_{xy} = \frac{\sum_{i \epsilon N_{xy}} (\frac{I_{xy} - I_i}{s_{xy}})^4}{card(N_{xy})} \tag{4}$$

where I_{xy} is the pixel value at (x, y), s_{xy} the standard deviation in that neighborhood and $card$ is the cardinality of the neighborhood. Homogeneity is computed as:

$$hom_{xy} = \frac{\sum_{i \epsilon N_{xy}} \frac{1}{1+(\frac{I_{xy} - I_i}{\overline{I_{xy}}})^2}}{card(N_{xy}) - 1} \tag{5}$$

while contrast is computed as:

$$contrast_{xy} = \frac{\sum_{i \epsilon N_{xy}} (\frac{I_{xy} - I_i}{\overline{I_{xy}}})^2}{card(N_{xy})} \tag{6}$$

Since the traditional K-Means technique requires prior knowledge of the actual number of regions in the test image, this was input to the algorithm. We also show the result of over-estimating the number of regions. We demonstrate the effect of K, the number of significant quantization levels in our technique by showing the results when K is chosen in a range from 2 to 16. The method of [37] was implemented with reasonable defaults and the best performing initialization [36] and the same set of parameters used for the entire data-set. The accuracy of segmentation is shown in Table I. For easy reference, the two best performances are shown in bold.

As expected, textures that are insufficiently captured by the texture representation are segmented poorly. When the texture is well represented, as in Figure 1c, the resulting segmentation is very good. It should be noted that the performance of all these segmentation algorithms is without the benefit of texture training – these are unsupervised texture segmentation algorithms that have not seen these objects before.

The result of segmenting the second test pattern using each of these techniques is shown in Figure 2. The method introduced here is a hierarchical method, and using only one of the resulting scales actually underestimates the performance of the technique. The different scale segmentations that result when segmenting the fourth test case (D112 and D19) is shown in Figure 3.

B. Real-world images

The aerial photograph of San Francisco [73] has been segmented and the results shown in Figure 4.

This is a particularly hard image to segment because there is very high variability within the regions of the image – note the presence of large black strips within the land portion of the image – and because there is very little difference between what are truly different parts of the image. For example, the glare on the water makes that parts of the sea resemble the land surface more than the darker water areas. Thus, it is not surprising that the results of segmentation using the Markov Random Field-based texture segmentation approach of [37] are not very good (See Figure 4e).

Statistical texture segmentation methods fare better in the comparison. In the original study [73], segmentation was performed by using statistical tests as a measure of homogeneity and formulating texture segmentation as a data clustering problem, with inter-cluster differences defined by a multi-scale Gabor filter image representation. Clustering was done by simulated annealing, with the number of clusters assumed to be known *a priori*. Unlike the study [73] from which this image was taken, we obtained these results without any *a priori* assumption of the number of regions in the image.

The edge-flow method of [74], like the original study, is Gabor-filter based. The result of segmentation using the edge flow method is shown in Figure 4f. Except for the water areas, the segmentation performs quite well.

It is noteworthy that we obtain a segmentation of quality comparable to the Gabor-filter methods without any *a priori* assumptions about the number of clusters in the image. Other images, including photographs of buildings and ulcerated leg wounds, are segmented using the technique described here and the results discussed in [56].

A single infrared satellite weather image was segmented using various unsupervised segmentation methods. The results are shown in Figure 5. The results of segmentation using the other approaches are poor. This is not surprising because the infrared satellite weather imagery has several characteristics that make it hard to segment: very low dynamic range (from about 225K to 240K) for the regions of interest, poor resolution as compared to the scale of the phenomena of interest, and high pixel value variance, even in the absence of

edges.

To give an idea of the scale of features we are interested in, and why we deem the segmentation result of the other techniques not detailed enough, we show a satellite image with more contextual information in Figure 6. The data from a from a ground-based radar in Oklahoma City, OK, USA at approximately the same time is shown superimposed on the infrared image. The infrared image, as can be seen from the background maps, covers the entire central United States.

V. Conclusion

When measured over synthetic images comprised of strips of Brodatz textures, as shown in Table I, the method of this paper yields a more accurate segmentation than the other techniques considered. Qualitatively, the method of this paper produces segmentation results that are much better than existing techniques when applied to real-world images, whether they are building photographs, terrain shots or medical images.

