Improved Unsupervised Clustering Over Watershed-Based Clustering

Sai Venu Gopal Lolla
School of Mechanical and Aerospace Engineering
Oklahoma State University
Stillwater OK 74078
venu.lolla@okstate.edu

Lawrence L. Hoberock
School of Mechanical and Aerospace Engineering
Oklahoma State University
Stillwater OK 74078
larry.hoberock@okstate.edu

***Abstract*—This paper improves upon an existing Watershed algorithm-based clustering method. The existing method uses an experimentally determined parameter to construct a density function. A better method for evaluating the cell/window size (used in the construction of the density function) is proposed, eliminating the need for arbitrary parameters. The algorithm has been tested on both published and unpublished synthetic data, and the results demonstrate that the proposed approach is able to accurately estimate the number of clusters present in the data.**

***Index Terms*—unsupervised clustering; watershed; scale;**

I. Introduction

Clustering or Cluster Analysis refers to the process of classifying data into meaningful homogeneous groups, and is a type of unsupervised classification. Clustering is a particularly difficult problem since the interpretation of the resulting clusters and their number depends upon domain-specific knowledge, practical experience, possible assumptions involved and human intuition [1]. An effective clustering method is often a well-balanced combination of data pre-processing methods, distance metrics, criterion functions, searching and sorting algorithms, and strategies to handle outliers and missing values [2].

Clustering algorithms are categorized into partitioning, hierarchical, density-based, grid-based, and model-based methods. Each method has its own set of advantages and disadvantages. Some work has also been devoted to combining several clustering methods into one algorithm [2]. There are two central issues that almost all clustering algorithms should address [3]: (1) Into how many clusters should the data be classified? and (2) How should data be classified, once the number of clusters has been decided? The former is considered to be a more difficult problem than the latter. Several clustering methods have been proposed that try to determine the "natural" / "optimum" number of clusters [1], [4]–[8]. To work properly, most algorithms require at least one parameter to be provided by the user, such as the number of clusters present in the dataset, or a parameter that in turn governs the number of clusters that can be detected. Selecting the "right" values for such parameters might be trivial in some cases, but it can often become impractical and infeasible due to the size and dimensionality of the data or other constraints.

TABLE I
"Correct" number of clusters for each dataset

Dataset	Cluster Count	Dataset	Cluster Count
S1	15	A3	50
S2	15	V1	12
S3	15	V2	9
S4	15	V3	13
A1	20	Z1	1
A2	35	Z2	49

The clustering algorithm proposed in this paper, and its predecessor [9], are grid-based methods, and also share some features with density-based methods. A major advantage of both these methods is that they do not require the user to provide parameters.

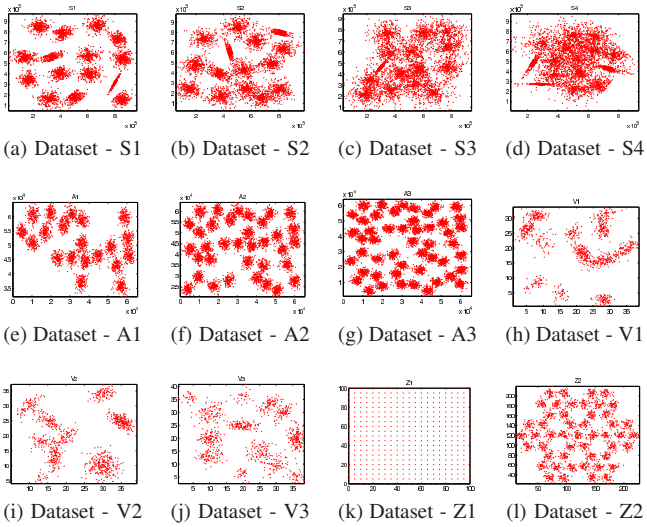
II. Datasets

A total of 12 synthetic datasets were used for the testing of the method proposed in this paper. While 7 of the datasets (S1-S4, A1-A3) were imported [10], the other 5 datasets (V1-V3, Z1-Z2) were created by the authors to test particular aspects of the proposed algorithm. Datasets S1-S4 have 15 clusters and 5000 points, each with various degrees of overlap. Datasets A1-A3 have varying numbers of data points and clusters [10]. Datasets V1-V3 have various degrees of overlap, and perhaps even multiple interpretations. Datasets Z1 and Z2 have interpretations that are scale-dependent. Visualizations of the datasets are provided in Fig.1a - 1l. Table I displays the "correct" number of clusters for each dataset, as determined by human inspection.

III. The Existing Method

The Watershed algorithm was developed from the fields of Image Processing and Mathematical Morphology, and is a region-based image segmentation method. The Watershed algorithm borrows its intuitive idea from geography - when a landscape or a topographic relief is flooded by water, water collects in *catchment regions*, and the catchment regions are divided by *watershed lines* [11].

