

# Unsupervised Clustering using Multi-Resolution Perceptual Grouping

Tanveer Syeda-Mahmood  
IBM Almaden Research Center  
650 Harry Road, San Jose, CA 95120  
stf@almaden.ibm.com

Fei Wang  
IBM Almaden Research Center  
650 Harry Road, San Jose, CA 95120  
wangfe@us.ibm.com

## Abstract

*Clustering is a common operation for data partitioning in many practical applications. Often, such data distributions exhibit higher level structures which are important for problem characterization, but are not explicitly discovered by existing clustering algorithms. In this paper, we introduce multi-resolution perceptual grouping as an approach to unsupervised clustering. Specifically, we use the perceptual grouping constraints of proximity, density, contiguity and orientation similarity. We apply these constraints in a multi-resolution fashion, to group sample points in high dimensional spaces into salient clusters. We present an extensive evaluation of the clustering algorithm against state-of-the-art supervised and unsupervised clustering methods on large datasets.*

## 1. Introduction

The problem of clustering a collection of data points occurs in a wide variety of practical applications ranging from data mining in web e-commerce [25], monitoring of metric streams in systems management [24], to people identification in computer vision [26]. The goal of clustering is to simplify the characterization of the data into semantically meaningful groups. In the absence of a priori knowledge, the nearest neighbor proximity constraint is fundamental to any clustering algorithm. This is based on the rationale that good feature representations map similar objects to be close together in a feature space. However, in many data distributions from applications, there is more structure in the data than can be captured by proximity constraints alone. Figure 1 shows several examples of such data distributions. For example, Figure 1a shows the distribution of 4000 time points from a pair of metric time series in systems monitoring. The data can be clearly seen to lie along three lines, a fact important in characterizing the functional dependency of these metrics, but difficult to discover in existing clustering methods based on nearest neighbor constraints alone. Similarly, Figure 1b shows a 4D dataset of optical flow vectors (2D location, magnitude, direction) of 15,000 samples drawn from a cardiac echo video sequence. Using proximity alone, it will be difficult to separate the heart valves from

the chamber walls, which could cause errors in computer-aided diagnosis of heart wall motion. Finally, Figure 1c shows a case where density rather than proximity determines the structure. A point C in the inner disc that lies on a line AB joining two points in the outer disc would be more proximal to A and B and yet belongs to a different group. Thus additional constraints may have to be used to discover such high-level structures during clustering.

In this paper, we introduce perceptual grouping, previously used for capturing structure in images, as an approach to unsupervised multidimensional clustering. Specifically, we model the feature space as a multidimensional image and use perceptual grouping constraints of proximity, density, contiguity, and orientation similarity, to group sample points into dense clusters. To accommodate perceptual groups at different scales, the grouping is performed at multiple image resolutions in a pyramidal fashion, with the top level of the pyramid representing the largest perceptual groups found in the multidimensional data.

The paper makes several novel contributions. Unlike previous approaches that have used conventional clustering for performing perceptual grouping, we introduce the opposite idea, i.e., perceptual grouping as a way of clustering. Again, unlike previous hierarchical clustering methods that indicate all data as grouped into one cluster at the top level, our algorithm converges to the stable number of perceived clusters at the top level. This gives an operational and efficient way of automatically determining the number of clusters in an unsupervised fashion. Finally, unlike other clustering algorithms, outliers are easily separated from the rest of the data distribution as isolated clusters making it applicable for noise removal in data distributions.

## 2. Related Work

Clustering is a well-researched field with algorithms available in data mining (database), machine learning, bioinformatics and other pattern recognition communities.

