The evaluation of segmentation and texture algorithm combinations for scene analysis

SAMEER SINGH, *Member* IEEE
MANEESHA SINGH, *Student Member* IEEE

Department of Computer Science
University of Exeter
Exeter EX4 4PT
United Kingdom

Tel: +44-1392-264053
Fax: +44-1392-264067
Email: {s.singh, m.singh}@ex.ac.uk

ABSTRACT

The optimisation of image processing tools individually in a chain of processes does not yield an optimal chain. If we are to consider different steps of image processing and pattern recognition within a scene analysis system as different components, the effect of one component on another should not be underestimated. However, most scene analysis systems for natural object recognition are developed without any consideration for component interactions. In this paper we demonstrate how the application of different image segmentation algorithms directly relates to the quality of texture measures extracted from segmented regions and directly impact on the classification ability. The difference between the best and the worst performances is found to be significant. We then develop the methodology for determining the optimal chain for scene analysis and show our experimental results on the publicly available benchmark "Minerva".

Index Terms: scene analysis, benchmark, object recognition, texture analysis, image segmentation

1. MOTIVATION

The object recognition process in image analysis can be considered as the output of a chain of processes, or components, including image acquisition, enhancement, segmentation, feature extraction, and classification. Each component can be viewed as a plug and play module for which several options are available. For a standardised system, the inputs and outputs from these processes are of a predefined format for a seamless operation. One of the traditional ways in which these components are selected is based on experience and recommendation of comparative studies. In this manner, the experimenter could plug in the best enhancement method, followed by the best segmentation method, and so on. In this paper we demonstrate that what is best cannot be considered in isolation, e.g. component interactions in an image analysis chain are of paramount importance for achieving good results. Hence, the traditional approach to tool selection is sub-optimal. The main contribution of our work is the specification of a methodology for estimating the best component combination that maximises classification ability by considering the combinations of two important image analysis operations, namely *image segmentation*, and *texture feature extraction*. This methodology improves the image classification ability of classifiers and allows the optimisation of image analysis tools, e.g. selection of segmentation method, on a per image basis. The novelty of our approach lies in the use of a new compactness measure for data characterisation, and a systematic approach to determining optimal segmentation/texture analysis chain for both a collection of images and also individual test image regions. Most of our discussion and techniques discussed are closely related to natural scene analysis application, however, the basic methodology is fairly generic to other image processing applications.

Given an unknown image X, with the objective of identifying objects in it, the image analysis task involves the following steps: image enhancement (improvement of image appearance), image segmentation (isolation of regions on the basis of homogeneity criteria), feature extraction (calculation of shape, texture or other statistical vectors that characterise the object), and classification (a classifier test labels an unseen object based on training data). In this paper we study how the region definitions, i.e. all regions in an image defined by their member pixels, affect the quality of features extracted to represent that object. Our initial conjecture, which we will subsequently prove in this paper, is that given *N* different segmentation algorithms, and *M* different feature extraction methods, on a given set of images *I*, the object recognition rates are very different depending on the combination chosen. In addition, if segmentation algorithms $\{S_1, S_2, \ldots S_N\}$ generate region definitions $\{R_1, R_2, \ldots R_N\}$ for the same object *Z* in a given image, then the application of a given feature extraction algorithm *T* on only one of these region definitions produces the best object definition, and in this manner defines the best segmentation algorithm for [*Z*,*T*] combination.

Performance evaluation of computer vision algorithms is a neglected yet very important area of work [28]. The understanding of which segmentation and feature extraction algorithms work best together will be naturally a major step forward in the field of image analysis, understanding, and computer vision. The major advance will lie in the fact that for different objects in images, we could determine the optimal image segmentation and feature extraction combination out of hundreds of possibilities. This work can be used for a knowledge-based system that optimises the image processing tool depending on the chosen task. Some success in the area of knowledge-based configuration of image processing algorithms has been achieved with systems such as CONNY[29] and SOLUTION[41]. In such systems, image processing operators and their parameters are manipulated till the desired output is achieved. Instead of an exhaustive search by varying all possible combinations of operators and their parameters, a rule base is used that guides the configuration. In our study, we define the methodology using which these rules can be generated for defining the optimal chain of image segmentation and texture analysis algorithms if the properties of an image are known.

In this paper we seek to establish the best image segmentation method given a known feature extraction method *T* that the experimenter has decided to work with. As discussed earlier, segmentation algorithms $\{S_1, S_2, \ldots S_N\}$ will generate region definitions $\{R_1, R_2, \ldots R_N\}$ for a known object (assume a single object region in an image for the sake of simplicity). If the application of feature method T generates n feature dimensional vectors $(l^n_{R_1}, l^n_{R_2}, ..., l^n_{R_N})$, then only one of these is the optimal. The optimal vector is supposed to best characterise the object, and therefore give the best results if used in the training data as opposed to the use of other vectors from the set, i.e. maximises recognition capability.

It should be mentioned here, that the question of determining the best feature extraction algorithm based on a given segmentation method is ill-posed, as the features that will work the best can not be predicted from the region definitions alone. In practice where we deal with more than one image, it may be unrealistic to apply the optimal segmentation algorithm on a per region/object basis, and the method that works well on the majority of regions may be chosen as optimal for the complete data set.

Our discussion is based on natural scene analysis of color images, as this is our primary research interest, however the discussions are relevant to most image analysis application areas. In our analysis, color is used to separate vegetation from other classes and then grey-scale features are used thereafter to classify further objects. Section 2 discusses the range of image segmentation and feature extraction tools that are used in scene analysis and in particular the ones that we select for analysis. In this study we focus on image segmentation algorithms that are based on region homogeneity and texture features computed on grey-scale images that analyse the spatial distribution of pixels. Section 3 describes the results of applying different combinations of image segmentation and texture feature extraction algorithms on the scene analysis task and demonstrates the variability in results. Our experiments are conducted on the publicly available MINERVA scene analysis benchmark (http://www.dcs.ex.ac.uk/minerva). The benchmark has a collection of natural scenes (448 images of natural scenes containing eight objects, namely grass, tree, leaves, bricks, pebbles, sky, clouds and road). It provides a good testing ground for the optimisation of image analysis tools for a practical application. In section 4, we formally describe the procedure for finding optimal segmentation algorithms for known objects with a pre-defined feature extraction procedure. Here we also discuss the parameter settings of segmentation algorithms. The conclusions are presented in section 5.

2. IMAGE SEGMENTATION AND TEXTURE ANALYSIS OF NATURAL SCENES

The recognition of artificial and natural objects in scenes is not trivial as objects are often complex[6,58], and requires sophisticated image processing and pattern recognition tools[9]. A review of scene analysis studies is available in [4,5,30]. Also [40] provides a detailed bibliography of research in this and other related computer vision areas. We first discuss some related work in the area of image segmentation and texture analysis in the context of scene analysis and then provide a critique of such work.

2.1 *Related Work*

Image segmentation algorithms can be classed as those based on histogram thresholding, edge based segmentation, tree/graph based approaches, region growing, clustering, probabilistic or Bayesian approaches, unsupervised neural networks, model-based approaches and other approaches. It is hard to generate good models of natural objects with respect to color, texture and shape and therefore some image segmentation methods are better suited to such tasks, e.g. unsupervised methods work better than model-based approaches. A number of reviews on image segmentation techniques are available including [14,17,22,35,43,45]. The key challenge for any segmentation methodology is to deal with noisy images, unimodal histogram images, low contrast and poor illumination, shadow effects, low resolution, and diffuse boundaries across objects. In studies where segmentation is the end process, its quality can be judged in isolation. However, as we have discussed, poor segmentation impacts on feature extraction from resultant regions. Hence, a more exhaustive approach based on classification accuracy is necessary since in most applications, accurate recognition of image objects is a primary objective. In this paper we have used four well-established segmentation methods including fuzzy c-means clustering (FCM), histogram based thresholding (HT), region growing (RG) and split and merge (SM).

Fuzzy c-means clustering assigns each sample to a cluster based on cluster membership. The segmentation of the image into different regions can be thought of as the assignment of pixels to different clusters. For a discussion on clustering techniques and merits of fuzzy clustering in comparison with other methods, see [16]. In histogram-based thresholding, the image histogram is used for setting various thresholds to partition the given image into distinct regions. It is expected that each region within the image will have some mean grey-level intensity and a small spread around this central value that pixels in this region will take. By examining the various peaks of the histogram denoting grey-levels that occur with the highest frequency, we can use them as thresholds to partition the image. Region growing is a procedure that groups pixels or subregions into larger regions. The simplest of these approaches is *pixel aggregation*, which starts with the first pixel of the image then starts growing that region by appending its neighbouring pixels that have similar properties (such as grey-level, texture, colour, etc.). After growing the first region it moves to the next pixel in the image that is not allocated to any region before. This process is continued until all of the pixels have been assigned to a region. The process can start with a manually selected pixel as the seed or using image histogram, or seed pixels can be automatically determined [1]. Split and merge is an image segmentation procedure that initially subdivides an image into a set of arbitrary, disjoint regions and then merges and/or splits the regions in an attempt to satisfy the conditions that ensure that the final regions are homogeneous. The split and merge algorithm iteratively works towards satisfying these homogeneity constraints.

A feature vector defines the unique characteristics of a given region by analysing its shape, texture, colour, or statistical information. In scene analysis, texture information plays a key role in the identification of objects. Texture techniques can be categorised as geometric and topological approaches, second or higher statistics based approaches, texture with masks and logical operators, texture with stochastic models or random walk, texture based on gradient information, texture based on spectral filters, and other methods. A review is available in [21]. Some attempts have been made to develop texture methods that have some visual significance, i.e. they have a strong correlation with our own visual system

[50]. However, most methods remain highly statistical in nature. Some of the methods proposed by researchers have become more established than others due to their widespread popularity. A number of different studies have investigated the usefulness of these texture feature extraction methods on artificial and real textures. Brodatz images have been used as a popular texture benchmark for studies interested in comparing texture analysis algorithms [7,37]. The Curet database has also been used for texture analysis for investigating reflectance in texture surfaces[13]. Some basis of objective performance comparison is however needed. Faugeras and Pratt[15] define the most popular comparison methods including synthesis, classification, and figure of merit. In the case of synthesis method, an artificial texture field is created on the basis of texture feature parameters that are obtained from the original field and some error functional is then performed on the original and reconstructed fields. The basic philosophy is that reconstruction error should be small for good texture measures. In classification methods, it involves the prediction of the classification error of independently categorised texture fields. In the third method of figure of merit, some functional distance measures between texture classes are developed in terms of feature parameters such that a large distance implies low classification error and vice-versa [55]. For scene analysis application, where the classification ability is a key factor, the second approach is to be preferred. Examples of other two approaches being used for texture comparison include, synthesis method [10], and figure of merit method [42].

2.2 *Critique*

There is considerable debate on the superiority of image segmentation and texture analysis algorithms. Natural objects are primarily defined using texture or colour information [4,33] and their segmentation is not a trivial task. Haralick and Shapiro[23] state that: "As there is no theory of clustering, there is no theory of image segmentation", (p. 100). Similarly. Fu and Mui[17] state that: "Almost all image segmentation techniques proposed so far are *ad hoc* in nature. There are no general algorithms that will work for all images", (p. 4). The selection of image segmentation techniques that are suited for a given image is therefore a difficult task. A range of evaluation methods for ranking how good image segmentation is have been proposed in literature [26,54,56,58,59]. These techniques are primarily based on ground truth segmented images and actual segmented images that can be compared either on the basis of shape, area, statistical distribution of pixels within them, or differences in features extracted from them. Such analysis is capable of defining a measure of performance for an image segmentation algorithm in isolation on a given data set. One of the daunting aspects of such work is to find the ground truth data of correctly segmented images and that is a fairly difficult task in the case of natural scenes. In outdoor scene analysis, the availability of accurate image segmentation ground truth data is nearly impossible. As such, most studies have used synthetic images of varying complexity and with increasing levels of noise contamination for evaluating image segmentation algorithms. These results are however difficult to generalise to outdoor scene analysis studies dealing with complex objects. In other words, a segmentation algorithm that performs well with synthetic regions will not necessarily perform well with real object regions.