The existing method [9] proposes that a grid be constructed over the feature space and then a density function be defined



(a) Dataset - S1 (b) Dataset - S2 (c) Dataset - S3 (d) Dataset - S4

(e) Dataset - A1 (f) Dataset - A2 (g) Dataset - A3 (h) Dataset - V1

(i) Dataset - V2 (j) Dataset - V3 (k) Dataset - Z1 (l) Dataset - Z2

Fig. 1. Datasets used for testing

over the grid. The density of each cell of the grid is treated as a height. Thus the density function takes on an interpretation of a landscape (3-D landscape for 2-D dataset). This landscape is then inverted and subjected to the Watershed algorithm. As a result of the Watershed algorithm, the minima in the inverted landscape, corresponding to the high density regions in the grid, are detected. Thus clusters are implicitly defined as regions of high density in the feature space, and are marked by corresponding catchment regions in the inverted landscape. The number of catchment regions found is taken to be the number of clusters present, and the catchment region itself represents the region spanned by the corresponding cluster. The formal representation according to Bicego et al. [9] is as follows. Let $\mathbf{Y} = \mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_n}$ represent the dataset where each observation is $\mathbf{y_i} = y_{i,1}, y_{i,2}, ..., y_{i,D}$. A grid with cells as *D*-dimensioned hypercubes of fixed size l_R is defined over the feature space with an origin O:

$$O = [\min_{i=1}^{n} y_{i,1}, \min_{i=1}^{n} y_{i,2}, ..., \min_{i=1}^{n} y_{i,D}] \tag{1}$$

A cell in the position $\mathbf{i} = (i_1, i_2, ..., i_D)$ is denoted as $R(\mathbf{i}) = R(i_1, i_2, ..., i_D)$. Once l_R is chosen (a choice which is critical and discussed later), a grid with $\left(\frac{k}{l_R}\right)^D$ cells spans the feature space, where k represents the maximum dimension width of the feature space, and is defined as follows:

$$k = \max_{j=1}^{D} \left(\max_{i=1}^{n} y_{i,j} - \min_{i=1}^{n} y_{i,j} \right) \tag{2}$$

The height function for the grid upon which the Watershed algorithm to operate, is then defined: the value of the function $I(R(\mathbf{i}))$ in a cell is the number of points that belong to that cell. The function $I(R(\mathbf{i}))$ is defined as:

$$I(R(\mathbf{i})) = \sum_{\mathbf{y_n} \in \mathbf{Y}} \chi_{R(\mathbf{i})}(\mathbf{y_n}) \tag{3}$$

where χ is the characteristic function of the set $R(\mathbf{i})$, and is defined as:

$$\chi_{R(\mathbf{i})}(\mathbf{y_n}) = \begin{cases} 1 & \text{if } \mathbf{y_n} \in R(\mathbf{i}); \\ 0 & \text{otherwise;} \end{cases} \tag{4}$$

The function $I(R(\mathbf{i}))$ (landscape) is an approximation of the density properties of the feature space, with an underlying assumption that similar points (points that are to be grouped into the same cluster) are near in the feature space (Assumption-1). The function values are inverted so that the local maxima mark the minima and vice-versa. The Watershed algorithm then marks the catchment regions present in the inverted landscape and thus the number of clusters is obtained.

Choosing the right value for l_R is critical for the aforementioned method to work meaningfully. Choosing too large a value results in coarse-segmentation, and too small a value will result in over-segmentation.

Bicego et al. [9] suggest that the value of l_R should be estimated for the data that is to be clustered. It is stated that a "good" value for l_R can be obtained by using the median of pairwise distances between all points. All distances $d(\mathbf{y_i}, \mathbf{y_j})[\forall i, j \in 1...n]$ are computed and then the median of all those distances is computed. From the data, the value for l_R is calculated by:

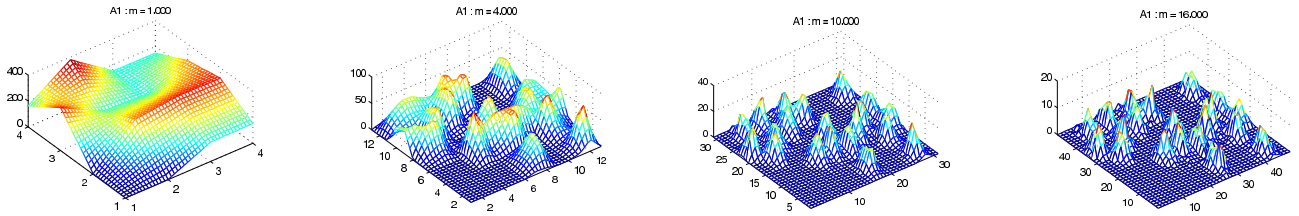
$$l_R = \frac{median(d(\mathbf{y_i}, \mathbf{y_j}))}{m} \tag{5}$$

In eq(5), m is a constant and is experimentally fixed at 4 for all the datasets evaluated by Bicego et al. [9]. It is suspected that this experimentally fixed value of $m = 4$ may not work well for all datasets. This is demonstrated with one of the datasets introduced in Section II. Fig.2a - 2d display the landscape for dataset A1 using $m = 1, m = 4, m = 10$ & $m = 16$ respectively. Fig.2a reveals that $m = 1$ produces too coarse a grid for dataset A1, and hence information about the number of clusters is lost during the construction of the landscape. Fig.2b shows that $m = 4$ produces a landscape that does not show 20 peaks corresponding to the clusters. Fig.2c shows that $m = 10$ produces a landscape that displays 20 peaks corresponding to the clusters. Fig.2d shows that $m = 16$ produces a landscape displaying many more than 20 peaks. Landscapes were constructed for the remaining datasets and the results told a similar tale, confirming the earlier suspicion that $m = 4$ does not work for all datasets. From Fig.2a-2d it can also be seen that the landscape has a rather "abrupt" and "angular" nature as opposed to a "continuous" and "smooth" one, even for the cases where the number of peaks may be correctly perceived.