Algorithms are available to cluster the data in feature space derived directly from data (eg. Color histogram)[12] or from a model of the data (eg. AR model for image texture) [13]. The popular techniques include partitional (k-means, k-mediod see [5] for a review), overlapping (fuzzy c-means), hierarchical (agglomerative) [5], probabilistic (EM-based such as Mixture of Gaussians [17]), graph-theoretic (spectral clustering variants [16]), and scale-space [7] approaches. More recently, dimensionality reduction techniques have been combined with traditional clustering to exploit the simple structure in

the data when the data lies along low-dimensional manifolds [23], as well as methods for clustering the manifolds directly [14]. Finally, there are clustering algorithms that are use density in addition to proximity for clustering [4][8]. Many of these algorithms are supervised, that is, need hints on the number of clusters. The unsupervised clustering schemes are compute-intensive requiring repeated computations of distances over an  $N \times N$  distance matrices. Moreover, they

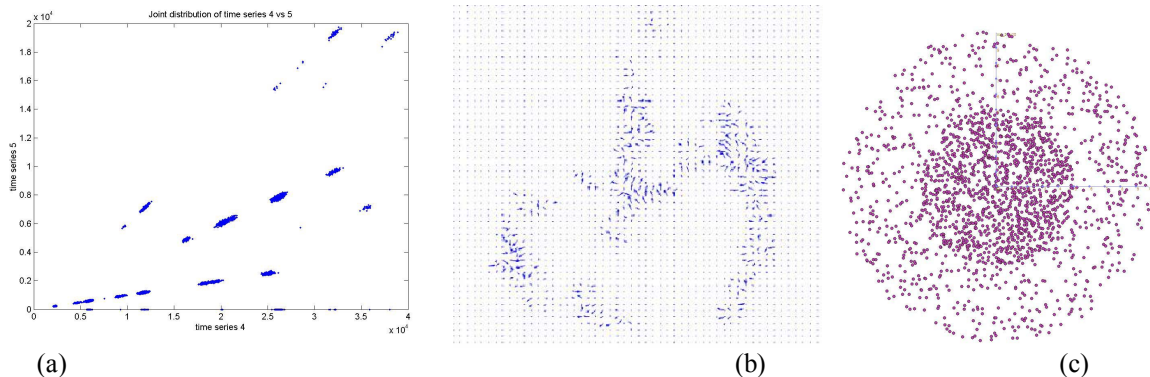


Figure 1. Illustration of the need to capture structure in the data distribution. (a) Data derived from time series metrics in systems monitoring. (b) 4D data set from (space+motion) cardiac echo videos. (c) Two disc discrimination dataset.

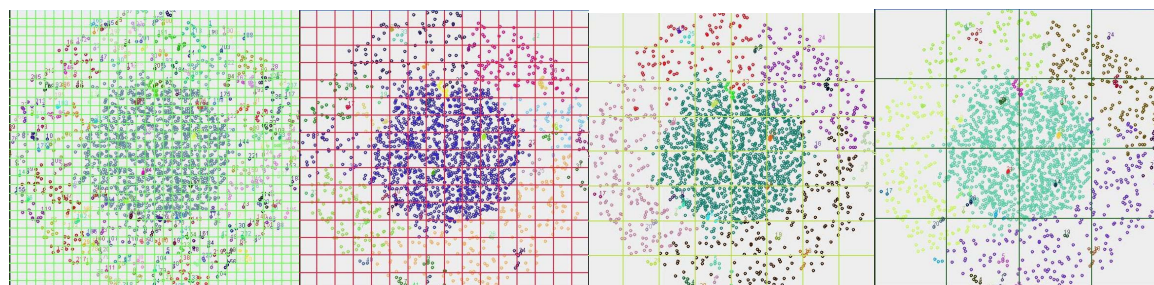


Figure 2. Illustration of unsupervised clustering using multi-scale grouping. The emergence of structure at different scales is shown here from left to right, top to bottom. Only 4 selected scales are shown due to space limitations.