A. Contributions of paper

A novel method of performing multiscale segmentation of images using texture properties has been introduced. Various methods of texture segmentation, were compared with the K-Means clustering of texture vectors used in this paper.

An objective way to measure the accuracy of texture segmentation algorithms independent of texture classification was developed and used to compare different texture segmentation algorithms.

Multiscale segmentation in the context of the segmented regions themselves was introduced. This new approach to multiscale segmentation was incorporated into the K-Means clustering technique as a steady relaxation of inter-cluster distances.

Our multiscale segmentation approach was demonstrated on several real-world images. It was shown that quality of the segmented results at the different scales was significantly better, in terms of accuracy, than existing approaches.

References

[1] R.M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, no. 5, pp. 786–804, May 1979.

[2] G. Smith, *Image Texture Analysis using Zero Crossings Information*, Ph.D. thesis, U. of Queensland, Brisbane, 1998.

[3] A. Bovik, M. Clarke, and W. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 12, pp. 55–73, 1990.

[4] A. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[5] X. Huang, Y. Akiri, and M. Jack, *Hidden Markov Models for Speech Recognition*, Edinburgh Univ. Press, Edinburgh, 1990.

[6] R. Chellappa, "Two dimensional discrete gaussian markov random field models for image processing," in *Progress in Pattern Recognition*, L. Kanal and A. Rosenfeld, Eds., pp. 79–112. North Holland, Amsterdam, 1985.

[7] C. Nikias, "Higher order spectral analysis," in *Advances in Spectrum Analysis and Array Processing*, S. Haykin, Ed., pp. 326–365. Prentice Hall, Englewood Cliffs, NJ, 1991.

[8] P. Chen and T. Pavlidis, "Segmentation by texture using correlation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 5, no. 7, pp. 64–69, Jan. 1983.

[9] A.K. Jain and F. Farroknia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[10] D. Dunn and W. E. Higgins, "Optimal Gabor filters for texture segmentation," *IEEE Trans. Image Proc.*, vol. 4, no. 7, pp. 947–964, July 1995.

[11] T. P. Weldon and W. E. Higgins, "An algorithm for designing multiple Gabor filters for segmenting multi-textured images," in *Proc. IEEE Int'l. Conf. Image Proc.*, Chicago, IL, October 4-7 1998.

[12] J. Havlicek, "The evolution of modern texture processing," *Elektrik*, vol. 5, no. 1, pp. 1–28, 1997.

[13] B. Julesz, "Experiments in the visual perception of texture," *Sci. Amer.*, vol. 232, pp. 84–92, 1975.

[14] D. Gabor, "Theory of communication," *J. Inst. Elec. Eng. London*, vol. 93, no. III, pp. 429–457, 1946.

[15] S. Marcelja, "Mathematical description of the responses of simple cortical cells," *J. Opt. Soc. Am.*, vol. 70, no. 11, pp. 1297–1300, 1982.

[16] A. Jain and K. Karu, "Learning texture discrimination masks," *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 18, pp. 195–205, 1996.

[17] A.R. Rao and B.G. Schunck, "Computing oriented texture fields," *Proc. IEEE Comput. Soc. Conf. Comput. Vision Patt. Recog.*, pp. 61–68, Jun 1989.

[18] M.R. Turner, "Texture discrimination by Gabor functions," *Biol. Cybern.*, vol. 55, no. 2, pp. 71–82, 1986.

[19] Y. Zeevi and M. Porat, "Combined frequency-position scheme of image representation in vision," *J. Opt. Soc. Am. A*, vol. 1, no. 12, pp. 1248, Dec 1984.

[20] M. Porat and Y. Zeevi, "Localized texture processing in vision: Analysis and synthesis in the Gaborian space," *IEEE Trans. Biomed. Engg.*, vol. 36, no. 1, pp. 115–129, Jan. 1988.

[21] A. Bovik, "Analysis of multichannel narrow-band filters for image texture segmentation," *IEEE Trans. Signal Proc.*, vol. 39, no. 9, pp. 2025–2043, Sep. 1991.

[22] M. Kass and A. Witkin, "Analyzing oriented patterns," *Comput. Vision, Graphics, Image Proc.*, vol. 37, pp. 362–385, 1987.

[23] J. Havlicek, D. Harding, and A. Bovik, "The multi-component AM-FM image representation," *IEEE Trans. Image Processing*, vol. 5, no. 6, pp. 1094–1100, 06 1996.