Since m influences the size of the cell on the grid and thus the construction of the landscape, fixing the value of m as a constant for all datasets is a major drawback of the existing method. Bicego et al. [9] duly acknowledge that future investigations should target construction of the grid to improve the existing method.

IV. A New Proposed Method

In the method proposed here, the major emphasis is on selecting the most appropriate cell size, which will be shown



(a) Dataset - A1 - m = 1 (b) Dataset - A1 - m = 4 (c) Dataset - A1 - m = 10 (d) Dataset - A1 - m = 16

Fig. 2. Density landscapes generated using existing method

to produce the most significant contribution. The method proposed here closely follows the central theme of the method described in Section III, and can be viewed as an improved version.

It is widely known that data can display different structures at different scales [12]. The term "scale" as applied to a given dataset can be loosely interpreted as the size of the smallest spatial structure that can be perceived from the dataset. Any structures smaller than a given "scale" have been suppressed in the rendering of the data at that scale. Since clustering can be interpreted as a method for detecting structure present in the data, different structures may be detected at different scales. For example: at very large scales all the data will be treated as one cluster, and at very small scales each data point can be treated as a cluster. Meaningful structures, and thus meaningful clusters, will be perceived when operating at the "right" scale(s) for the data. Thus, clustering methods should consider scale to accurately detect the number of clusters while operating on a given dataset [6].

In the existing method described in Section III and the method proposed here, scale relates to the size of the cell, based upon which the grid is constructed. From here on, use of the term "scale" will loosely refer to the cell size used to construct the grid.

In order to handle scale, some sort of smoothing operations might need to be performed. It is also known that smoothing operations need to abide by certain scale-space axioms. The Gaussian kernel satisfies these axioms and hence is the kernel of choice for the work that follows. Use of other kernels for smoothing is possible [12].

A. Construction of Matrix Form for a dataset

To implement the smoothing operation using the Gaussian kernel, both the dataset as well as the Gaussian kernel should be transformed to matrix forms so that the convolution operation may be performed easily. To construct a matrix form for a given dataset, a grid is constructed over the feature space, in a fashion similar to that in Section III. With $\mathbf{Y}$ and $\mathbf{y_i}$ defined as in Section III, define d_{cw} as the cell size, given by:

$$d_{cw} = \frac{\min\limits_{\forall i,j \in 1...n} (d(\mathbf{y_i}, \mathbf{y_j}))}{(2+\epsilon)} \tag{6}$$

where $d(\mathbf{y_i}, \mathbf{y_j})$ represents the distance between $\mathbf{y_i}$ & $\mathbf{y_j}$, and ϵ is any positive real number. This is inspired from the Nyquist-Shannon sampling theorem [13], and $\epsilon \to 0$ marks the limiting condition specified by the theorem for no loss of information. This should result in a grid $\mathcal{M}_{w_1,w_2,...,w_D}$ with a size of w_i in the i^{th} dimension. $\mathcal{M}$ is a matrix representation of the dataset without any loss of information.

B. Gaussian Kernels for smoothing

Let K_t represent the sampled version of the Gaussian kernel G_t given by:

$$G_t(x) = \frac{1}{\sqrt{(2\pi)^{(D/2)}|\Sigma|}} e^{(\frac{1}{2}(-x'\Sigma^{-1}x))} \tag{7}$$

where Σ is the covariance matrix for the Gaussian kernel given as:

$$\Sigma = \begin{pmatrix} t_1 & 0 & . & . & 0 \\ 0 & t_2 & 0 & . & . \\ . & 0 & . & 0 & . \\ . & . & 0 & t_{D-1} & 0 \\ 0 & . & . & 0 & t_D \end{pmatrix} \tag{8}$$

where t_i is the standard deviation of the i^{th} dimension and $|\Sigma|$ is the determinant of the covariance matrix. When employing K_t for smoothing $\mathcal{M}$, any spatial structural detail whose size is less than t_i will be suppressed in the i^{th} dimension. Also when operating upon $\mathcal{M}$, t_i has two meaningful limits: (1) the smallest meaningful value t_i can assume is 1, since the matrix would not contain any information about structures whose size is less than a single cell; and (2) the largest meaningful value t_i can assume is w_i, since the matrix would not contain complete information about structures whose size is larger than the matrix itself.