Just as there is no suggestion on the superiority of segmentation algorithms, similarly there is no consensus on which texture measures are the best. Comparisons are often not reliable as experimental conditions are different. Appendix 2 details the different studies that have compared texture features and their key observations. Some of the key observations from above studies can be stated as follows. First, there is no consensus on which texture features perform the best. The results are very much application dependent and due to most studies working with small data sets, results are hard to generalise. Second, most texture evaluation studies are motivated by trying to understand the meaning of features and their individual discrimination ability. Hence, single or pair of features has been used. The objective is not to necessarily get the best classification ability, for example by combining feature sets. Third, texture evaluation is based on mostly those images that do not necessarily need segmentation prior to texture analysis. These include popular texture benchmarks such as Brodatz images [7]. Finally, we have come across only few studies that evaluate the utility of different texture methods on scene analysis benchmarks. In this paper we use seven texture analysis methods including autocorrelation (ACF) [47], co-occurrence matrices (CM) [3,18,19, 20,44,53], edge frequency (EF), Law's masks (LM), run length (RL) [47], Texture operators (TO) [30] and Texture Spectrum (TS) [25]. These are detailed in Appendix 1.

3. PERFORMANCE VARIABILITY

In our study, a total of four image segmentation methods have been applied to the 448 images from the MINERVA benchmark including fuzzy c-means clustering (FCM), histogram thresholding (HT), region growing (RG), and split and merge (SM). Each segmentation method yields its own unique output on a given image. Some example images with their different segmentation results are shown in Figure 1. Since we perform a leave-one-out cross-validation, all data is ground-truthed initially. We use a simple graphical user interface for assigning pixels to known classes for generating the training data. The number of regions generated by the four methods for different objects is shown in Table 1. Split and merge generates the maximum number of samples and takes the longest to compute. Region growing generates the smallest number of samples. Vegetation categories, including trees, grass and leaves have more samples than other natural objects such as sky, clouds, bricks, pebbles and road. It should be noted that the number of regions is much larger than the actual number of objects because of over-segmentation. On the application of seven texture extraction methods, to the segmented images, a total of 28 feature sets are derived. A data distribution plot shows strong overlap across vegetation classes (trees, grass and leaves) and other natural objects (sky, clouds, bricks, pebbles, and road). On the basis of texture features obtained on greyscale images, discriminating between these eight classes is very difficult. We therefore adopt a multistage classification strategy (multistage strategy has proved important in scene analysis for generating better classification accuracy, [36,51]). On the basis of colour histogram information, we discriminate between vegetation and natural object samples. Classification is therefore performed using two classifiers. The first classifier learns to classify vegetation classes and the second classifier learns to classify natural object classes. Both of these classifiers use grey-scale texture features. The experimental results are detailed on these two tasks separately. For each task, there are 28 feature sets.

3.1 *Vegetation data analysis*

In Figure 2 we show the classification rates of an optimised k nearest neighbour classifier for different feature sets. A range of other classifiers can also be used [27] but our previous experience suggests kNN be a good classifier in scene analysis. Each texture extraction method is applied to regions obtained from the four segmentation schemes. For the kNN classifier, the results of the average of leave-one-out cross-validation classification performance are quoted. The confidence intervals at 98% level have been marked on the recognition rates. On the whole we find that the edge frequency method gives the best performance. The best classification rate of 82.0% correct obtained using edge frequency measures computed on regions generated by the split and merge method. The worst performance of 52.2% correct classification is obtained by texture spectrum features computed on regions generated by region growing segmentation. In fact, there is a wide range of performances depending on which combination of image segmentation and texture extraction is chosen. Texture methods can be ranked in the following order of increasing variance with respect to their results on four segmentation methods: RL (σ=2.7), EF (σ=3.6), ACF (σ=4.4), TO (σ=5.4), CM (σ=6.1), LM (σ=6.3) and TS (σ=8.1). On the basis of these, it is reasonable to conclude that texture extraction methods such as RL and EF are more robust and least affected in our vegetation analysis by changes in different region definitions generated by different segmentation techniques. The other algorithms including LM, CM and TS methods are considerably more affected.

Another manner in which we can analyse our data is to measure the variability in performance of different texture algorithms with the same image segmentation method. This is shown in Figure 3. For each segmentation method, we plot the classification rates obtained using different texture extraction methods. In this case, a single segmentation strategy is not the best for all texture extraction methods. For each segmentation method, we can calculate the variability in performances obtained from different texture algorithms, and rank segmentation methods accordingly. Thus, in increasing order of variance we have:

SM (σ=6.3), RG (σ=7.1), FCM (σ=7.8), and HT (σ=8.3). In general, there is a greater variability across different segmentation methods (each with varying texture algorithms), compared to texture methods (each obtained on regions from different segmentation schemes).

3.2 *Other natural objects data analysis*

We can perform the above analysis on natural object data. The best classification performance of 76.5% correct is obtained using split and merge segmentation method and texture operator measures. The worst performance of 50.1% correct is obtained with histogram thresholding segmentation and run length features. In Figure 4 we display the recognition rates for each data set displayed for each feature extraction method separately. On the whole, the edge frequency and autocorrelation methods appear to the best two. Texture methods can be ranked in the following order of increasing variance with respect to their results on four segmentation methods: EF (σ=1.7), CM (σ=2.2), ACF (σ=2.8), LM (σ=4.5), RL (σ=5.4), TS (σ=6.9) and TO (σ=7.3). This ranking is dissimilar to that obtained in vegetation analysis.

In Figure 5, we display for each segmentation method how well the texture methods perform on the classification task. There appears to be a very similar trend with all texture methods used. This similarity in plots shows that the ranking of texture algorithms is preserved irrespective of the image segmentation method used. For each segmentation method, we can calculate the variability in performances obtained from different texture algorithms, and rank segmentation methods accordingly. Thus, in increasing order of variance we have: RG (σ=5.6), FCM (σ=7.3), HT (σ=7.6), and SM (σ=8.4).

4. OPTIMAL CHAINS

In the discussion above we have highlighted the variability in the performance of different chains or sequences of image segmentation and texture extraction algorithms for a scene analysis task. We mentioned earlier that we will define how to find the optimal segmentation algorithm for any given feature extraction method. In addition, we also stated that such analysis cannot be isolated from the type of image object being analysed. In our opinion, the optimality of the segmentation algorithm is best determined by the quality of texture features generated for the same object across a range of images. The region definitions that are best segmented will yield texture features that are most representative of that object, and their distribution across images for the same object and same texture method will be more compact or dense (lower spread), that in the case of sub-optimal segmentation methods. There are two main suggestions here. First, a data set that is most compact, i.e. least variability, is to be preferred over any other data set that has a higher variability. Second, in order to select an optimal segmentation method for a given texture method, or the overall best combination, we construct a radar plot of data spread for individual objects and estimate the area of the polygon so constructed. The data set with the least area is most compact and to be preferred. These two suggestions will allow the choice of appropriate training data set for a given image processing application.

Figure 6 shows two pseudo-algorithms. The first algorithm determines an optimal segmentation method for a collection of images whereas the second algorithm describes the selection of an optimal segmentation method for a given test image region. The first figure illustrates the use of a compactness measure C_p (described next), for determining such optimality for a three-class problem. In this case, segmentation method S_1 is preferable to S_2 since it generates most compact polygon. The second figure shows four example segmentations generated by segmentation methods (S_1, S_2, S_3, S_4) where optimal segmentation is best matched by method S_2.

4.1 *Compactness measure* C_p

There are a number of available measures of data distribution spread. Considering the fact that we are dealing with multi-dimensional data, the variance estimated as the sum of Euclidean distances of all points from the group centroid can be used as a simple measure of compactness. However, such measures are likely to suffer from the bias introduced by outliers in the estimate of the data distribution mean or centroid. Hence, we propose here a new measure of compactness, called C_P which is based on data partitioning scheme. On its basis we evaluate our feature sets to determine optimal segmentation methods for each object for each chosen texture method.

Algorithm Compactness C_P

1. Multivariate data is represented as a set of *n* feature vectors for a set of *N* measurements (Nxn matrix). Each row *i* of this matrix $(x_{i1}, x_{i2},..., x_{in})$ can be assigned to one of the *k* possible classes $\{c_1, c_2,...c_k\}$. If we consider each feature column as an axis, then on axis *j*, $j \le n$, the range of data R_j is given by $[\min_j, \max_j]$.

2. Given a real positive number *d*, we can have a list *L* of hypercubes of side length *d* covering the *N* data points. If we are to have side lengths d_j on different feature axis *j*, then we have a hypercuboid, or cell, whose volume is given as $\prod_{j=1}^{n} d_j$.

3. The number of cells on axis *j* is given by $P_j = \text{int}\left[\frac{R_j}{d_j}\right]$, and therefore the total number of cells is given by $H_{total} = \prod_{j=1}^{n} P_j$. Assign each data point to a given cell depending on the coordinates of the cell vertices.

4. Start by partitioning each axis into *M* partitions and at each stage note the number of non-empty cells. The number of partitions per axis *M* can vary from 0 to 31 as for data of dimensionality D, 32^D is a fairly large number of cells and further splitting does not give any advantage to the estimate.

5. Plot the normalised number of partitions per axis on the x-axis and the normalised number of non-empty cells for a given data set. The *y*-axis measure is weighted by a factor of $w = \left\{1 - \frac{1}{2^{(z)}}\right\}$, where *z* is the number of partitions per axis. Calculate the area under the curve as the compactness measure C_P .

Given that both x- and y-axis are within the [0,1] range, the measure C_P is bound within the [0,1] range and that data with dense distributions will have a lower measurement than sparse data. The compactness measure across different data sets is comparable if weighted by the number of samples considered.

The utility of the compactness measure is illustrated in Figure 7. A total of ten simulated data sets of dimensionality (d=2) are generated. The first seven data sets represent data with the same range but with varying level of compactness (Figure 7(a)- 7(g) are the most compact). The variability determined in terms of the sum of point distances from the data centroid, and the compactness is shown in sequence in Figure 7. In general, across these ten simulated data sets, both measures correlate extremely well (Pearson's correlation coefficient of 0.94) showing that C_P measures the concept of data variability. However, we prefer the compactness measure as it is more robust to the presence of outliers. To prove this, we investigate the relationship between C_P and variance s^2 on synthetic data. A total of 100 synthetically generated data sets are created each with 2000 uniformly distributed random data points in two dimension within a range of [0,1] with increasing number of outliers from 5 to 500 in steps of 5. The data without outliers is bounded within a rectangle defined by extreme points $\{(.33,.33),(.66,.66)\}$. The outliers are present within a rectangular tube, whose inside extreme points are $\{(.13,.13),(.86,.86)\}$ and the outside extreme points are $\{(.0,.0),(1.0,1.0)\}$. In Figure 8, we plot the normalised variance and compactness measures showing their relative rate of increase as more outliers are added. It is clearly visible that for small number of outliers, their rate of increase is similar, however, as more outliers are added, the variance measurement change is much larger than compactness change. This lends support to the argument that compactness measures similar property of data variance but does so with more robustness to outliers.