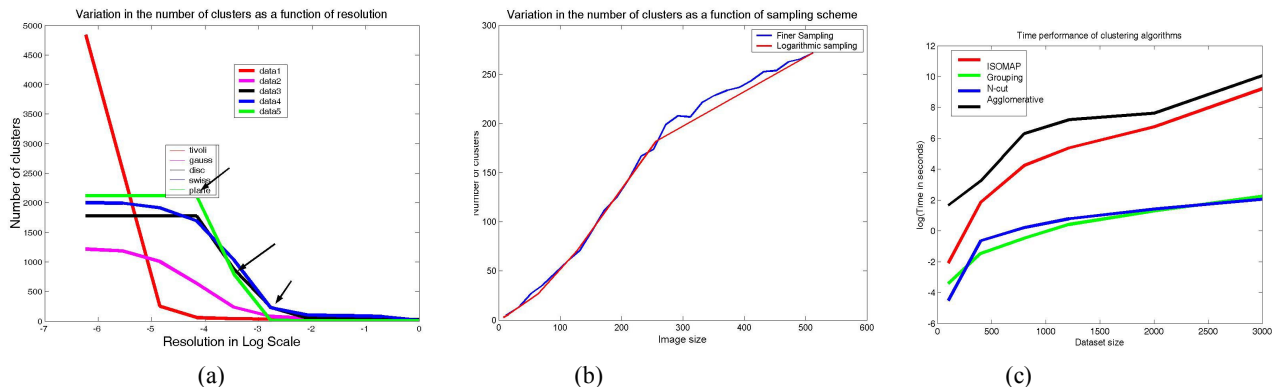


Figure 3: Assorted figures. (a) Convergence on the number of clusters in pyramidal grouping. (b) Effect of logarithmic versus regular sampling. (c) Time performance of different clustering algorithms

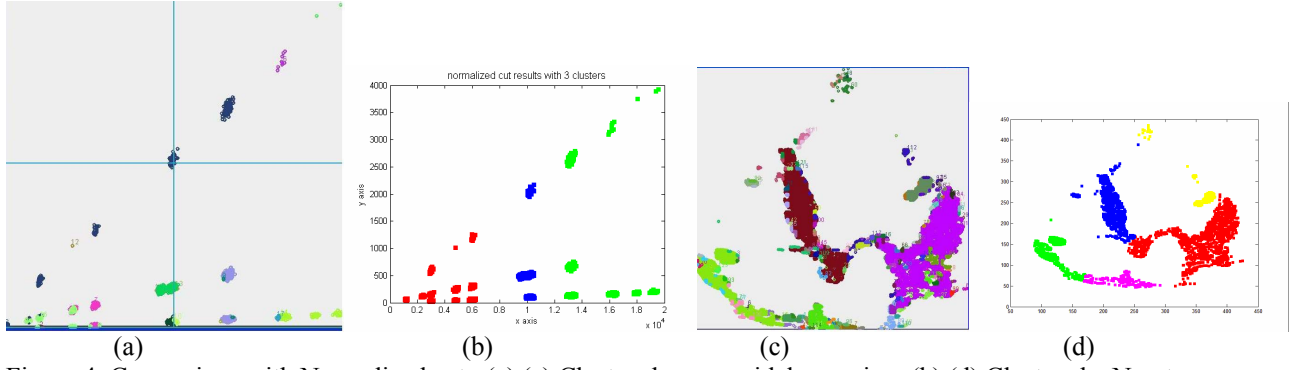


Figure 4. Comparison with Normalized cut. (a),(c) Clusters by pyramidal grouping. (b) (d) Clusters by N-cut.

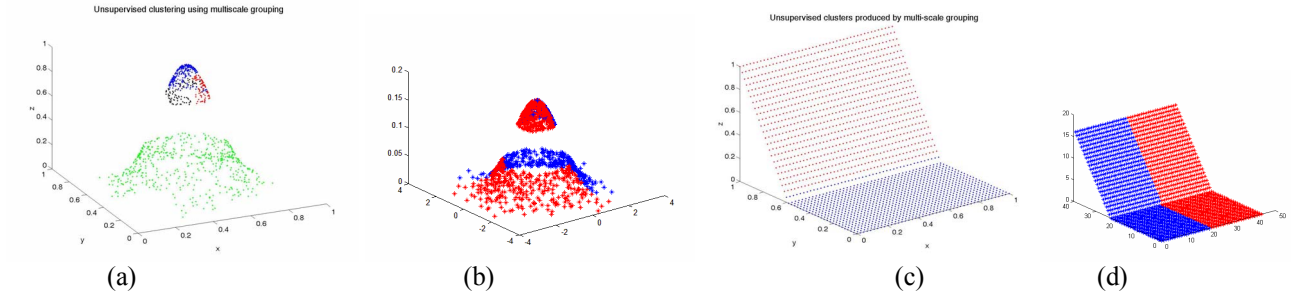


Figure 5. Comparison with ISOMAP. (a)(c) Clusters by pyramidal grouping. (b)(d) Clusters by ISOMAP.