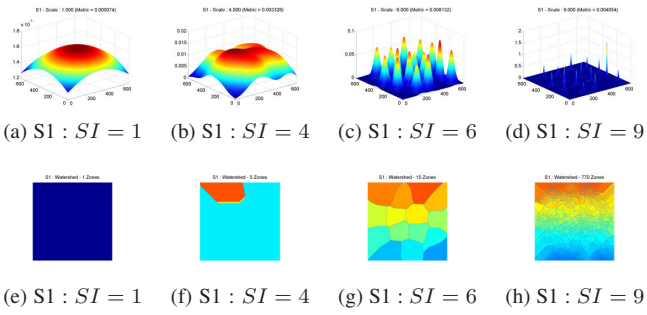
C. Generating Landscapes for a dataset

Consider *scale index (SI)*, related to t_i, by:

$$\begin{gathered} t_i = \frac{w_i}{2^{(SI-1)}} \\ 1 \leq SI \leq [1 + \log_2(\min(w_1, w_2, ..., w_D))] \end{gathered} \tag{9}$$

where w_i is the width of $\mathcal{M}$ in the i^{th} dimension, and t_i is the scale for the i^{th} dimension. SI is used to ensure that all dimensions in the features space (and hence all dimensions of $\mathcal{M}$) are given equal weight.

A landscape $\mathcal{L}_{SI}$ relating to scale index SI may be generated for a dataset by convolving the matrix representation of the dataset $\mathcal{M}$ with the sampled version of the D-dimensional



(a) S1 : $SI = 1$ (b) S1 : $SI = 4$ (c) S1 : $SI = 6$ (d) S1 : $SI = 9$

(e) S1 : $SI = 1$ (f) S1 : $SI = 4$ (g) S1 : $SI = 6$ (h) S1 : $SI = 9$

Fig. 3. Density landscapes (a)-(d) and corresponding watersheds (e)-(h) results generated for S1 using $SI = 1, 4, 6, \&9$

Gaussian kernel. Since the Gaussian kernel is separable:

$$\mathcal{L}_i = \begin{cases} \mathcal{M} * K_{t_i} & \text{if i=1;} \\ \mathcal{L}_{i-1} * K_{t_i} & \text{otherwise;} \end{cases} \tag{10}$$

where the convolution operation "$*$" is performed along the i^{th} dimension with K_{t_i} (the sampled version of the 1-dimensional Gaussian kernel with standard deviation t_i). The final result ($\mathcal{L}_{SI}$) is the matrix resulting after the convolution is performed along the D^{th} dimension ($\mathcal{L}_D$).

The landscape $\mathcal{L}_{SI}$ will have a structure that does not contain details smaller than t_i in the corresponding i^{th} dimension. The landscape also can be interpreted as a weighted density function. The weights are dictated by the values of the elements of the Gaussian kernel used in the process of convolution. The height of the mound at a particular point in the landscape is determined by the density of $\mathcal{M}$ at that point and the value of SI used to generate the the Gaussian kernel, and in turn the landscape. Fig.3a-3d display the landscapes for dataset S1 generated for $SI = 1, 4, 6\&9$. Fig.3a, 3b show only one large mound, indicating that at those scales, the data points can all be grouped into one cluster. Fig.3c ($SI = 6$) clearly show 15 smooth mounds indicating that at these scales, the data can be grouped into 15 clusters. Fig.3d has 15 dominating peaks, but also contain several other "noisy" spikes. Construction of landscapes for other datasets using for a range of values of SI resulted in similar observations.

Fig.3e - 3h display the results of subjecting the landscapes (Fig.3a-3d) to the Watershed algorithm. It can be seen from Fig.3e - 3f that the corresponding landscapes resulted in too coarse a clustering (coarse-segmentation). Fig.3g ($SI = 6$) shows that the corresponding landscapes resulted in the "correct" number of clusters. Fig.3h shows that the corresponding landscape resulted in too fine a clustering (over-segmentation). The problem now lies in selecting the "optimal" value of SI to construct a "best" landscape on which to execute the Watershed algorithm.

D. Selection of Optimal Scale Index

The selection of the optimal scale index SI_{opt} is critical, since it governs the scale at which the landscape is generated for the given dataset. If the scale is selected appropriately, it will result in a landscape that reflects the underlying cluster structure "well".

A heuristic from the observation of the density landscapes for a given dataset (and the corresponding watershed results) is: *Select the scale index which generates the landscape in which the base-width appears to be roughly proportional to mound-height for the majority of the mounds perceived in the landscape.* A quantitative version (or an approximation) of this heuristic would assist in automating the selection of the optimal scale index (SI_{opt}), and this is given in what follows.