So far we have discussed our results with an underlying assumption that the segmentation methods that yield the most compact texture features are the best suited for analysis. Now we support this claim with the classifier analysis of our data. In section 4 we detailed the results of a kNN classifier applied to the 28 feature sets obtained by using 4 image segmentation algorithms followed by 7 texture analysis methods on a total of 448 natural scenes. It is reasonable that the choice of optimal image segmentation algorithms should give better classification performance compared to a random choice. In Tables 2 and 3 we answer the question: “Given that we can predict the optimal segmentation method for each texture analysis method, will this choice generate the best classification results when it comes to training and testing classifiers?” The answer is yes, however the relationship between compactness and classification success must be treated with caution (compact data has a less chance of overlap with other classes but not necessarily). Table 2 shows four columns labelled the first best, second best, third best and fourth best. The first column shows the texture analysis algorithm (ACF…RL). The first best column lists the image segmentation algorithm whose combination feature set yields the best classifier cross-validation result when testing unseen data. Similarly, the last three columns show the image segmentation algorithms whose combination feature sets gave the respectively ranked recognition performance. For example, ACF texture set generated with FCM segmentation gave the highest classifier recognition rate, followed by HT, SM and RG. Next to each image segmentation method we put within brackets the number of objects for which texture data obtained following that particular image segmentation algorithm was the most compact. For example, SM (2) implies that for 1 out of 3 vegetation objects, SM generated the most compact feature sets on two objects. Ideally, we expect that the winning segmentation methods will show the highest proportion of objects that had the most compact distributions. This is indeed the case as Table 2 has higher figures in the left half.

For each texture method we can now plot the compactness of its features resulting from a region definition generated from a segmentation algorithm. The radar plots in Figure 9 and 10 show this. Figures 9(*a*-*g*) show results for vegetation analysis and Figures 10(*a*-*g*) show results for other natural object analysis. Each axis of the plot corresponds to a different object under consideration. For each object, the compactness of the features generated by a fixed texture method and varying preceding segmentation algorithms is plotted. For example, in Figure 9(*a*) we show the compactness of autocorrelation function (ACF) features. For object *leaves*, ACF features based on FCM and RG segmentation algorithms have nearly the same compactness, SM is most compact and HT least compact. For the object *trees*, result from HT segmentation algorithm is the most compact, followed by in order SM, RG and FCM. Finally for the object *grass*, the order is SM, FCM, HT and RG. When these measurements on different object axes are joined together, the smallest area polygon can give a rough indication of the corresponding optimal segmentation method. Figures 9(*b*-*g*) correspond to texture analysis methods co-occurrence matrices (CM), edge frequency (EF), Law's method (LM), run length (RL), texture operators (TO) and texture spectrum (TS) respectively. We note two important observations. First, the region growing method generates the least compact set of features regardless of the texture method used. Second, split and merge segmentation algorithm is the best but for most texture methods, the histogram thresholding and fuzzy clustering segmentation methods are in close competition. The results on the analysis of other natural objects is shown in Figures 9(*a*-*g*) for texture methods ACF, CM, EF, LM, RL, TO and TS respectively. The following conclusions can be drawn. First, split and merge segmentation performance is fairly convincing in terms of the compactness measure. Second, region growing generates region definitions with the least compact features. In addition to the visual analysis of polygons in these figures, we can also comment on the differences in compactness across different objects. For example, in vegetation analysis, *grass* shows more variability or least compactness compared to *tree* and *leaves* features. Similarly, in natural data analysis, clouds and bricks are more variable compared to other objects.

Consider a simple image segmentation algorithm selection rule, "select the algorithm that generates the smallest least polygon area (as in Figures 9 and 10)". The polygon area computations are shown in Tables 4 and 5 for vegetation and other natural objects data. The results can be considered now for each feature extraction method for vegetation analysis. For ACF texture analysis, on the basis of compactness we select SM segmentation algorithm, however, the best results were obtained with FCM of 74.6% (classification accuracy difference of 5.5%). This result, though not in line with our argument, is hardly surprising if we see Figure 9(*a*) where SM and FCM are in close competition. For CM texture analysis, considering the smallest area polygon of Figure 9(*b*) for SM algorithm, this would be our suggested choice. Fortunately, this is also the best algorithm for CM features with the best classification accuracy of 69.9%. For EF texture analysis, SM algorithm is the selected choice and this combination also gives the highest classification accuracy of 82.0%. For LM texture analysis, considering the smallest area polygon of Figure 9(*d*) for FCM algorithm, this would be our suggested choice. The best algorithm for LM features is however SM with best classification accuracy of 73.5% (classification accuracy difference of 4%). For RL texture analysis, SM algorithm is the most compact with the smallest polygon area and is our selected choice. It also generates the best classification accuracy of 63.4%. For TO texture analysis, the least polygon compactness is given by SM segmentation method based data set, which by far generates the best classification accuracy of 77.8%. Finally for TS texture analysis, SM segmentation based data set generates the least polygon compactness and the highest classification accuracy of 63.4%. The above results show that in 5 out of 7 cases, we are able to correctly choose the correct segmentation-texture analysis algorithm combination that would maximise classification accuracy.

Now let us consider the other natural object data in turn as shown in a similar fashion in Table 3. For ACF analysis, SM is our selected choice however FCM algorithm gave better results (classification accuracy difference of 5%). For LM texture analysis, LM is our selected choice, however, the best results are obtained using FCM segmentation based data (classification accuracy difference of 7%). For CM, EF, RL, TS and TO texture analysis, our selected choice of SM segmentation algorithm again is the best choice with respect to classification accuracy. The above results show that again in 5 out of 7 cases, we are able to correctly choose the correct segmentation-texture analysis algorithm combination that would maximise classification accuracy. For both vegetation and natural object analysis, ACF and LM texture analysis methods are the ones where we fail to correctly determine the optimal dataset based on polygon compactness.

The following important key observations can be drawn from the above discussion:

a) In our study, we find that the Split and Merge segmentation algorithm generates the best regions which give the most compact set of feature regardless of the texture analysis method used.

b) From a total of 14 predictions made using our technique on the suitability of image segmentation algorithm for each texture method (7 times for vegetation analysis and 7 times for other natural object data analysis), 10 times we correct predict the best image segmentation algorithm which turns out to be split and merge (71.4% correct).

c) Image segmentation algorithms that gives the best result with one texture analysis method compared to all others, is also fairly good with the others, e.g. SM algorithms gives throughout a very good performance. However, there is no linear correlation between how the segmentation algorithms rank for different texture methods.

4.2 *Image segmentation algorithm parameters*

All image segmentation algorithms used in this study, and most otherwise, have some parameters that can influence the nature of segmentation. For example, in the case of algorithms discussed here, these are: FCM (*number of clusters, maximum iterations, termination threshold, fuzzy factor*); HT algorithm (*distance between peaks*), RG algorithm (*threshold*), SM algorithm (*node and merge factor*). For FCM algorithm, for determining the number of clusters we generate between 1 and 10 clusters per image and use the figure that generates the least value of Davies-Bouldin cluster validity index. The fuzzy factor is set to 2.0, termination threshold equals 0.5 and the total number of iterations is 15. These parameters were set manually in line with [49]. In the case of histogram thresholding segmentation algorithm, the aim is to find peaks that represent different objects. We set a minimum distance of 10 grey-levels between adjacent peaks for them to be considered as representing two different objects. For region growing algorithm, pixels are merged with the growing region if the difference in grey-levels is no more than 30 grey-levels. Finally, for the split and merge algorithm we set the node size to 4 and the merge factor to 10. The node size corresponds to the minimum size that an image quadrant can have after recursive splitting. The merge factor defines acceptable difference between the average grey-level of quadrant to be merged and that of its neighbours to allow the merge process to take place. Our parameters have been set based on experimentation with a number of images to select the ones that work the best on the majority of images.

Optimisation of parameters for image segmentation algorithms is a tedious task and there are very few, if at all, automated methods of setting these. Our parameter settings are kept fixed for all images although there is a good argument for developing a methodology to set these on a per image basis. A simple manner in which this can be done is to select those parameters that lead to the largest Euclidean distance between features from different objects within the same image. This will ensure that the features have been extracted from well-segmented regions and that these are highly separable. On a collection of images, the methodology described in this paper of choosing segmentation methods that generate the most compact features can be used for finding the best set of parameters. Hence, similar to our comparison of FCM, RG, HT and SM algorithms, we could consider N different versions of FCM algorithm alone with different parameter settings. The main contribution of this paper therefore is the illustration of a generic framework using which image segmentation algorithms and their parameters can be selected. Obviously an exhaustive analysis on parameter selection itself is outside the scope of this paper for the segmentation algorithms considered taking into account the amount of variations possible in parameter setting.

5. CONCLUSIONS

In this paper we have investigated the variability in scene classification ability of a system that uses different combinations of image segmentation and texture extraction algorithms. We have argued that optimising the scene analysis chain is not simply a matter of finding the best image segmentation or texture extraction method in isolation. The different combinations give vastly different results and these component interactions must be taken into account when designing a practical scene analysis system. In our analysis on a public scene analysis benchmark MINERVA, we have shown that the difference between the best and the worst component combinations is as great as 25% or more classification accuracy. It should be remembered that texture extraction method rankings on synthetic texture benchmark analysis are not preserved when working with naturally found textures. For example, Singh and Sharma [46] find that autocorrelation and co-occurrence matrices give some of the best results on Meastex and Vistex benchmarks, whereas edge frequency results are only modest in comparison. However, in this paper we have presented results showing edge frequency texture method for scene analysis to be a powerful feature extractor despite its simplicity. Also, in our analysis, the split and merge algorithm works very well. The main contribution of our paper has been a detailed methodology for selecting optimal image operator chain and the introduction of new measures such as compactness measure C_P and area of the radar plot for determining optimality. Further work should now address parameter optimisation of image segmentation algorithms using the same methodology.

REFERENCES

1. R. Adams and L. Bischof, "Seeded region growing", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 6, 1994.
2. M.F. Augustejin, "Performance evaluation of texture measures for ground cover identification in satellite images by means of a neural-network classifier", IEEE Transactions on Geoscience and Remote Sensing, vol. 33, pp. 616-625, 1995.
3. A. Baraldi and F. Parmigianni, "An investigation of the textural characteristics associated with gray level co-occurrence matrix statistical parameters", IEEE Transactions on Geoscience and Remote Sensing, vol. 33, no. 2, pp. 293-302, 1995.
4. J. Batlle, A. Casals, J. Freixenet and J. Marti, "A review on strategies for recognising natural objects in colour images of outdoor scenes", Image and Vision Computing, vol. 18, pp. 515-530, 2000.
5. D.C. Becalick, Natural scene classification using a weightless neural network, PhD Thesis, Department of Electrical and Electronic Engineering, Imperial College, London, 1996.
6. M. Betke and N.C. Makris, "Information-conserving object recognition", Technical Report no. CS-TR-3799, Computer Vision Laboratory, University of Maryland, 1997.
7. P. Brodatz, Textures: a photographic album for artists and designers, Dover publications, New York, 1966.
8. J.M.H. Buf, M. Kardan and M. Spann, "Texture feature performance for image segmentation", Pattern Recognition, vol. 23, no. 3/4, pp. 291-309, 1990.
9. N.W. Campbell, W.P.J. Mackeown, B.T. Thomas, and T. Troscianko, "Interpreting image databases by region classification", Pattern Recognition, vol. 30, no. 4, pp. 555-563, 1997b.
10. R.W. Conners and C.A. Harlow, "A theoretical comparison of texture algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, no. 3, pp. 204-222,1980.
11. C.C. Chen and C.C. Chen, "Filtering methods for texture discrimination", Pattern Recognition Letters, vol. 20, pp. 783-790, 1999.