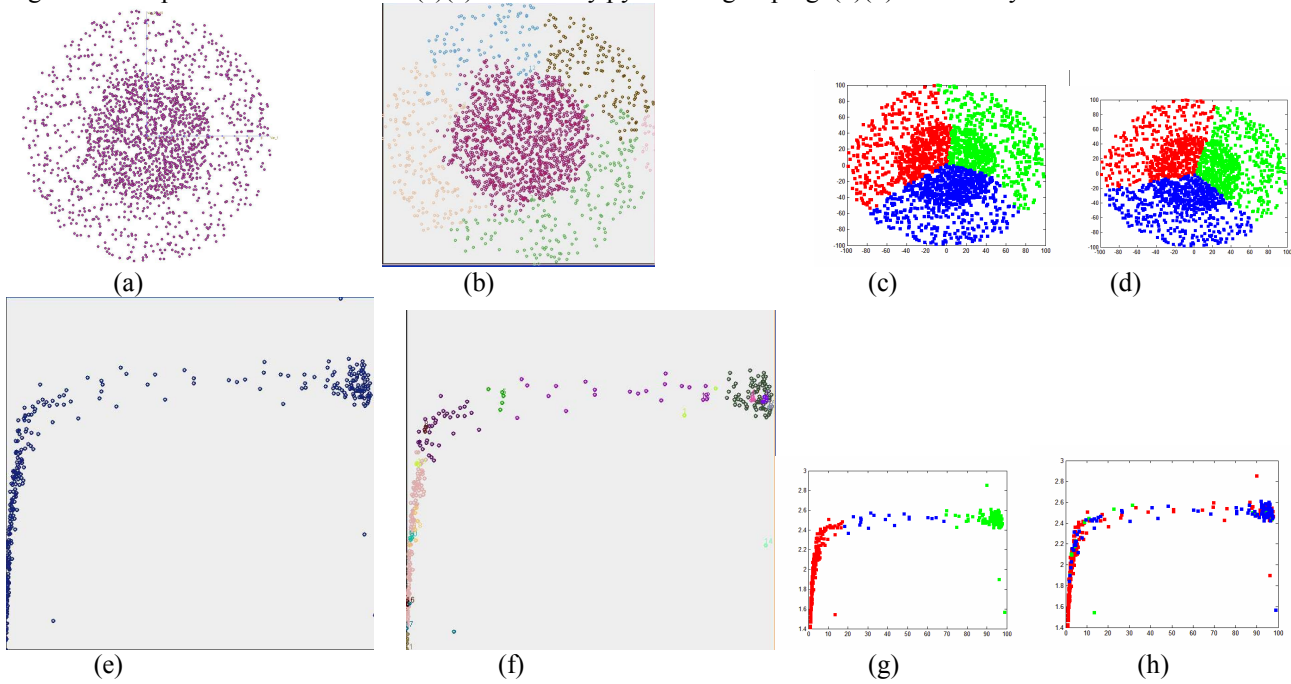


Figure 6. Illustration of the cross-data comparison between pyramidal grouping, Normalized cuts, and ISOMAP. (a)(e) Original data. Not the outliers in (e). (b)(f) Pyramidal grouping (c)(g) N-cut cluster (d)(h) ISOMAP clusters.

explore many more combination groupings of points than are practically plausible, and group all data into one at the top level. Of these, scale-space clustering is similar in spirit to ours, although it often overestimates both the centers of the clusters and the clusters themselves due to

the non-directional region blurring using isotropic kernel density functions [7].

### 3. Clustering by perceptual grouping

We now present our approach to unsupervised clustering using perceptual grouping. Perceptual grouping

refers to the human visual ability to extract significant image relations from lower-level primitive image features without any knowledge of the content, and to group them to obtain a meaningful higher-level structures [1][11]. A number of factors have been known to influence the parts of an image that are combined to form an object. These are based on the Gestalt principles of psychology [1]. Of these, we consider the principles of proximity, good continuation, connectivity, and density as also relevant for clustering. Local changes in dominant orientation of regions can signal important structural changes in the data distributions as can density changes. The contiguity constraint is needed to characterize region clouds from sparse point sets.