[24] J.T. Johnson, P. Mackeen, A. Witt, E.D. Mitchell, G. Stumpf, M. Eilts, and K. Thomas, "The storm cell identification and tracking algorithm: An enhanced WSR-88D algorithm," *Weather and Forecasting*, vol. 13, no. 6, pp. 263–276, June 1998.

[25] K.A. Browning, C.G. Collier, P.R. Larke, P. Menmuir, G.A. Monk, and R.G. Owens, "On the forecasting of frontal rain using a weather radar network," *Monthly Weather Review*, vol. 110, pp. 534–552, 1982.

[26] A. Witt, M. Eilts, G. Stumpf, J.T. Johnson, E.D. Mitchell, and K.W. Thomas, "An enhanced hail detection algorithm for the WSR-88D," *Weather and Forecasting*, vol. 13, pp. 286–303, 1998.

[27] E. Lai, P. Li, C. Chan, M. Chu, and W. Wong, "Pattern recognition of radar echoes for short-range rainfall forecast," in *15th Intl. Conf. on Pattern Recog.* IEEE, 2000, vol. 4, pp. 299–302.

[28] J. Lee, R.C. Weger, S.K. Sengupta, and R.M. Welch, "A neural network approach to cloud classification," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 28, no. 5, pp. 846–855, Sep. 1990.

[29] Moving Pictures Experts Group, "Coding of moving pictures and audio," Tech. Rep. ISO/IEC JTCI/SC29/WG11, International Organization for Standardisation, 1997.

[30] Jinsang Kim and Tom Chen, "A VLSI architecture for image sequence segmentation using edge fusion," in *International Workshop on Computer Architectures for Machine Perce*, Padova, Italy, Sep. 2000.

[31] M.M. Wolfson, B.E. Forman, R.G. Hallowell, and M.P. Moore, "The growth and decay storm tracker," in *8th Conference on Aviation*, Dallas, TX, 1999, Amer. Meteor. Soc., pp. 58–62.

[32] V. Lakshmanan, R. Rabin, and V. DeBrunner, "Identifying and tracking storms in satellite images," in *Second Artificial Intelligence Conference*, Long Beach, CA, 2000, American Meteorological Society, pp. 90–95.

[33] J. Peak and P. Tag, "Segmentation of satellite weather imagery using hierarchical thresholding and neural networks," *Journal of Applied Meteorology*, vol. 33, no. 5, pp. 605–616, 1994.

[34] M. Yang and N. Ahuja, "Recognizing hand gestures using motion trajectories," in *IEEE Conf. on Comp. Vision and Patt. Recog.*, Jun 1999, vol. 1, pp. 466–472.

[35] H. Shi, "Cloud movement detection for satellite images," in *4th Intl Conf. on Signal Proc.* IEEE, 1998, vol. 2, pp. 982–985.

[36] V. Lakshmanan, V. DeBrunner, and R. Rabin, "Texture-based segmentation of satellite weather imagery," in *Int'l Conference on Image Processing*, Vancouver, Sept. 2000, pp. 732–735.

[37] J. Blum and J. Rosenblat, *Probability and Statistics*, W.B. Saunders Company, 1972.

[38] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *Int'l J. Comp.Vis.*, vol. 12, no. 1, pp. 43–77, 1994.

[39] T. Corpetti, E. Memin, and P. Perez, "Dense estimation of fluid flows," *IEEE Trans. on Patt. Anal. and Mach. Intell.*, vol. 24, no. 3, pp. 365–380, Mar. 2002.

[40] C. Chen, J. Luo, and K. Parker, "Image segmentation via adaptive k-mean clustering and knowledge-based morphological operations with biomedical applications," *IEEE Trans. on Image Processing*, vol. 7, no. 12, pp. 1673–1683, Dec 1998.

[41] T. Pappas, "An adaptive clustering algorithm for image segmentation," *IEEE Trans. Signal Processing*, vol. 40, pp. 901–914, 1992.

[42] H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using gibbs random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, pp. 39–55, 1987.

[43] K.C. Gowda, "A feature reduction and unsupervised classification algorithm for multispectral data," *Pattern Recognition*, vol. 17, pp. 667–676, 1984.

[44] J. Theiler and G. Gisler, "A contiguity-enhanced K-Means clustering algorithm for unsupervised multispectral image segmentation," in *Proc. SPIE*, 1997, vol. 3159, pp. 108–118.