12. P.C. Chen and T. Pavlidis, “Segmentation by texture using correlation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 5, no. 1, pp. 64-68, 1981.

13. K.J. Dana, B. van Ginneken, S.K. Nayar and J.J. Koenderink, “Reflectance of texture of real world surfaces”, ACM Transactions on Graphics, vol. 18, no. 1, pp. 1-34, 1999.

14. L.S. Davis, “Image texture analysis techniques - a survey”, Digital Image Processing, Simon and R. M. Haralick (eds.), pp. 189-201, 1981.

15. O.D. Faugeras and W.K. Pratt, “Decorrelation methods of texture feature extraction”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, no. 4, pp. 323-333, 1980.

16. H. Frigui and R. Krishnapuram, “A robust competitive clustering algorithm with applications in computer vision”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 5, pp. 450-465, 1999.

17. K.S. Fu and J.K. Mui, “A survey on image segmentation”, Pattern Recognition, vol. 13, pp. 3-16, 1981.

18. J.F. Haddon and J.F. Boyce, “Image segmentation by unifying region and boundary information”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 10, pp. 929-948, 1990.

19. J.F. Haddon and J.F. Boyce, “Co-occurrence matrices for image analysis”, Electronics and Communication Engineering Journal, pp. 71-83, 1993.

20. R.M. Haralick, K. Shanmugam and I. Dinstein, “Textural features for image classification”, IEEE Transactions on Systems, Man and Cybernetics, vol. 3, no. 6, pp 610-621, 1973.

21. R.M. Haralick, “Statistical and structural approaches to texture”, Proceedings of IEEE, vol. 67, pp. 786-804, 1979.

22. R.M. Haralick, Image segmentation survey, in Fundamentals in Computer Vision, O. D. Faugeras (ed.), pp. 209-224, Cambridge University Press, Cambridge, 1983.

23. R.M. Haralick and L.G. Shapiro, “Survey- image segmentation techniques”, Computer Vision Graphics and Image Processing, vol. 29, pp. 100-132, 1985.

24. R.M. Haralick and L.G. Shapiro, Computer and robot vision, vol. 1, Addison Wesley, 1993.

25. D.C. He and L. Wang, Texture features based on texture spectrum, Pattern Recognition, vol. 25, no. 3, pp. 391-399, 1991.

26. A. Hoover, G. Jean-Baptiste, X. Jiang, P.J. Flynn, H. Bunke, D.B. Goldof, K. Bowyer, D.W. Eggert, A. Fitzgibbon and R.B. Fisher, "An experimental comparison of range image segmentation algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 7, pp. 673-689, 1996.

27. A.K. Jain, R.P. Duin and J. Mao, "Statistical pattern recognition: an overview", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 4-37, 2000.

28. B. Jahne, Hausseccker, Peter Geissler Handbook of computer vision and applications, Academic Press, 1999.

29. C.E. Liedtke and A. Bloemer, "Architecture of the knowledge-based configuration system for image analysis CONNY", Proc. 11th ICPR conference, Hague, vol. 1, pp. 375-378, 1992.

30. W. Mackeown, A labelled image database and its application to outdoor scene analysis, PhD Thesis, University of Bristol, UK, 1994.

31. V. Manian, R. Vasquez and P. Katiyar, Texture classification using logical operators, IEEE Transactions on Image Analysis, vol. 9, no. 10, pp. 1693-1703, 2000.

32. P.P. Ohanian and R.C. Dubes, "Performance evaluation for four classes o
Recognition, vol. 25, no. 8, pp. 819-833, 1992.

33. Y. Ohta, T. Kanade and T. Sakai, "Colour information for region segmentation", Computer Graphics, Vision and Image Processing, vol. 13, pp. 222-241, 1980.

34. T. Ojala, M. Pietikäinen and D. Harwood, "A comparative study of texture measures with classification based on feature distributions", Pattern Recognition, vol. 29, pp. 51-59, 1996.

35. N.R. Pal and S.K. Pal, "A review on image segmentation techniques", Pattern Recognition, vol. 26, pp. 1277-1294, 1993.

36. J.A. Parikh, "A comparative study of cloud classification techniques", Remote Sensing of Environment, vol. 6, pp. 67-81, 1977.

Pattern Analysis and Machine Intelligence, vol. 4, no. 4, pp. 436-441,1982.

43. P.K. Sahoo, S. Soltani, A.K.C. Wong and Y.C. Chen, "A survey of thresholding techniques", Computer Vision, Graphics and Image Processing, vol. 41, pp. 233-260, 1988.
44. S. Singh, M. Markou and J.F. Haddon, "Nearest Neighbour Classifiers in Natural Scene Analysis", Pattern Recognition, vol. 34, issue 8, pp. 1601-1612, 2001.
45. S. Singh and K.J. Bovis, "Medical image segmentation in digital mammography", in Advanced Algorithmic Approaches to Medical Image Segmentation, J. Suri, K. Setarehdan and S. Singh (eds.), Springer, London, 2001.
46. S. Singh and M. Sharma, "Texture experiments with Meastex and Vistex benchmarks", Proc. International Conference on Advances in Pattern Recognition, Lecture Notes in Computer Science no. 2013, S. Singh, N. Murshed and W. Kropatsch (eds.), Springer, 2001.

47. M. Sonka, V. Hlavac and R. Boyle, Image processing, analysis and machine vision, PWS press, 1998.
48. J. Strand and T. Taxt, “Local frequency features for texture classification”, Pattern Recognition, vol. 27, no. 10, pp. 1397-1406, 1994.
49. M.A. Sutton and J. Bezdek, “Enhancement and analysis of digital mammograms using fuzzy models”, 26th Applied Image and Pattern Recognition (AIPR) workshop, Proc. SPIE 3240, SPIE press, Washington, pp. 179-190, 1998.
50. H. Tamura, S. Mori and T. Yamawaki, “Textural features corresponding to visual perception”, IEEE Transactions on Systems, Man and Cybernetics, vol. 8, no. 6, pp. 460-473,1978.
51. A. Vailaya, A. Jain, and H.J. Zhang, “On image classification: city images vs. landscapes”, Pattern Recognition, vol. 31, no. 12, pp. 1921-1935, 1998.
52. L. van Gool, P. Dewaele and A. Oosterlinck, “Texture analysis: Anno 1983”, Computer Vision, Graphics and Image Processing, vol. 29, pp. 336-357,1985.
53. R.F. Walker, P. Jackway and I.D. Longstaff, “Improving co-occurrence matrix feature discrimination”, Proc. of DICTA’95, 3rd International Conference on Digital Image Computing: Techniques and Applications, pp. 643-648, 1995.
54. Z. Wang, A. Guerriero and M.D. Sario, “Comparison of several approaches for the segmentation of texture images”, Pattern Recognition Letters, vol. 17, pp. 509-521, 1996.
55. A. Webb, Statistical pattern recognition, Arnold, London, 1999.
56. J.S. Weszka and A. Rosenfeld, “An application of texture analysis to materials inspection”, Pattern Recognition, vol. 8, pp. 195-199, 1976.
57. J.S. Weszka, C. R. Dyer and A. Rosenfeld, “A comparative study of texture measures for terrain classification”, IEEE Transactions on Systems, Man and Cybernetics, vol. 6, no. 4, pp. 269-285, 1976.
58. W.A. Yasnoff, J.K. Mui and J.W. Bacus, “Error measures for scene segmentation”, Pattern Recognition, vol. 9, pp. 217-231, 1977.
59. Y.J. Zhang, “Evaluation and comparison of different segmentation algorithm”, Pattern Recognition Letters, vol. 18, pp. 963-974, 1997.

Captions for Figures and Tables

Figure 1 Sample image segmentation.

Figure 2 Comparing texture algorithms for vegetation analysis.

Figure 3 Comparing segmentation techniques for vegetation analysis.

Figure 4 Comparing texture algorithms for natural object analysis.

Figure 5 Comparing segmentation techniques for natural object data analysis.

Figure 6 Pseudo-algorithms for determining optimal segmentation method

Figure 7 Data compactness measurement using C_P: (*a*-*g*) data of increasing sparsity; (*h*-*j*) data with increasing number of outliers. The last three plots show outliers.

Figure 8 Robustness of C_p vs. variance

Figure 9 Radar plots: (*a*-*g*) Vegetation data

Figure 10 Radar plots: (*a*-*g*) Natural object data

Table 1. MINERVA benchmark data composition in terms of regions generated by different segmentation methods.

Table 2. The correlation between the compactness measure and the recognition rates for vegetation data

Table 3. The correlation between the compactness measure and the recognition rates for other natural object data

Table 4 The polygon area from the radar plots 8(*a*-*g*) for vegetation analysis

Table 5 The polygon area from the radar plots 9(*a*-*g*) for other natural object data analysis

Class	FCM	Histogram Thresholding	Region Growing	Split and Merge
Trees	387	314	180	316
Grass	268	241	69	379
Sky	293	419	187	400
Clouds	247	303	176	315
Bricks	137	274	143	302
Pebbles	121	114	65	310
Road	152	196	59	215
Leaves	206	184	134	248
Total	1811	2045	1013	2485

Table 1

Segmentation / Texture	1st Best		2nd Best		3rd Best		4th Best	
ACF	FCM	0	HT	1	SM	2	RG	0
CM	SM	1	HT	2	FCM	0	RG	0
EF	SM	2	HT	0	FCM	1	RG	0
LM	SM	2	FCM	1	RG	0	HT	0
RL	SM	2	RG	0	FCM	0	HT	1
TO	SM	3	FCM	0	HT	0	RG	0
TS	SM	2	FCM	1	HT	0	RG	0

Table 2

Segmentation / Texture	1st Best		2nd Best		3rd Best		4th Best	
ACF	FCM	1	RG	0	SM	3	HT	1
CM	SM	3	FCM	0	RG	0	HT	2
EF	SM	3	FCM	2	RG	0	HT	0
LM	RG	0	FCM	0	SM	2	HT	3
RL	SM	4	RG	0	FCM	0	HT	1
TO	SM	4	FCM	1	RG	0	HT	0
TS	SM	4	FCM	0	RG	0	HT	1

Table 3

Algorithm	FCM	HT	RG	SM
ACF	.232	.216	.197	.111
CM	.211	.194	.158	.149
EF	.138	.206	.155	.094
LM	.065	.085	.163	.079
RL	.004	.008	.029	.002
TO	.225	.263	.289	.202
TS	.023	.032	.088	.022

Table 4

Algorithm	FCM	HT	RG	SM
ACF	.185	.231	.369	.146
CM	.493	.309	.394	.2749
EF	.212	.315	.303	.195
LM	.217	.155	.250	.145
RL	.044	.050	.075	.019
TO	.418	.475	.511	.365
TS	.121	.060	.142	.045