Due to sparse and irregular nature of point distribution, grouping has to be applied at multiple scales to see the emergence of high-level structures. Using the analogy of multi-resolution image pyramids [6][18], we sub-sample the feature space at multiple levels of resolution in a pyramidal fashion. At each sampling step, the perceptual grouping constraints of proximity, contiguity, orientation and density are applied to form candidate groups. While not all resolution levels produce a distinct change in the number of groups, the number of clusters is expected to monotonically decrease with the decrease in resolution levels, with salient bends emerging at levels where a new hierarchical structure emerges as shown by the arrows in Figure 3a for sample data distributions. We propose that these change points be used to signal the various levels in a hierarchical clustering, with the clusters at the top level emerging as the stable number of automatically determined clusters in the dataset.

We now describe the clustering algorithm in detail. It involves 3 main steps: (1) Determining a sampling scheme for the multi-dimensional feature space, (2) Using perceptual grouping constraints to assemble the clusters at successive pyramidal levels, and (3) Estimating the bends in the cluster curve to determine the clustering hierarchy, with the top level indicating the stable number of clusters perceived.

### 3.1. Terminology

We begin with some necessary terminology. Consider an M-dimensional data set of N samples  $X = (X_1, X_2, \dots, X_N)$  where each of the samples  $X_i = (f_{1i}, f_{2i}, \dots, f_{Mi})$  is an M-dimensional feature vector with features normalized so that  $0 \leq f_{ji} \leq 1.0$ . Each of the  $X_i$  is a point in a  $R^M$  space, which in turn, can be regarded as an M-dimensional image  $I_k$  of size  $L_k \times L_k \times \dots \times L_k$  at level  $k$ . Each sample then has an image coordinate

at level  $k$  that is an M-tuple  $D_i^k = [q_{1i}^k, \dots, q_{Mi}^k]^T$  where  $0 \leq q_{ji}^k \leq L_k - 1$  are the coordinates representing the pixel in the image. Each image pixel at level  $k$  is an M-dimensional unit of size  $(\frac{1}{L_k} \times \frac{1}{L_k})$ .

Using this model, clusters become multi-dimensional regions with image intensity formed from the cluster number, so that at each image size  $L_k$ , the sample  $X_i$  belonging to cluster  $c_i^k$  is represented by intensity  $c_i^k$  at the pixel  $D_i^k = [q_{1i}^k, \dots, q_{Mi}^k]^T$ . The set of clusters at each image size  $L_k$  is denoted by  $C^k = \{c_1^k, c_2^k, \dots, c_{n_k}^k\}$  with  $n_k$  as the distinct number of regions at image size  $L_k$ .

In the ensuing discussions, we refer to clusters interchangeably as regions, pixels as bins, scale as resolution, and feature space as image, respectively. In addition, we will refer to our clustering method as pyramidal grouping.

The variation in the number of clusters as a function of image size is given by a 1D-cluster curve  $z = \{(n_k, L_k) | k = 0, \dots, T\}$ . The bends in the curve, i.e., points where there is a significant change of curvature are denoted by  $z_p = (x(p), y(p))$ . These form the clusters in the clustering level  $C^p$  of the clustering hierarchy. Note that we distinguish between pyramidal image levels and hierarchical clustering levels, although for some data distributions, they may coincide.

### 3.2. Pyramid image sampling

Following the convention in pyramid image representations [6], we use a logarithmic sampling scheme. Since all the feature dimensions are normalized to be in the range  $[0, 1]$ , we can use a square grid. We start with an image of size  $L_0 \times L_0 \times \dots \times L_0$ . Each successive image is of size  $L_i \times L_i \times \dots \times L_i$  where  $L_i = L_{i-1}/2$  until we reach an image size of  $1 \times 1$ . Section 3.7 discusses the implications of logarithmic sampling over other sampling schemes.

### 3.3. Perceptual grouping constraints

We use four grouping constraints, namely, proximity, orientation similarity, and density similarity, and region contiguity.