[45] A. Jain and R. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[46] B. Everitt, *Cluster Analysis*, John Wiley & Sons, New York, 1974.

[47] A. Solberg and A. Jain, "Texture fusion and feature selection applied to SAR imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 2, pp. 475–479, 3 1997.

[48] K. Etemad and R. Chellappa, "Signal and image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 7, pp. 1453–1465, Oct 1998.

[49] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic, New York, 2 edition, 1990.

[50] P. Burt, "Fast algorithms for estimating local image properties," *Computer Graph. Image Processing*, vol. 21, pp. 368–382, 1983.

[51] R. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, pp. 179–188, 1936.

[52] M. Unser and M. Eden, "Multiresolution feature extraction and selection for texture segmentation," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 11, no. 7, pp. 717–728, 1989.

[53] N. Ahuja, "A transform for multiscale image segmentation by integrated edge and region detection," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 18, no. 12, pp. 1211–1235, 1996.

[54] J. Eledath, J. Ghosh, and S.V. Chaktravarthy, "Image enhancement using scale-based clustering properties of the radial basis function network," Tech. Rep. UT-CVIS-TR-97-006, Center for Vision and Image Sciences, U. Texas, Austin, 1997.

[55] Z.P. Liang, H. Pan, Y. Hui, R. Magin, N. Ahuja, and T.S. Huang, "Automated image registration by maximization of a region similarity metric," *Int. J. Imaging Sys. and Tech.*, , no. 6, pp. 513–518, 1997.

[56] V. Lakshmanan, *A Heirarchical, Multiscale Texture Segmentation Algorithm for Real-World Scenes*, Ph.D. thesis, U. Oklahoma, Norman, OK, 2001.

[57] L. Bobrowski and J. Bezdek, "C-means clustering with the l_1 and l_∞ norms," *IEEE Trans. Systems, Man and Cybernetics*, vol. 28, no. 3, pp. 545–554, 1991.

[58] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space defined by implicit equations with application to edge and range segmentation," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 13, no. 11, pp. 1115–1138, Nov 1991.

[59] L. Kaufman and P. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley Interscience, New York, 1990.

[60] Y. Ohashi, "Fuzzy clustering and robust estimation," in *Ninth Meeting of SAS Users Group Int'l*, Hollywood Beach, Fla., 1984.

[61] R. Dave and K. Bhaswan, "Adaptive fuzzy c-shells clustering and detection of ellipses," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 643–662, May 1992.

[62] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 21, no. 5, pp. 450–465, May 1999.

[63] H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration," *Pattern Recognition*, vol. 30, no. 7, pp. 1223–1232, 1997.

[64] P.P. Ohanian and R.C. Dubes, "Performance Evaluation for Four Classes of Textural Features," *Pattern Recognition*, vol. 25, pp. 819–833, 1992.

[65] V. Manian, R. Vasquez, and P. Katiyar, "Texture classification using logical operators," *IEEE Trans. on Image Proc.*, 2000.

[66] G. Smith, "Meastex image texture database and test suite," http://www.cssip.uq.edu.au/staff/meastex/meastex.html, May 1997.

[67] W.M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the Amer. Stat. Assoc.*, vol. 66, pp. 846–850, 1971.

[68] H. Zimmerman and P. Zysno, "Latent connectives in human decision making," *Fuzzy Sets and Systems*, vol. 4, pp. 37–51, 1980.

[69] W. Pedrycz and F. Gomide, *An Introduction to Fuzzy Sets: Analysis and Design*, MIT Press, 1998.

[70] G. Klir and T. Folger, *Fuzzy Sets, Uncertainty and Information*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

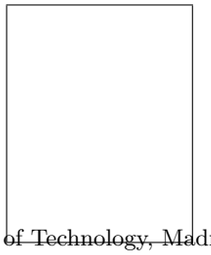
[71] R. Yager, "On ordered weighted averaging aggregation operations in multicriteria decision making," *IEEE Trans. on Sys., Man, and Cyber.*, vol. 18, pp. 183–190, 1988.

[72] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover Publishing Co., Toronto, 1966.

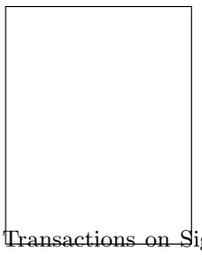
[73] T. Hofmann, J. Puzicha, and J. Buhmann, "A deterministic annealing framework for unsupervised texture segmentation," Tech. Rep. IAI-TR-96-2, Institut fr Informatik III, U. Bonn, 1996, http://www-dbv.cs.uni-bonn.de/image/example4.html.