Observation of histograms of "height" data (shall be referred to as $\mathcal{Z}$ from here on) in the landscapes for several values of SI hints that metrics of statistical dispersion should reach maxima for landscapes generated using the optimal scale index. Several measures of dispersion were computed for $\mathcal{Z}$ for various SI: Standard Deviation and Range (Fig.4a); Several indices of qualitative variation (MODVR, RANVR, AVDEV, MNDIF, VARNC, STDEV, and HREL) due to Wilcox [14] (Fig.4b); and Inter-Quartile Range (IQR) [15] and Median Absolute Deviation (MAD) [15] (Fig.4c). Only IQR & MAD reveal some clearly defined peaks. Since both IQR and MAD are robust measures of statistical dispersion, they are *outlier-resistant* [16]. The clearly defined peaks produced with IQR and MAD hinted that some sort of a filter should be applied to the height data $\mathcal{Z}$ before the variation indices are evaluated to make the variation indices less vulnerable to outliers in the height data. A technique developed by Tukey [15] is applied to the height data $\mathcal{Z}$. All values of $\mathcal{Z}$ in the range $[Z_{ll}, Z_{ul}]$ in eq(11) are accepted, and any values outside that range are filtered out during the evaluation of the variation index. Z_{ll} and Z_{ul} are defined as follows:

$$\begin{aligned} Z_{ll} &= Z_{25} - 1.5{\cdot}IQR \\ Z_{ul} &= Z_{75} + 1.5{\cdot}IQR \end{aligned} \tag{11}$$

where Z_{25} and Z_{75} are the 25 and 75 percentile values of $\mathcal{Z}$ respectively, and IQR is the Inter-Quartile Range of $\mathcal{Z}$.

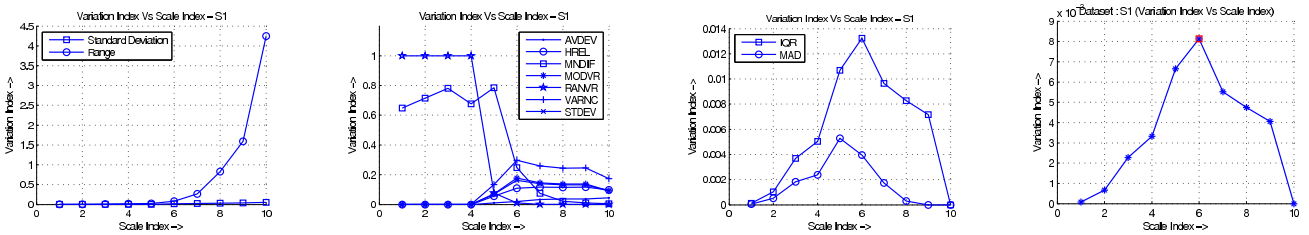
When Standard Deviation was computed for filtered $\mathcal{Z}$ for several values of SI, well defined peaks were observed. After the Tukey filter is applied to the height data $\mathcal{Z}$, the scale index SI that maximizes the Standard Deviation is selected as the optimal scale index SI_{opt}. An iterative search procedure is used to find the value of SI_{opt}. Fig.4d shows how the variation index changes with respect to the scale index, and this demonstrates the peaking we seek.

E. Detection of Clusters - Count & Location

Once the optimal scale index SI_{opt} has been found, the related optimal landscape ($\mathcal{LS}_{opt}$) is generated. Fig.5 shows the optimal landscape for dataset S1. This landscape is then inverted as described by:

$$\mathcal{LS}_{inv} = max(\mathcal{LS}_{opt}) - \mathcal{LS}_{opt} \tag{12}$$

The inverted landscape $\mathcal{LS}_{inv}$ can then be subject to the Watershed algorithm to detect the catchment regions. The number of catchment regions yields the number of clusters,



(a) Range & Standard Deviation

(b) MODVR, RANVR, AVDEV, MNDIF, VARNC, STDEV & HREL

(c) IQR & MAD

(d) Standard Deviation after Tukey Filtering

Fig. 4. Dispersion metrics evaluated for landscape generated using various values of SI (Dataset S1).

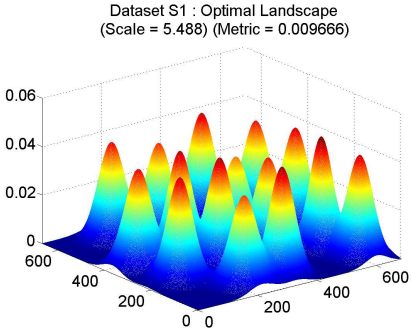


Fig. 5. Optimal Landscape for Dataset S1.

and the cluster centers may be evaluated using the data points present in each catchment region. Based on Assumption-1 from Section III, we assert that cluster centers must be regional maxima in $\mathcal{LS}_{opt}$. SI_{opt} indicates the minimum size of the structure that can be detected in the optimal landscape $\mathcal{LS}_{opt}$. Using this observation, if a peak on $\mathcal{LS}_{opt}$ has the maximum height in its neighborhood of size X, given by:

$$X = 2[t_1, t_2, t_3, ..., t_D] \tag{13}$$

centered at the peak, then that peak represents a cluster center. Eq(13) is constructed from a geometric interpretation for the condition to prevent overlapping clusters. Reflection shows that the aforementioned might result in spurious clusters being identified due to isolated data points in the dataset, far removed from all the "real" clusters. To avoid such spurious detections, only those maxima whose height is greater than the median of the height data $\mathcal{Z}$ are chosen for further processing. In other words, regional maxima whose height is less than the median of the height data $\mathcal{Z}$ will not be considered as clusters. We designate this as *median filtering*.