Table 5

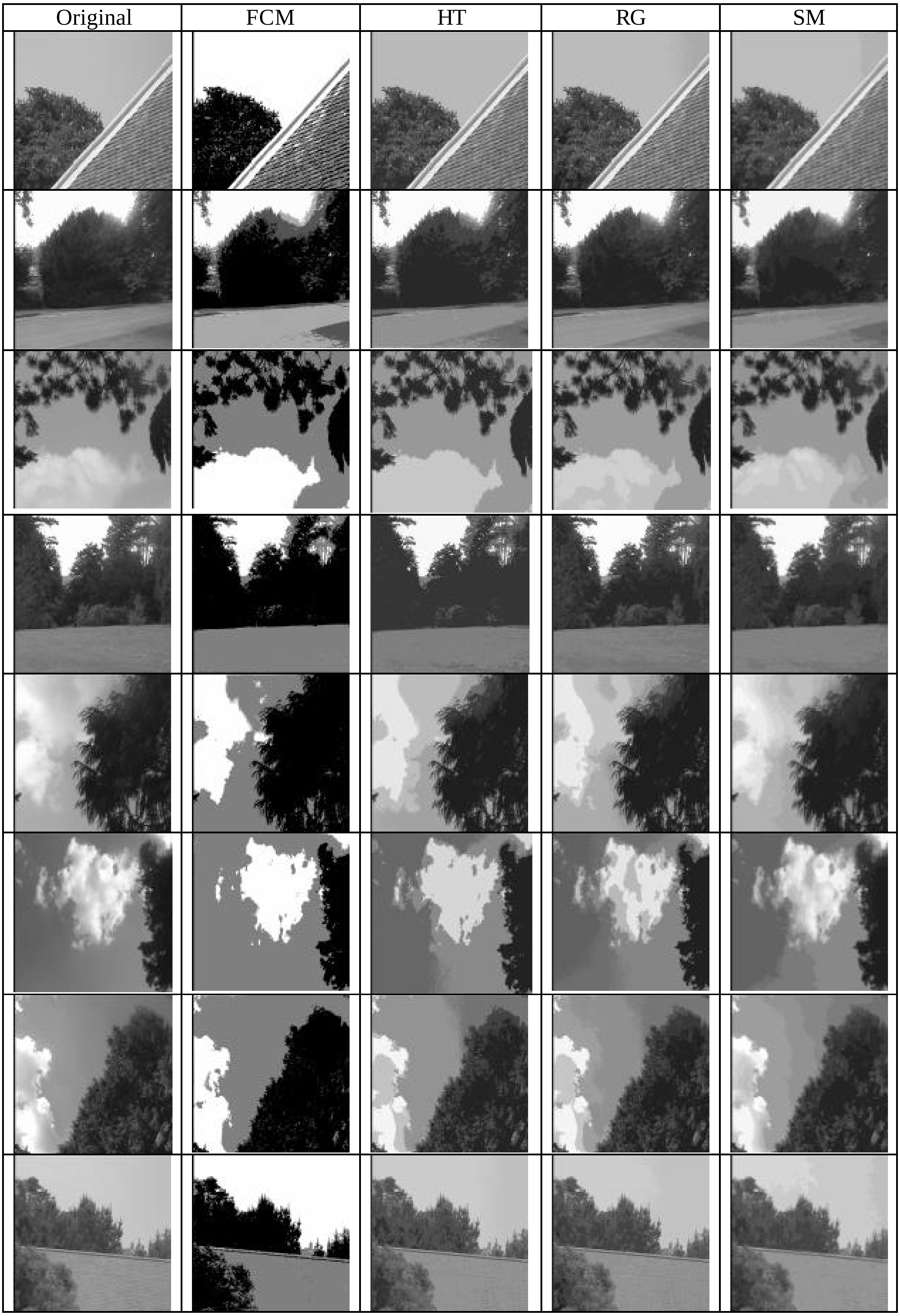


Figure 1

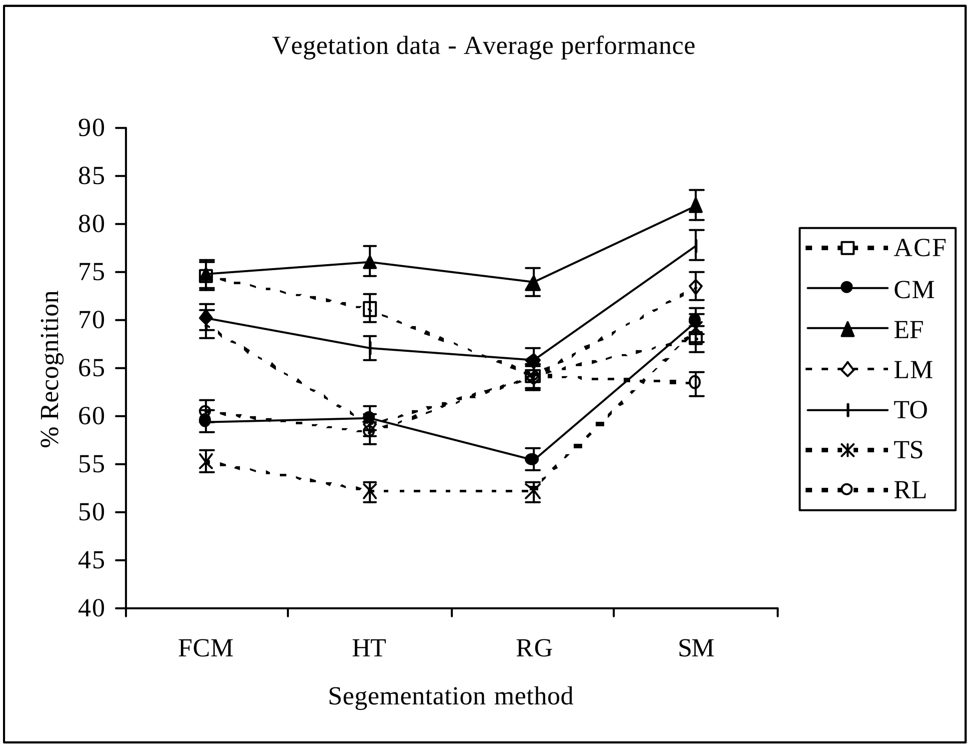


Figure 2

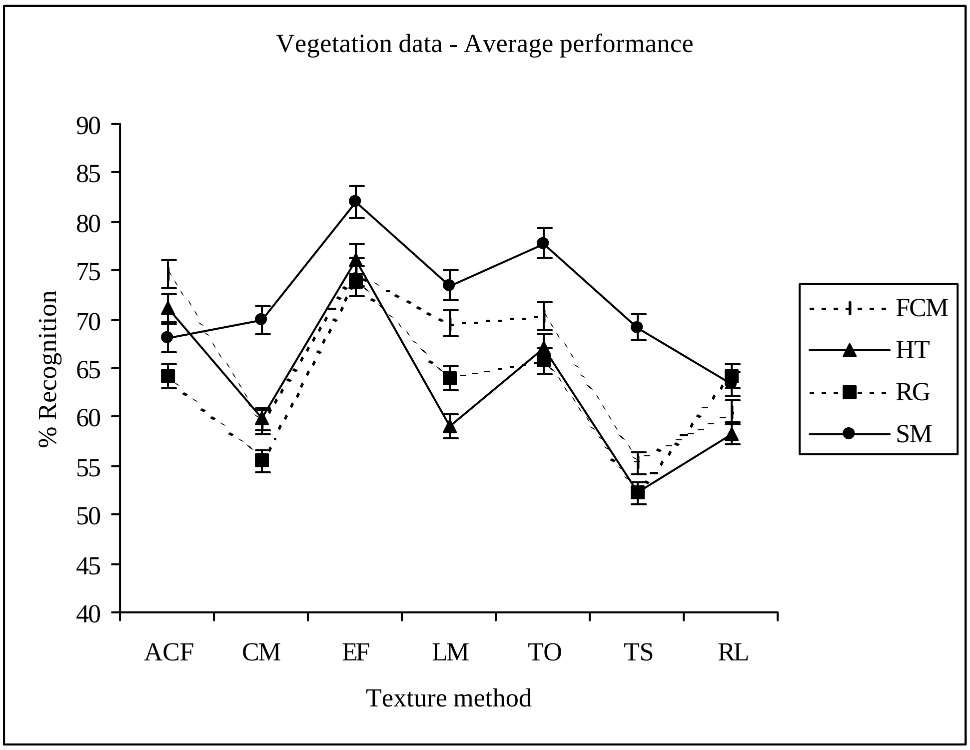


Figure 3

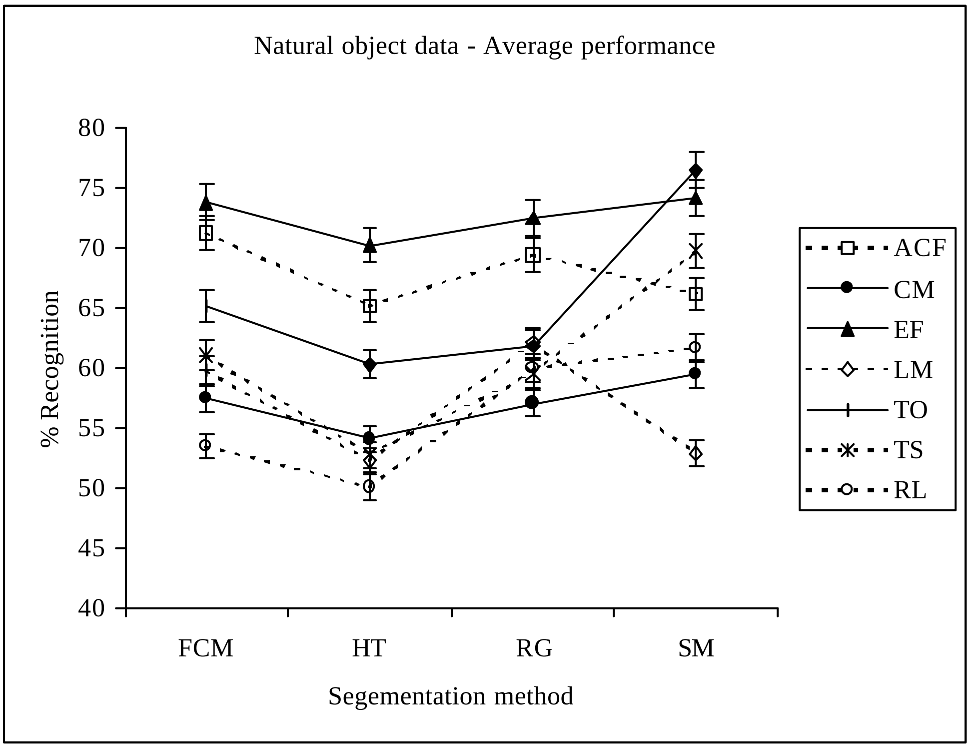


Figure 4

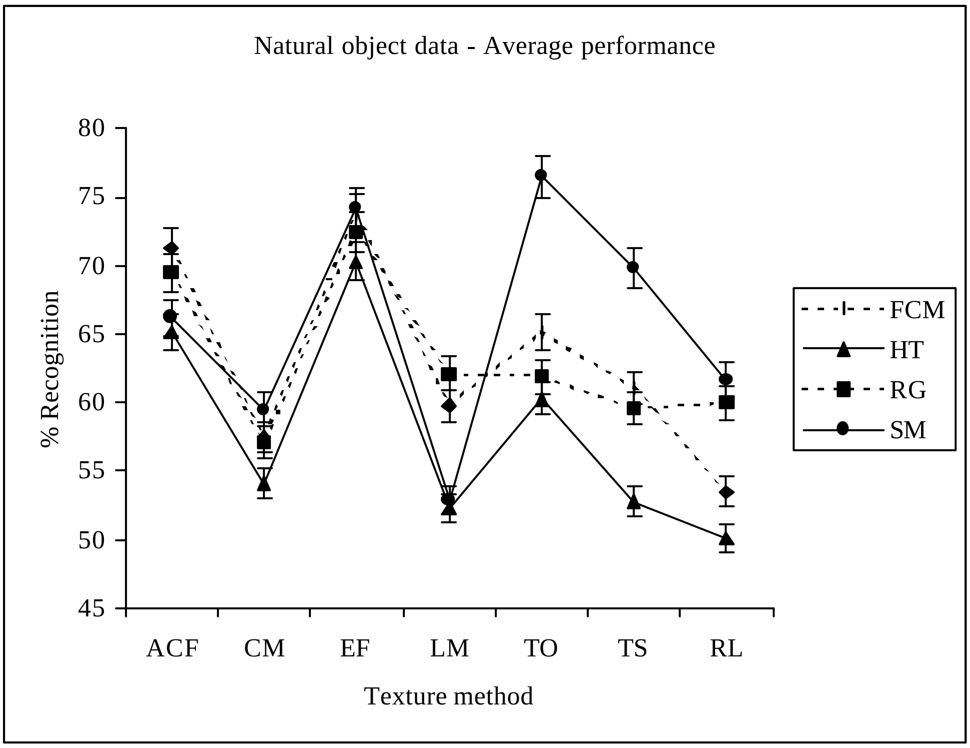
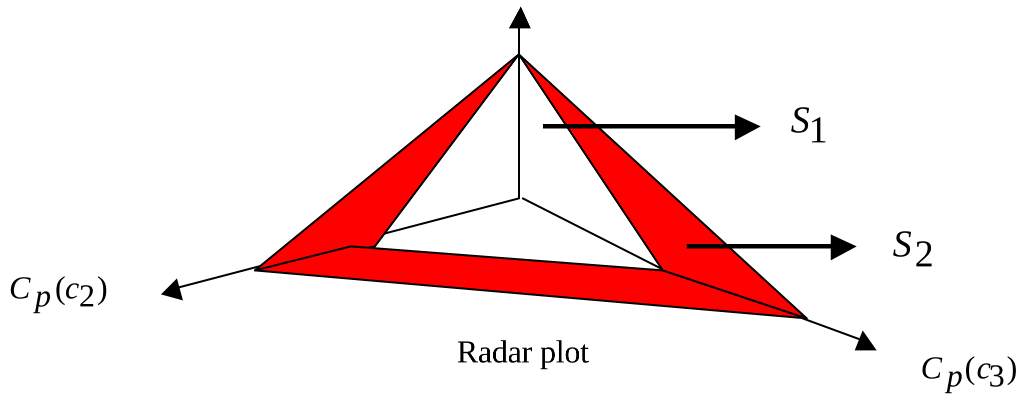


Figure 5

Pseudo-Algorithm for Determining Optimal Segmentation Method on Collection of Images

1. Given a collection of images containing object classes $(c_1, c_2, ..., c_k)$, segmentation methods $\{S_1, S_2, \ldots S_N\}$ and texture feature method T.
2. Calculate the compactness metric C_p for data of each class based on segmentation algorithms $\{S_1, S_2, \ldots S_N\}$.
3. Plot C_p for each segmentation algorithm separately as a radar plot where each axis corresponds to different object.
4. Optimal segmentation method is based on the least polygon area based on radar plots, e.g. S_1 is preferable to S_2 in the figure below.



Radar plot

Pseudo-Algorithm for Determining Optimal Segmentation Method on a Single Test Image Region