#### 3.3.1 Proximity constraint

Using the multi-dimensional image model, a pair of points  $(X_i, X_j)$  are considered proximal at pyramidal level  $k$ , if

$$|D_i^k - D_j^k| \leq 1 \text{ or } |q_{li}^k - q_{lj}^k| \leq 1, \forall l \leq l \leq M, \quad (1)$$

Where the  $||$  operation stands for the absolute value.

Since at each pyramidal level, the grouping will consider the clusters from the previous level as the grouping elements, the proximity constraints states that they can be merged provided at least a pair of their respective image bins are adjacent. Let  $D^k(c_i) = \{D_{li}^k, \dots, D_{lj}^k\}$  be the set of image pixels at level

$k$  occupied by the cluster  $c_i^{k-1}$  (i.e. at least one of the sample points of the cluster belongs to one of these image bins). Then the proximity constraint to group two clusters  $c_i^{k-1}, c_j^{k-1}$  from level  $k-1$  into one at level  $k$  can be given as

$$\exists l, m \mid |D_{li}^k - D_{mj}^k| \leq 1 \quad (2)$$

### 3.3.2 Density constraint

Using the image sampling grid, the average density of a cluster at pyramid level  $k$  is obtained by noting the average number of samples with the given cluster label within an image pixel. Let  $N_c^k$  be the number of image pixels for cluster  $c$  at pyramid level  $k$  and  $n_{cl}^k$  be the number of sample points belonging to cluster  $c$  in bin  $l$  at pyramid level  $k$ . Then the average density of the cluster  $c$  is given by

$$\text{Density}(c^k) = \frac{\sum_{l=1}^{N_c^k} n_{cl}^k}{N_c^k} \quad (3)$$

The grouping constraint of density attempts to group clusters that have a small difference in density. That is, given two clusters from pyramid level  $k-1$ ,  $c_i^{k-1}, c_j^{k-1}$ , the density constraint is

$$|\text{Density}(c_i^{k-1}) - \text{Density}(c_j^{k-1})| \leq \tau \quad (4)$$

### 3.3.3 Orientation constraint

The orientation of the region is characterized by the eigenvectors of the covariance matrix. For simplicity, we only consider the eigenvector corresponding to the largest eigenvalue. Let  $v_i, v_j$  be the eigenvectors corresponding to the largest eigenvalues for two clusters  $c_i^{k-1}, c_j^{k-1}$  respectively. Then the two clusters are merged at the next pyramid level  $k$  if

$$\rho \leq \Theta(c_i^{k-1}, c_j^{k-1}) = |v_i \circ v_j| \leq 1.0, \quad (5)$$

where  $0 \leq \rho \leq 1.0$ . Here the  $\circ$  represents dot product between the two (unit) eigenvectors.

### 3.3.4 Contiguity constraint

Using only the three constraints above, it is possible to have physically implausible clusters consisting of intersecting regions belonging to different clusters, particularly at higher levels of the image pyramid. Unlike

other constraints, such contiguity check must be a three-way constraint, to check if two of the groups being merged potentially intersect with a third group already assembled. Determining region intersections of point clouds in  $n$ -dimensional spaces, in general, is a challenging problem in computational geometry. For purposes of perceptual grouping, we detect contiguity of two potential groups  $c_i^{k-1}$  and  $c_j^{k-1}$  if the potential minimum spanning tree

(MST) formed from their merger  $V_{ij}^k$  does not have an edge intersecting with the MST  $V_l^k$  of a group already formed at this level  $c_j^k$  or with  $V_m^{k-1}$  for the region  $c_m^{k-1}$  at previous scale. Notice that  $V_{ij}^k = V_i^{k-1} \cup V_j^{k-1} \cup \{E_{\min}\}$  where  $E_{\min} = \min\{E_{uv}, u \in c_i^{k-1}, v \in c_j^{k-1}\}$  and  $E_{uv}$  is the distance between the  $M$ -dimensional points  $u$  and  $v$  belonging to groups  $c_i^{k-1}$  and  $c_j^{k-1}$  respectively. The two groups  $c_i^{k-1}$  and  $c_j^{k-1}$  meet the contiguity constraint if

$$E_{\min} \triangleright V_l^{k'} \quad (6)$$

where  $k'=k$  or  $k-1$  (as the case may be) and  $\triangleright$  denotes no proper line segment intersection.