Accordingly, we can replace the Watershed algorithm with a computationally simpler regional maxima finding algorithm - we designate this as the *Regional Maxima Finding* approach. Thus the number of regional maxima in $\mathcal{LS}_{opt}$ is the number of clusters "present" in the dataset, and the cluster centers are given by projecting the maxima locations from $\mathcal{LS}_{opt}$ to $\mathcal{M}$ and in turn to the feature space spanned by the dataset $\mathbf{Y}$. This information (cluster count and cluster center locations) can then be used by any partioning algorithm such as the *k-means* algorithm to determine cluster labels for all the datapoints.

V. Experiments & Results

The method developed in Section IV was to be tested on the datasets introduced in Section II. The algorithm was coded in MATLAB, and numerical experiments quickly encountered computational difficulties.

The experiments were stalled by memory limitations while trying to compute $\mathcal{M}$ as described in Section IV-A. While the data could have been scaled down so as to overcome the memory limitations, what follows is an additional mechanism that may be used in conjunction with the proposed method to achieve a workable method, should similar limitations be encountered while applying the proposed method to other datasets.

A modified matrix representation $\mathcal{M}'$ is constructed instead of $\mathcal{M}$ and the rest of the algorithm proceeds as described earlier. $\mathcal{M}'$ is computed in a fashion similar to $\mathcal{M}$, but instead of using d_{cw} as described in (6), d'_{cw} is used, given by:

$$d'_{cw} = SF\frac{P_{cp}(d(\mathbf{y_i}, \mathbf{y_j}))}{2} \qquad 0 < SF < 1 \tag{14}$$

where $P_{cp}(d(\mathbf{y_i}, \mathbf{y_j}))$ is the cp percentile value for the pairwise distances $d(\mathbf{y_i}, \mathbf{y_j})(\forall i, j \in 1...n)$, and SF is a user selected *s*hrinkage factor. This definition will create $\mathcal{M}'$ with a much smaller memory requirement than $\mathcal{M}$. However, total preservation of structural information cannot be guaranteed. Structural details smaller than d'_{cw} will not be preserved. For all the tests conducted herein, cp was set at 1. It is contended that structural details with size less than 1 percentile of the pairwise distances in the dataset should not significantly affect the cluster structure. This contention can be easily verified visually. The contention is further verified if the modified definitions produce the "correct" result with the chosen value of cp and several values of SF.

We note that this modification is proposed only for cases where a computational limitation restricts the original method altogether. Should a computational system be available such that a given dataset can be processed without running into storage limitations, this modification is not needed.

TABLE II
RESULTS - REGIONAL MAXIMA WITH MEDIAN FILTERING

SF	S1	S2	S3	S4	A1	A2	A3	V1	V2	V3	Z1	Z2
0.400	15	15	15	15	20	35	50	**10**	9	13	**441**	49
0.425	15	15	15	15	20	35	50	**10**	9	13	**441**	49
0.450	15	15	15	15	20	35	50	**10**	9	13	**441**	49
0.475	15	15	15	15	20	35	50	**10**	9	13	**441**	49
0.500	15	15	15	15	20	35	50	**10**	9	13	**441**	49
0.525	15	15	15	15	20	35	50	**10**	9	**14**	**441**	49
0.550	15	15	15	15	20	35	50	**10**	9	13	1	49
0.575	15	15	15	15	20	35	50	**10**	9	13	1	49
0.600	15	15	15	15	20	35	50	**10**	9	13	1	49
0.625	15	15	15	15	20	35	50	**10**	9	13	1	**50**
0.650	15	15	15	15	20	35	50	**10**	9	**14**	1	49
0.675	15	15	15	15	20	35	50	**10**	9	13	1	**50**
0.700	15	15	15	15	20	35	50	**10**	9	13	1	49
0.725	15	15	15	15	20	35	50	**10**	9	**15**	1	**50**
0.750	15	15	15	15	20	35	50	**10**	9	13	**49**	49
0.775	15	15	15	15	20	35	50	**10**	9	13	**36**	49
0.800	15	15	15	15	20	35	50	**10**	9	**19**	**16**	49
0.825	15	15	15	15	20	35	50	**10**	9	13	**9**	**50**
0.850	15	15	15	15	20	35	50	**10**	9	13	1	49
0.875	15	15	15	15	20	35	50	**10**	9	13	1	**50**
0.900	15	15	15	15	20	35	50	**10**	9	13	1	49
0.925	15	15	15	15	20	35	50	**10**	9	13	1	49
0.950	15	15	15	15	20	35	50	**10**	9	13	1	**50**
0.975	15	15	15	15	20	35	50	**10**	9	**14**	1	49
1.000	15	15	15	15	20	35	50	**10**	9	13	**30**	**50**
	← Number of Clusters →											

- SF changes across rows and dataset changes across columns.
- Bold red entries indicate disagreement between actual cluster count and cluster count reported by the algorithm.

Table II displays the results obtained using the method proposed in Section IV in conjunction with the *Regional Maxima Finding* approach for detecting cluster count and cluster centers. Entries displayed in bold red in Table II indicate cases where there is disagreement between the actual cluster count (Table I) and the cluster count reported by the method proposed. Fig.6a through 6l display results of cluster detection (count & location) using the proposed method overlaid on the original datasets with $SF = 0.5$. A standard MATLAB implementation of the *k-means* algorithm was used to determine the cluster labels for the data points. Results for other values of SF are very similar.