1. Given a test image region R, segmentation methods $\{S_1, S_2, \ldots S_N\}$, classifier C, and training data set Ω.
2. Generate texture features based on different segmented regions as $(l^n_{R_1}, l^n_{R_2}, ..., l^n_{R_N})$, for n dimensional texture features.
3. Calculate the posteriori probability of the samples using classifier C of belonging to class c_i as $(p_i(l^n_{R_1}), p_i(l^n_{R_2}), ..., p_i(l^n_{R_N}))$.
4. The optimal segmentation method is associated with the maximum probability estimate, i.e. $\underbrace{\max}_{i=1...k} \left\{ p_i(\lambda^n_{R_1}), p_i(\lambda^n_{R_2}), ..., p_i(\lambda^n_{R_N}) \right\}$.

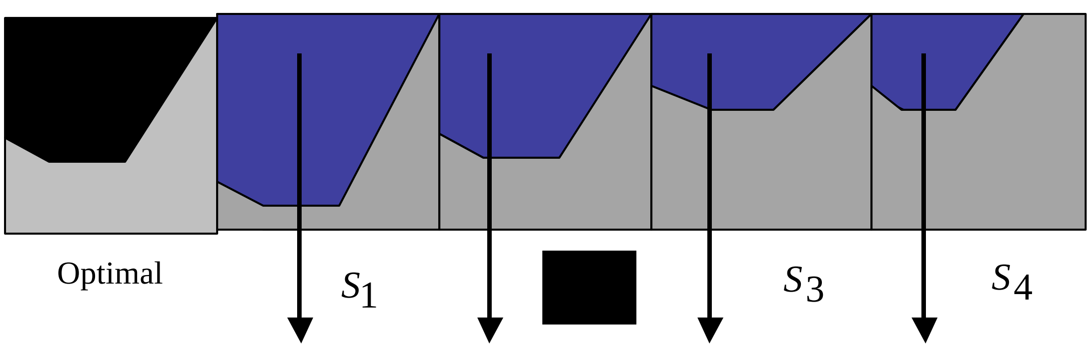
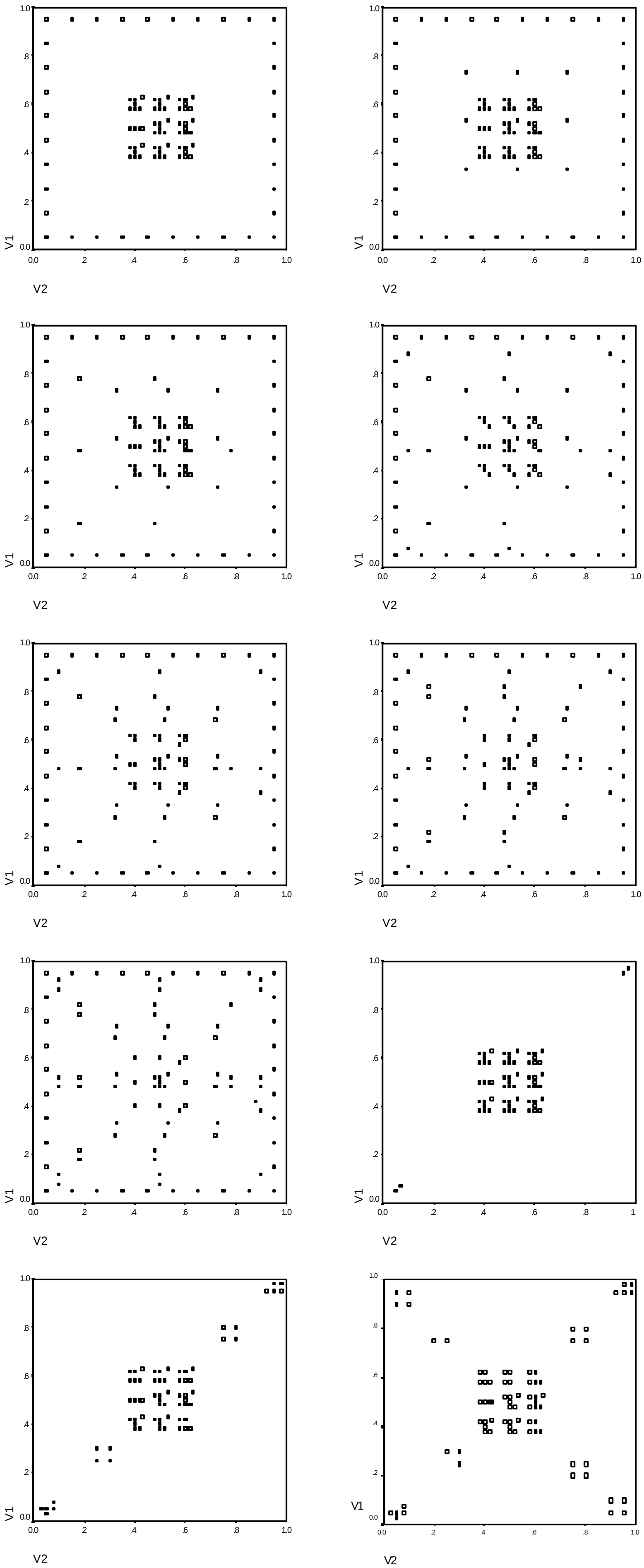


Figure 6



(a)

$(s = .262)\ (C_p = .356)$

(b)

$(s = .272), (C_p = .396)$

(c)

$(s = .286), (C_p = .418)$

(d)

$(s = .313), (C_p = .442)$

(e)

$(s = .322), (C_p = .452)$

(f)

$(s = .339), (C_p = .460)$

(g)

$(s = .469),\ (C_p = .371)$

(h)

$(s = .132), (C_p = .154)$

(i)

$(s = .182), (C_p = .165)$

(j)