[74] W.Y. Ma and B.S. Manjunath, "Edge flow: a framework of boundary detection and image segmentation," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997, pp. 744–749.

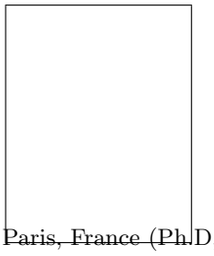
[75] L. Najman and M. Schmitt, "Geodesic saliency of watershed contours and hierarchical segmentation," *IEEE Trans. Patt. Anal. and Mach. Intell.*, vol. 18, pp. 1163–1173, 1996.



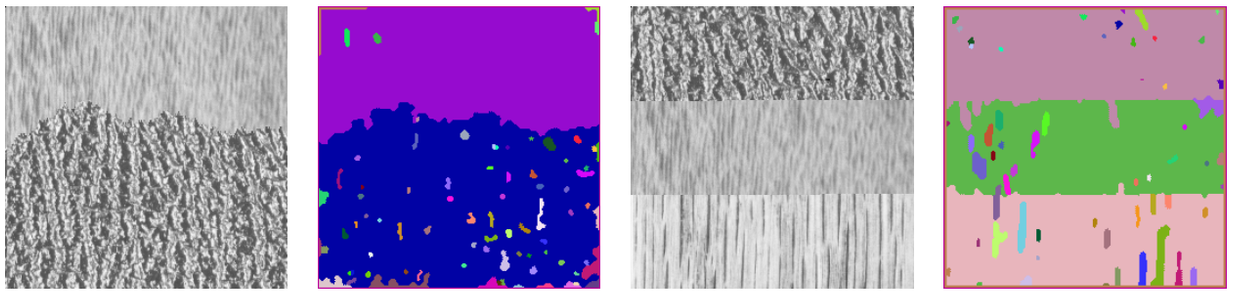
V Lakshmanan is with the National Severe Storms Laboratory, Norman, Oklahoma and the Coöperative Institute of Mesoscale Meteorological Studies at the University of Oklahoma where his work includes developing image processing and computer vision algorithms for weather phenomena. He received degrees from the University of Oklahoma (Ph.D, 2001), The Ohio State University (M.S., 1995) and the Indian Institute of Technology, Madras (B.Tech, 1993).



Victor E DeBrunner was born in Auburn, AL on August 21, 1962. He received the B.EE degree from Auburn University in 1984 and the M.S. and Ph.D. degrees in Electrical Engineering from Virginia Polytechnic Institute and State University in 1986 and 1990, respectively. He has been at the University of Oklahoma since 1990 where he is currently a Professor. Dr. DeBrunner has served as an associate editor for the IEEE Transactions on Signal Processing, is serving on the DSP Theory and Methods committee of the SP Society, and he currently serves on the steering committee for the Asilomar Conference on Signals, Systems, and Computers. His research interests include signal and image processing algorithms and implementations.



Robert M Rabin is a research meteorologist with the National Severe Storms Laboratory, Norman, Oklahoma and the Cooperative Institute for Meteorological Satellite Studies, Madison, WI. His work includes developing applications of satellite and radar data in weather analysis and forecasting. He received degrees from McGill University, Montreal, Canada (B.Sc, 1974, M.Sci, 1977) and University Pierre et Marie Curie in Paris, France (Ph.D, 1987).



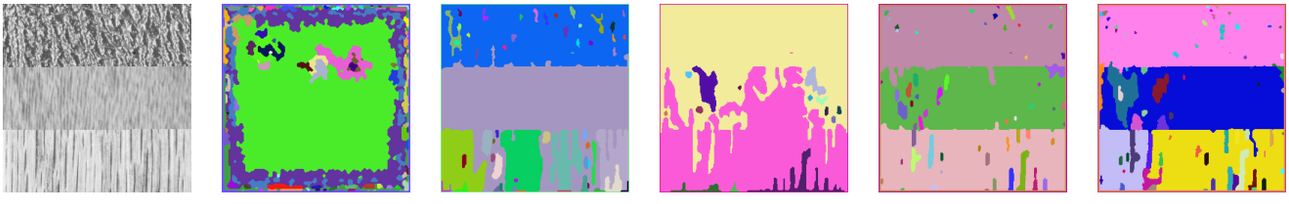
a b c d

Fig. 1. (a) Image comprised of Brodatz [72] swathes D38 and D24. (b) The result of segmentation by the method of this paper with K=4. Note that K is not the number of regions – there are actually 83 regions in the segmentation result shown. Most of these regions are agglomerated in later stages of the multiscale segmentation algorithm (See, for example, Figure 3). (c) D24, D38 and D68. (d) The result of segmentation by the method of this paper with K=8. The texture vector captures all three of these textures well, hence the excellent performance.