Table II indicates that the method works well with S1-S4,A1-A3 & V2. The method consistently picks only 10 clusters for dataset V1 instead of the perceivable 11, which is likely due to the "low density" of the undetected cluster as compared to the other clusters in the dataset (See Fig.6h). Some experimental runs indicate an incorrect cluster count for dataset V3, and these are cases where median filtering fails to suppress the detection of spurious clusters. Fig.7a and 7b display two such cases. Dataset Z1 has a perfect square number of clusters in most cases. Fig.7c - 7d display such clustering results. It can be seen from these figures that these different perfect square counts are reported due to slight structural differences introduced during the construction of the modified matrix representation $\mathcal{M}'$ using different values for SF. The proposed method detects 49 clusters in dataset Z2 in most cases, but sometimes a cluster count of 50 is reported. It is suspected that this is also due to structural differences introduced during the construction of the modified matrix representation $\mathcal{M}'$ using different values for SF.

VI. CONCLUSIONS

In this paper, an improved approach toward Watershed-based clustering is presented. The improvements made are: (1) An automatic method is given for selecting cell size, based entirely on the data to be clustered itself, and eliminates the need for experimentally determined parameters; (2) The computationally intensive Watershed algorithm is replaced with a much simpler regional maxima finding process; and (3) An approach toward incorporating the concept of scale while generating landscapes is introduced. The main advantage of the proposed method is its unsupervised and automatic nature requiring no parameters to be tuned or to be determined experimentally.

The authors opine that future investigations should explore issues such as: (1) optimal selection of SF and cp (where the modified matrix representation $\mathcal{M}'$ needs to be constructed) to minimize loss of structural information; (2) construction of statistical dispersion metrics that could be used to locate optimal scale indices with improved fidelity; (3) methodologies to handle spurious clusters in cases where median filtering fails; (4) use of non-Gaussian kernels; and (5) further reduction of computational complexity.

REFERENCES

[1] G. Hamerly and C. Elkan, "Learning the k in k-means," in *In Neural Information Processing Systems*. MIT Press, 2003, p. 2003.

[2] S. B. Kotsiantis and P. E. Pintelas, "Recent advances in clustering: A brief survey," *WSEAS Transactions on Information Science and Applications*, vol. 1, pp. 73–81, 2004.

[3] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, vol. 18, no. 4, pp. 267–276, 1953. [Online]. Available: http://econpapers.repec.org/RePEc:spr:psycho:v:18:y:1953:i:4:p:267-276

[4] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data (SIGMOD '99)*. New York, NY, USA: ACM, 1999, pp. 49–60.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996, pp. 226–231.

[6] R. Kothari and D. Pitts, "On finding the number of clusters," *Pattern Recognition Letters*, vol. 20, no. 4, pp. 405–416, 1999.

[7] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001. [Online]. Available: http://dx.doi.org/10.1111/1467-9868.00293

[8] Q. Zhao, M. Xu, and P. Fränti, "Knee point detection on bayesian information criterion," in *ICTAI '08: Proceedings of the 2008 20th IEEE International Conference on Tools with Artificial Intelligence*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 431–438.