$(s = .234), (C_p = .209)$

Figure 7

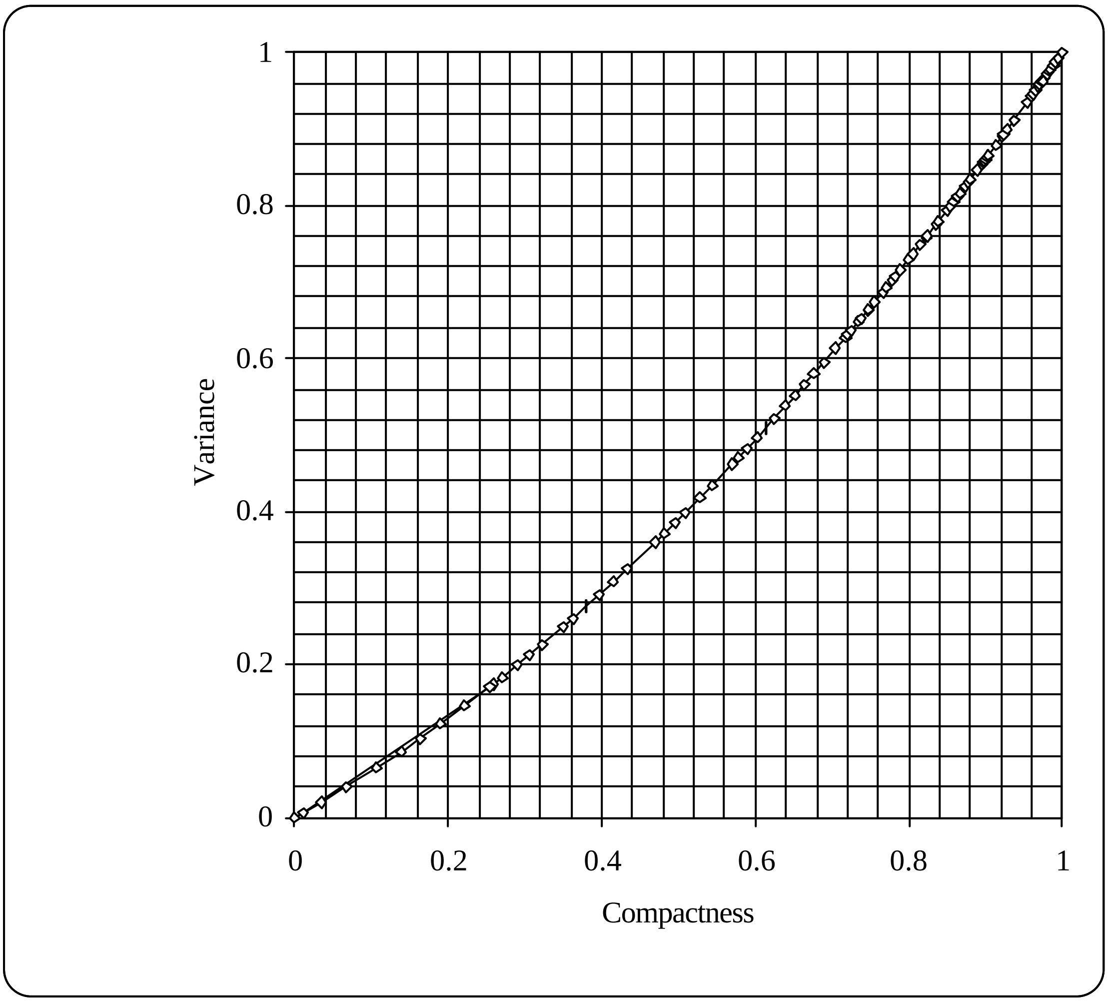
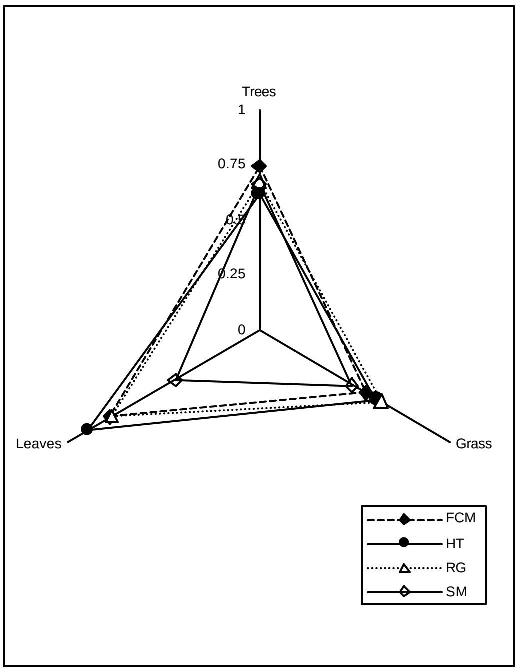
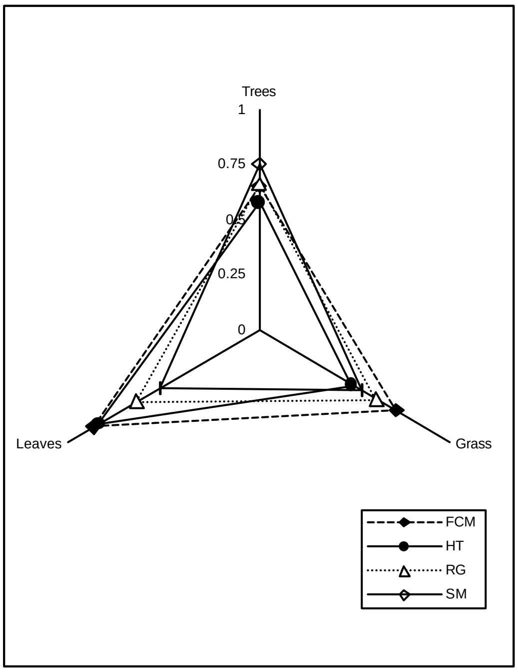