Test/Technique (Parameter)	KS Test [37], [36]	K-Means [44]		Hierarchical K-Means			
		K=2or3	K=4	K=2	K=4	K=8	K=16
D12 and D15	0.22	0.21	0.03	0.03	**0.37**	**0.42**	0.37
D24, D38, D68	0.05	0.18	0.08	0.06	0.02	**0.84**	**0.57**
D84, D94, D29	**0.32**	0.02	0.04	0.00	0.02	0.04	**0.13**
D112 and D19	0.05	0.03	0.33	0.02	**0.58**	**0.72**	0.53
D15, D112,D38	0.14	**0.39**	0.20	0.03	0.02	0.24	**0.28**
D38, D64	0.06	0.73	0.69	0.76	**0.89**	**0.88**	0.88
μ	0.14	0.26	0.228	0.149	0.316	**0.523**	**0.459**
σ/μ	0.722	0.938	1.013	1.822	1.051	**0.599**	**0.519**

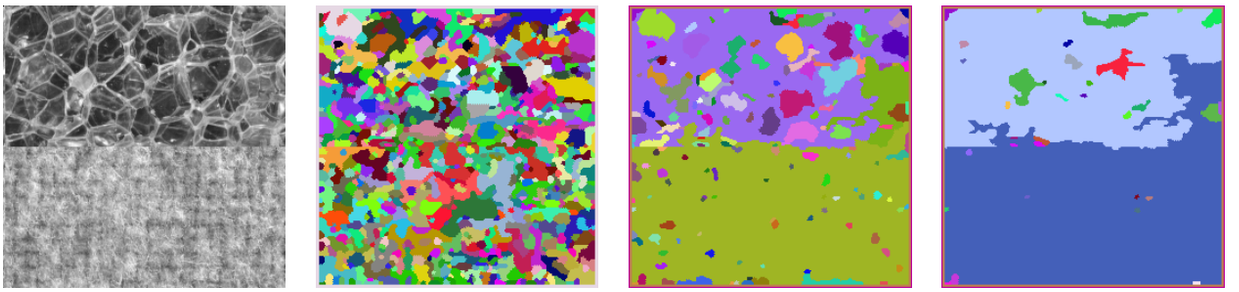
TABLE I

IMAGES COMPRISED OF BRODATZ [72] SWATHES (SEE FIGURE 1) WERE SEGMENTED USING VARIOUS UNSUPERVISED, UNTRAINED ALGORITHMS AND THE ACCURACY COMPUTED AS IN EQUATION 3. AS CAN BE SEEN, THE METHOD OF THIS PAPER IS THE BEST PERFORMING (MAXIMUM MEAN) AND THE MOST CONSISTENT (LEAST SIGMA/MEAN).



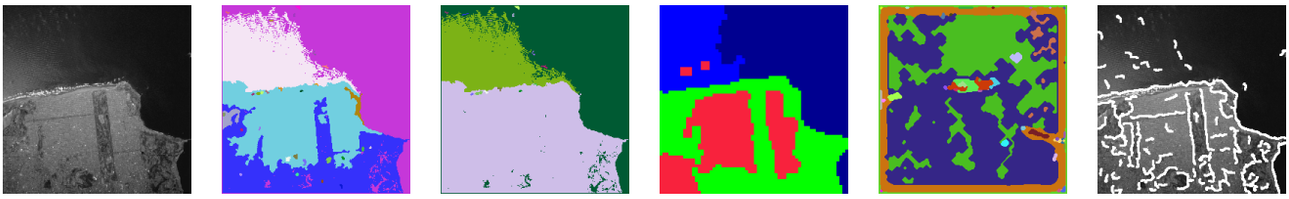
a b c d e f

Fig. 2. (a) Image comprised of Brodatz [72] textures D24, D38 and D68. The result of segmentation by (b) the method of [37]. (c) the method of [44] with K=3 (there are 3 clusters). (d) the method of this paper with K=2. (e) the method of this paper with K=8 (the best performance). (f) the method of this paper with K=16. This is the second test referred to in Table I.