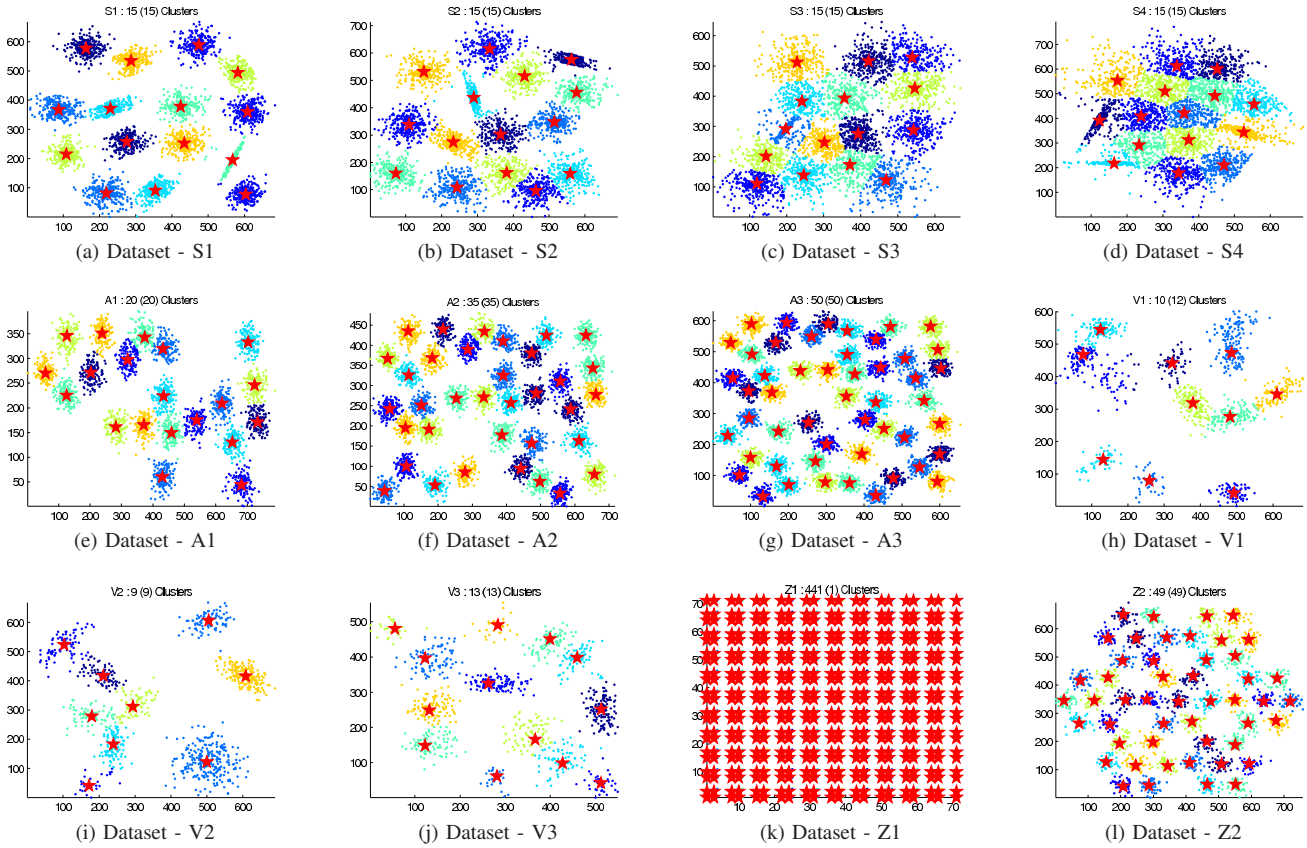
[9] M. Bicego, M. Cristani, A. Fusiello, and V. Murino, "Watershed-based unsupervised clustering," in *EMMCVPR*, ser. Lecture Notes in Computer Science, A. Rangarajan, M. A. T. Figueiredo, and J. Zerubia, Eds., vol. 2683. Springer, 2003, pp. 83–94.

[10] P. Fränti, "Clustering datasets," http://cs.joensuu.fi/sipu/datasets/, 2006 (accessed February 19, 2010).

[11] J. B. T. M. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms and parallelization strategies," *Fundamenta Infomaticae*, vol. 41, no. 1-2, pp. 187–228, 2000.

[12] T. Lindeberg, *Scale-Space Theory in Computer Vision*. Norwell, MA, USA: Kluwer Academic Publishers, 1994.

[13] C. E. Shannon, "Communication in the Presence of Noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1697831

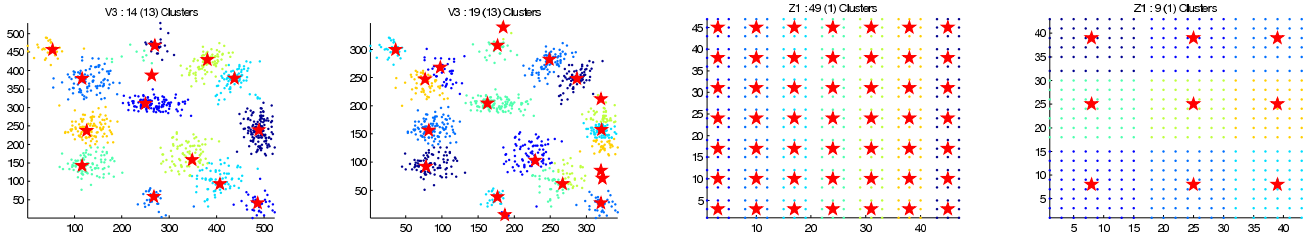


(a) Dataset - S1 (b) Dataset - S2 (c) Dataset - S3 (d) Dataset - S4

(e) Dataset - A1 (f) Dataset - A2 (g) Dataset - A3 (h) Dataset - V1

(i) Dataset - V2 (j) Dataset - V3 (k) Dataset - Z1 (l) Dataset - Z2

Fig. 6. Clustering results : $SF = 0.5$ (Red points indicate cluster centers)



(a) V3 : $SF = 0.525$ (b) V3 : $SF = 0.800$ (c) Z1 : $SF = 0.750$ (d) Z1 : $SF = 0.825$

Fig. 7. (a),(b) - Examples of *median filtering* failure; (c),(d) - Differences in clustering results due to induced structural differences (Red points indicate cluster centers).

[14] A. R. Wilcox, "Indices of qualitative variation," Oak Ridge National Laboratory, Tech. Rep. 3445605133753, 1967.

[15] D. C. Hoaglin, F. Mosteller, and J. W. Tukey, Eds., *Understanding Robust and Exploratory Data Analysis*, ser. Wiley Series in Probability and Mathematical Statistics. Wiley-Interscience, 1983.

[16] P. J. Huber, *Robust Statistics / Peter J. Huber*. Wiley, New York, 1981.