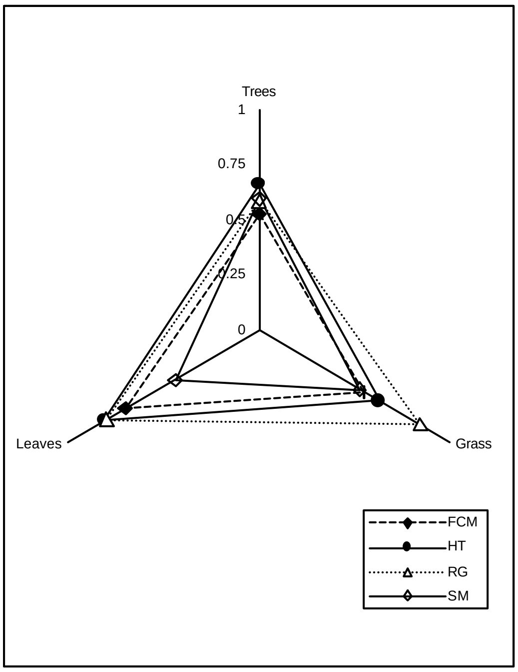
Figure 8



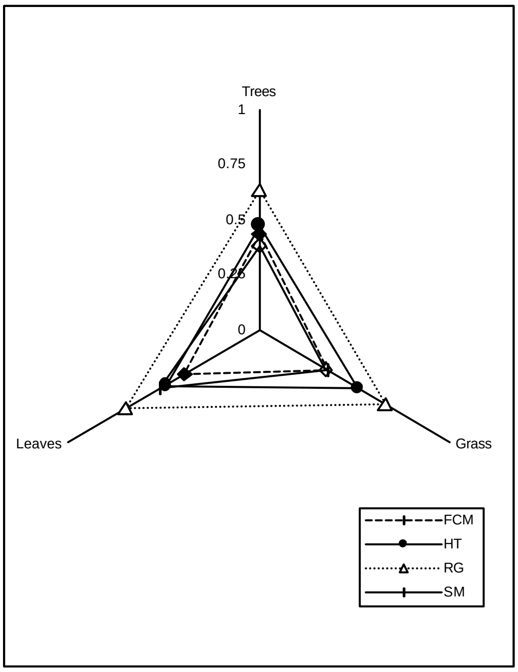
(*a*) ACF



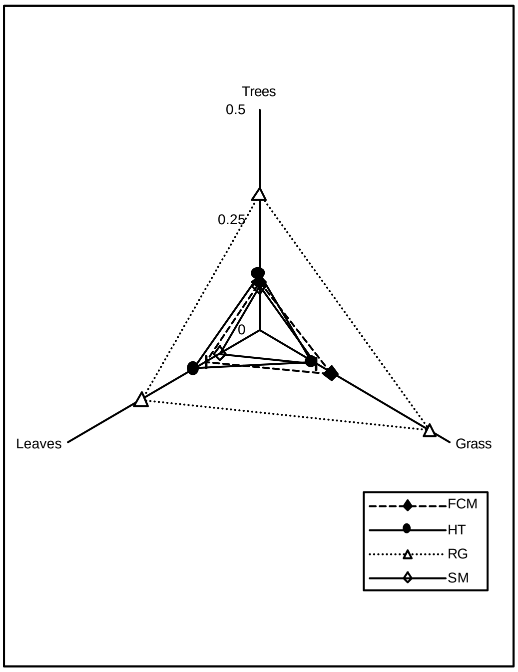
(*b*) CM



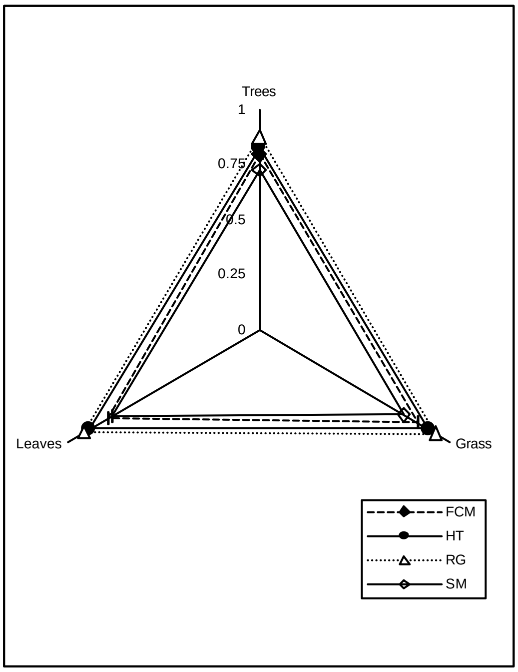
(*c*) EF



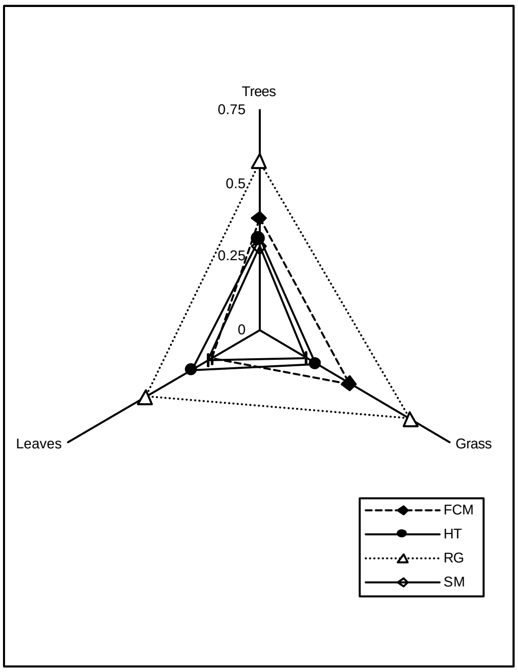
(*d*) LM



(*e*) RL

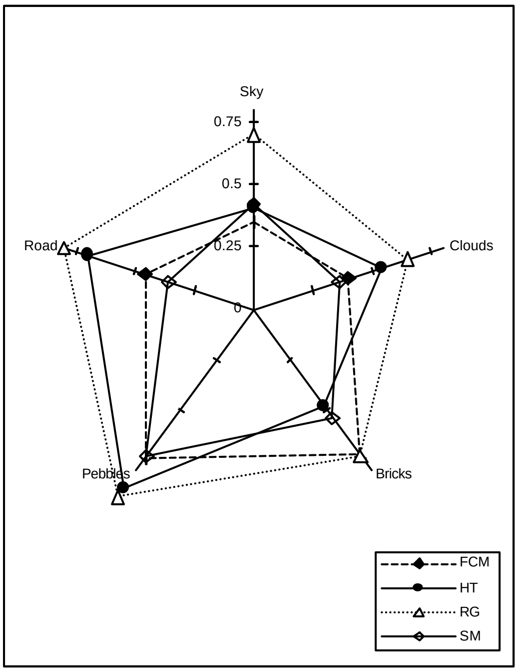


(*f*) TO

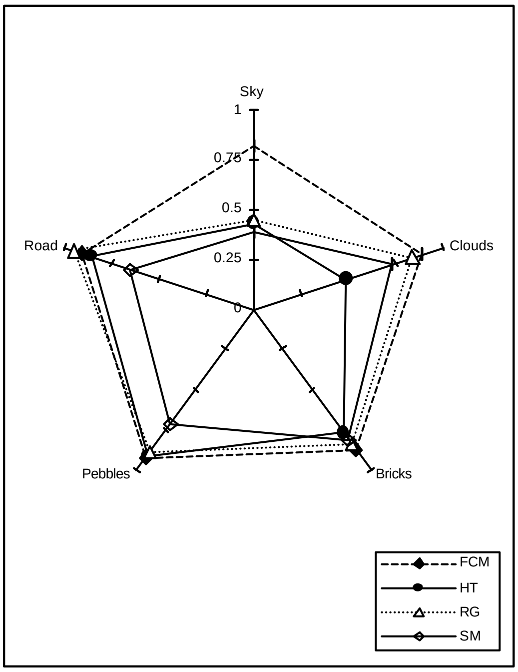


(*g*) TS

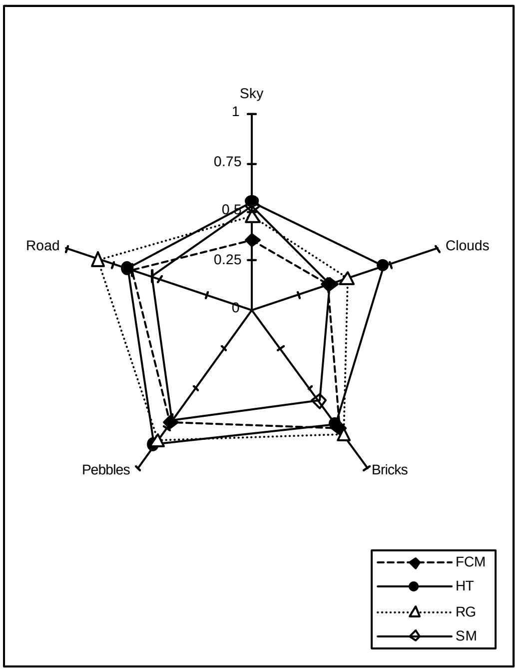
Figure 9



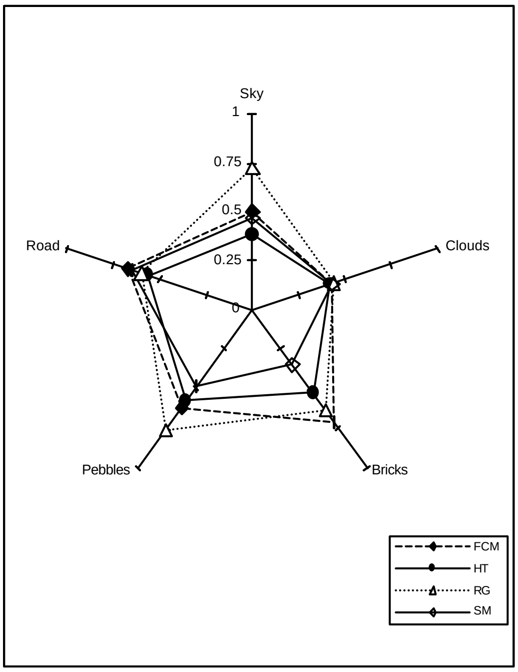
(*a*) ACF



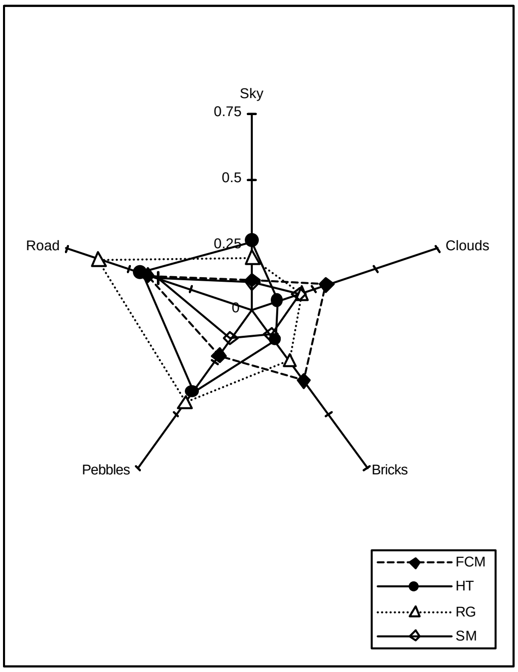
(*b*) CM



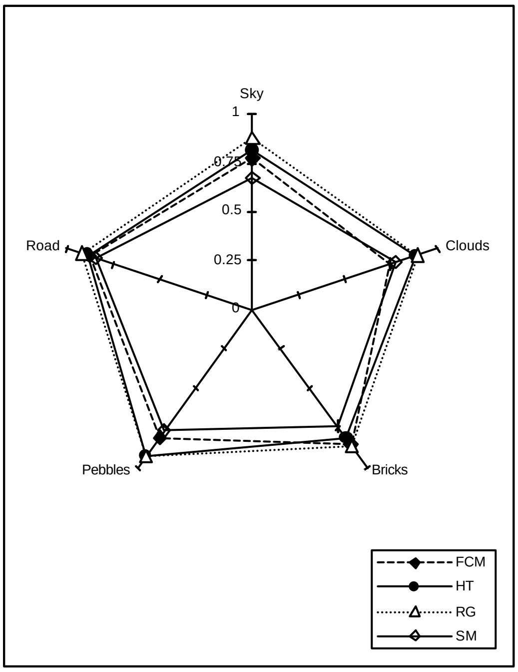
(*c*) EF



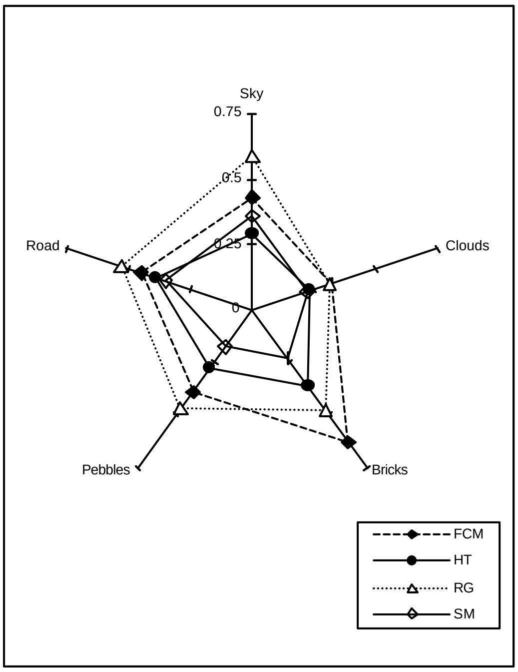
(*d*) LM



(*e*) RL



(*f*) TO



(*g*) TS

Figure 10

Autocorrelation	99 coefficients extracted using the equation $C_{ff}(p,q)=\frac{MN}{(M-p)(N-q)}\frac{\sum_{i=1}^{M-p}\sum_{j=1}^{N-q} f(i,j)f(i+p,j+q)}{\sum_{i=1}^{M}\sum_{j=1}^{N} f^{2}(i,j)}$ where $p=1...10$, $q=1...10$ is the positional difference in the i, j direction, and M, N are image dimensions.
Co-occurrence matrices	14 texture features including angular second moment, contrast, variance, inverse different moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, two information measures of correlation, and maximum correlation coefficient and six other measures on statistics of co-occurrence matrices.
Edge-frequency	$E(d)= \mid f(i,j)-f(i+d,j)\mid + \mid f(i,j)-f(i-d,j)\mid + \mid f(i,j)-f(i,j+d)\mid + f(i,j)-f(i,j-d)\mid$ for $1 \le d \le 50$
Law's mask features	The energy measure for a neighbourhood centred at *F (j, k)*, *S (j, k)*, is based on the neighbourhood standard deviation computed from the mean image amplitude: $S(j,k)=\frac{1}{W^{2}}\left[\sum_{m=-w}^{w}\sum_{n=-w}^{w}[F(j+m,k+n)-M(j+m,k+n)]^{2}\right]^{\frac{1}{2}}$ where W x W is the pixel neighbourhood and the mean image amplitude *M (j, k)* is defined as: $M(j,k)=\frac{1}{W^{2}}\sum_{m=-w}^{w}\sum_{n=-w}^{w}F(j+m,k+n)$
Run Length	Let B(a, r) be the number of primitives of all directions having length r and grey level a. Let A be the area of the region in question, let L be the number of grey level within that region and let $\boldsymbol{N_r}$ be the maximum primitive length within the image. The texture description features can then be determined as follows. Let K be the total number of runs: $K=\sum_{a=1}^{L}\sum_{r=1}^{Nr}B(a,r)$, then the features are given by: $f_1=\frac{1}{K}\sum_{a=1}^{L}\sum_{r=1}^{Nr}\frac{B(a,r)}{r^{2}}$, $f_2=\frac{1}{K}\sum_{a=1}^{L}\sum_{r=1}^{Nr}B(a,r)r^{2}$

	$f_3 = \frac{1}{K}\sum_{a=1}^{L}[\sum_{r=1}^{Nr}B(a,r)r^2]$, $f_4 = \frac{1}{K}\sum_{a=1}^{L}[\sum_{r=1}^{Nr}B(a,r)]^2$ $f_5 = \frac{K}{\sum_{a=1}^{L}\sum_{r-1}^{Nr}rB(a,r)} = \frac{K}{A}$
Texture Operators	Manian et al. [30] present a new algorithm for texture classification based on logical operators. These operators are based on order-2 elementary matrices whose building blocks are numbers 0, 1, and –1 and matrices of order 1x1. These matrices are operated on by operators such as row-wise join, column-wise join, etc. A total of six best operators are used and convolved with images to get texture features. Features are computed using zonal-filtering using zonal masks that are applied to the standard deviation matrix. Features obtained include horizontal and vertical slit features, ring feature, circular feature and sector feature.
Texture Spectrum	He and Wang [25] proposed the use of texture spectrum for extracting texture features. If an image can be considered to comprise of small texture units, then the frequency distribution of these texture units is a texture spectrum. The features extracted include black-white symmetry, geometric symmetry, degree of direction, orientation features and central symmetry.

Appendix 1: Description of various feature extraction methods

Study	Features	Key observations
Weszka et al.[57]	Fourier power spectrum, second order grey level joint co-occurrence statistics, grey level difference statistics, grey level run length statistics	Statistical features are better than Fourier features. Grey-level statistics gave better results for larger sizes and distances than statistics on single grey level.
Parikh[36]	Infra-red vs. texture features, texture vs. spectral features.	Spectral features are most important.
Conners and Harlow[10]	Co-occurrence matrices, run length, grey level difference method and power spectral method.	Co-occurrence matrices are best.
Van Gool et al.[52]	Co-occurrence matrices, grey level difference method, run length, Fourier power spectrum, autocorrelation, filter masks, random walk procedures, and methods based on texture models. Structural approaches including placement techniques and primitive extraction, and syntactic approach.	Provides an overview of other studies-survey.
Buf et al.[8]	Co-occurrence matrices, fractal dimension, Law's features, Hadamard mask, spectral decomposition, and grey level extrema method.	Haralick's method[20], Law's masks[45] and Hadamard masks gave the best results.
Ohanian and Dubes [32]	Markov Random Field, Gabor multi-channel parameters, fractal based features and co-occurrence features.	Co-occurrence features are best followed by fractal features.
Reed and Buf [39]	Feature based (Law's masks, co-occurrence matrices, Hadamard masks, Chen and Pavlidis[12] method, and transform domain method), model based (fractal models and stochastic models) and structural features (region based methods, boundary based methods and hybrid methods). Also spatial frequency methods (Gabor power spectrum and global power spectrum).	Co-occurrence based features are best as reported by Strand and Taxt[46].
Augustejin[2]	Co-occurrence matrices, grey-level differences, texture-tone analysis, Fourier transform features, Gabor filter features.	No universal best feature set. Fourier features in general were the best followed by co-occurrence matrices and Gabor features.
Ojala et al.[34]	Grey-level difference method, Law's texture method, center-symmetric covariance measures and local binary patterns.	Local binary patterns perform the best followed by histogram features. Covariance features performed better than Law's features.

Randen and Husøy[38]	Law's masks, ring/wedge filters, Gabor filter banks, wavelet transforms, wavelet packets and frames, Quadrature Mirror Filters (QMF), Discrete Cosine Transform (DCT), eigen filters, optimised Gabor filters, linear predictors, and optimised finite impulse response filters.	Different methods are found to perform better on different images. Law's method and ring/wedge filters are never winners or stand out as being very good. Poor performances are observed for Gabor filter and DCT. DCT has the least computational complexity. QMF and wavelet frame approaches are among the best for most images. Co-occurrence matrix features are worst.
Chen and Chen[11]	Fourier transform, spatial filter, Gabor filter and wavelet transform	Wavelet and Gabor features perform equally well but wavelet method is computationally less intensive.

Appendix 2. A summary of studies comparing texture features