a b c d

Fig. 3. (a) Image comprised of Brodatz [72] textures D112 and D19. The result of segmentation using the method of this paper with K=16 at various scales. (b) most detailed (accuracy=0.04) (c) second most detailed – this is what is used in the accuracy computation in Table I. accuracy=0.53. (d) coarse – this is actually the most accurate (accuracy=0.76). This is the fourth test referred to in Table I.



a b c d e f

Fig. 4. Left to right, top to bottom: (a) An aerial image of San Francisco. (b) The most detailed segmentation of the image using the method of this paper. (c) Segmentation at a coarser level using the method of this paper. There is no *a priori* assumption of the number of regions in the image in the method described in this paper. (d) The result of segmentation using the method of [73] (a Gabor filtering and clustering based-method, from which this image was taken) assuming that there are four clusters. (e) Final segmented result, using the MRF-based approach of [37]. (f) Edges found using the Gabor filter texture segmentation method of [74]

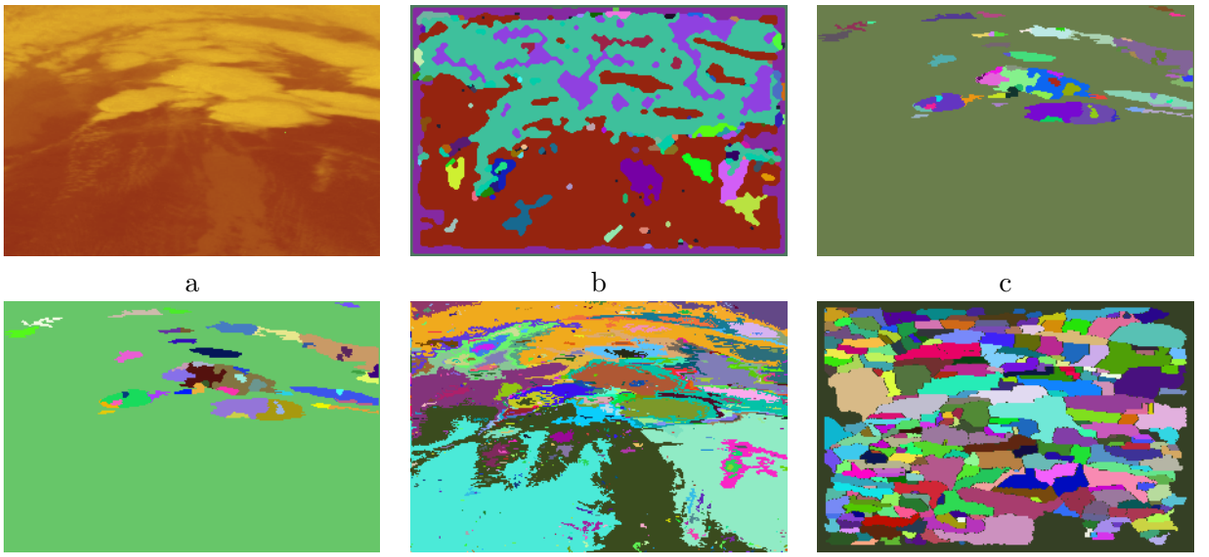


Fig. 5. Segmenting an infrared satellite weather image. (a) The infrared image being segmented. Notice the various storms at the top of the image. The darker areas in the bottom correspond to ground. (b) The result of segmenting the image using Markov Random Field (MRF) approach of [37]. There is no detail – the segmentation result has very large segments. (c) The result of segmenting the image using the method of this paper (the most detailed scale). Notice the fine detail within the clouds. (d) The next higher scale of segmentation using the method of this paper. The strong storm cells being significantly colder are retained – the large cloud masses are merged. (e) Thresholding the image into successively increasing $1Kelvin$ thresholds. This thresholding captures a lot of detail, but no organization. (f) Using the watershed segmentation approach of [75]. Because of the textural nature of the data, the watershed algorithm has very poor performance.

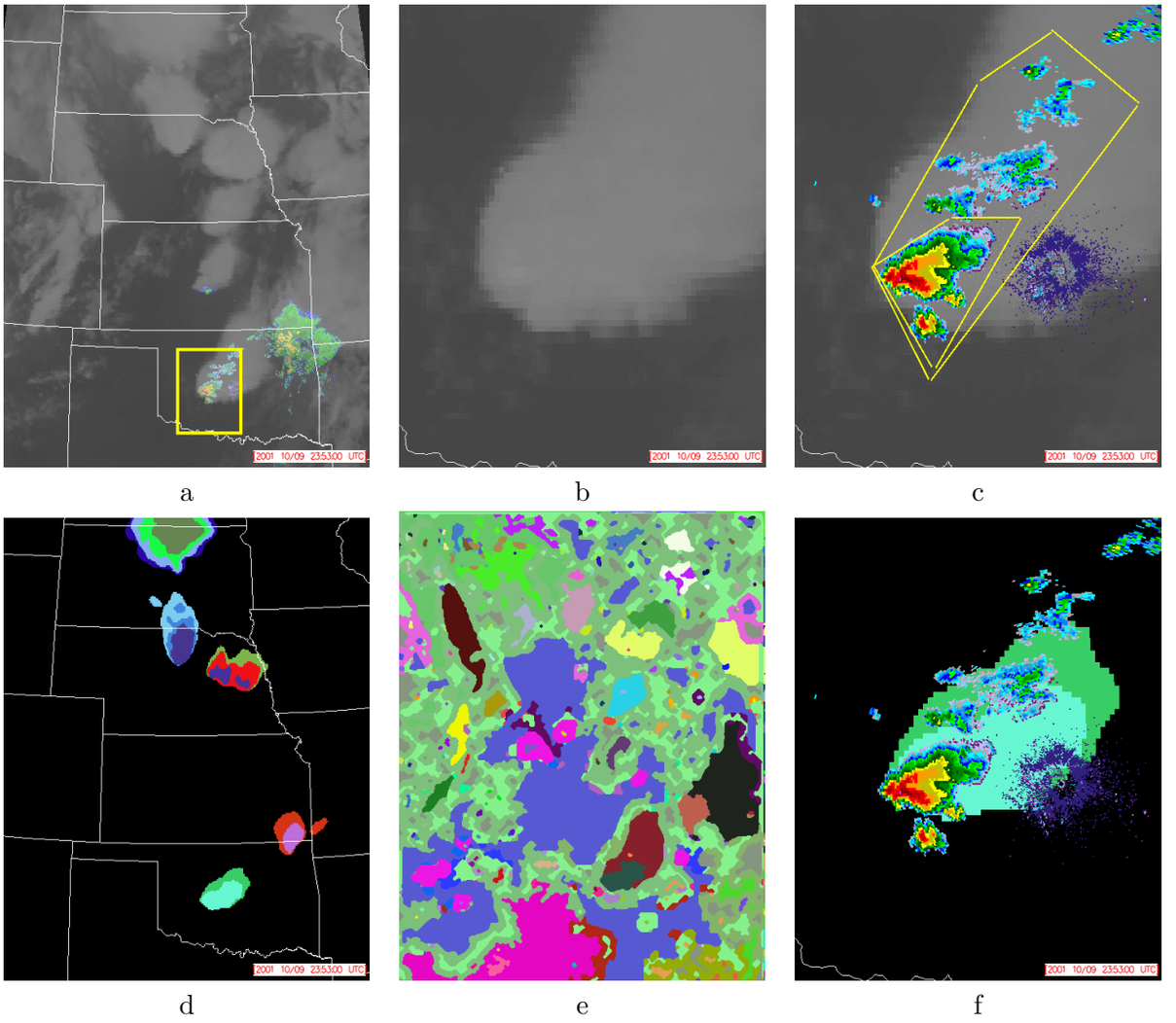


Fig. 6. Segmenting an infrared satellite weather image. (a) The infrared image being segmented. To add contextual information, a map background with the outlines of states in the Central United States has been added. The area that will be zoomed into for a closer look is shown in a rectangular area. (b) Showing in more detail an area of the infrared image for which there is corresponding radar data. (c) Human segmentation of the images based on using both the radar data and the satellite data. (d) The result of segmenting the image using the method of this paper (the most detailed scale) using only the infrared image. (e) The result of segmenting the image using Markov Random Field (MRF) approach of [37]. The required detail is not present in the segmentation. (f) Close-in look at the result in (d). Notice that the method has found the two significant regions, and placed the storms in the right regions. The exact location of the boundaries is different because the human was able to use the radar data to locate the extent of the two regions while the algorithm could use only the textural content of the infrared image.