## 3.4. Salient bend extraction from cluster curve

The levels of the clustering hierarchy are determined by extracting the salient bends in the cluster curve. That is, the number of clusters at each pyramidal level are collected to form a curve. A line segment approximation to the curve is performed and corners thus extracted are retained as salient bends in the curve.

## 3.5. Unsupervised Clustering Algorithm

1. At level 0,  $c_i^0 = X_i$ ,  
 $\text{Density}(c_i) = 1.0$ , and  $\Theta(c_i^0, c_j^0) = 1.0$ , and  
 $D^0(c_i) = \{D_i^0\}$ .
2. For  $k=1$  to  $\log L_0$  do
  - a. Let  $n_k$  be the number of clusters at pyramid level  $k$ . Let them be in their own components.
  - b. Merge clusters  $c_i^{k-1}$  and  $c_j^{k-1}$  if all of the conditions specified in Equations (2), (4), (5), and (7) are true.
3. Find salient bends in the curve of  $n_k$  vs.  $L_k$  as described in Section 3.4. Let the bends be at positions  $\{(L_{t_1}, n_{t_1}), \dots, (L_{t_s}, n_{t_s})\}$ . This gives the levels of the hierarchy as well as the clusters at such levels.

The clusters at the top level are the automatically determined clusters.

### 3.6. Complexity Analysis

The connected component generation is made efficient using the Union-Find data structure for merging taking  $O(1)$  amortized time per merge. The number of iterations is  $O(\text{Log } L_0)$ . The eigenvector computations take  $O(n^2)$  for a region of size  $n$ . Since the number of regions  $n_k$  rapidly decrease with scale (See Figure 3a), the overall cost of these computations remains low. The minimum spanning tree computations are also of the  $O(n^2)$  per iteration, which is much less than doing the MST on the entire graph taking  $O(N^4)$  as in graph-based clustering since  $N = \sum n$ . In practice, the region intersection test can be made much faster using a simple intersection of line joining centroids and extremal points of regions, which are already available from the eigenvector computations. Thus the overall complexity of the algorithm is  $O(n^2 n_k \text{Log } L_0) = O(N^2)$ .

### 3.7. Choice of parameters

The algorithm has three parameters, namely,  $L_0, \tau, \rho$ . The starting scale  $L_0$  for the pyramidal sampling is taken as the minimum distance separating the samples when projected along each dimension. This is a lower bound on the minimum separating distance between points but can be computed much faster without pair-wise distance comparisons. In practice, we found a separating distance of  $0.00195 = 1/512$  to be sufficient, thus making  $L_0$  independent of the dataset for clustering. The thresholds for the density perception and orientation perception are also independent of the dataset. They were chosen based on recommendations of earlier researchers in cartography that performed psychophysical studies on the perception of

spatial dispersion on point distributions [2][3]. There it was observed that salient regions could be distinguished based on a density difference by as much as 30% from their surroundings. Similarly, the orientation difference over 5 degrees was taken as sufficient distinction between the orientation of regions.

Since each cluster is successively refined, it may be questioned that the logarithmic sampling is too coarse, and that some of the clusters can be missed due to the coarse sampling. In general, if we take a histogram of the nearest neighbor distances, the distribution is dense for low values of the distance. Since we desire to minimize the intra-cluster distance while maximizing inter-cluster distance, the lower distance values must be sampled more finely than the larger distance values. Thus logarithmic sampling which samples the lower distances finely ( $1/2, 1/4, 1/8, 1/16, 1/32, \dots$ ) can give a good approximation to higher frequency distance sampling. This is also observed from our experiments as shown in Figure 3b which shows the difference in the number of clusters based on uniform and logarithmic sampling. Here, the cluster curve for logarithmic sampling appears to be a smoother version of the cluster curve formed from uniform sampling. Thus the salient bends in the cluster curve can still be estimated from a logarithmic sampling. It may be noted that other sampling schemes could be substituted into the clustering algorithm without fundamentally changing the result.

## 4. Results

We performed a number of experiments to assess the performance of the algorithm. The datasets were assembled from a variety of sources including benchmark 2D and 3D datasets such as the Sequoia benchmark 2000 and the sonar, liver and diabetes datasets from UCI Machine

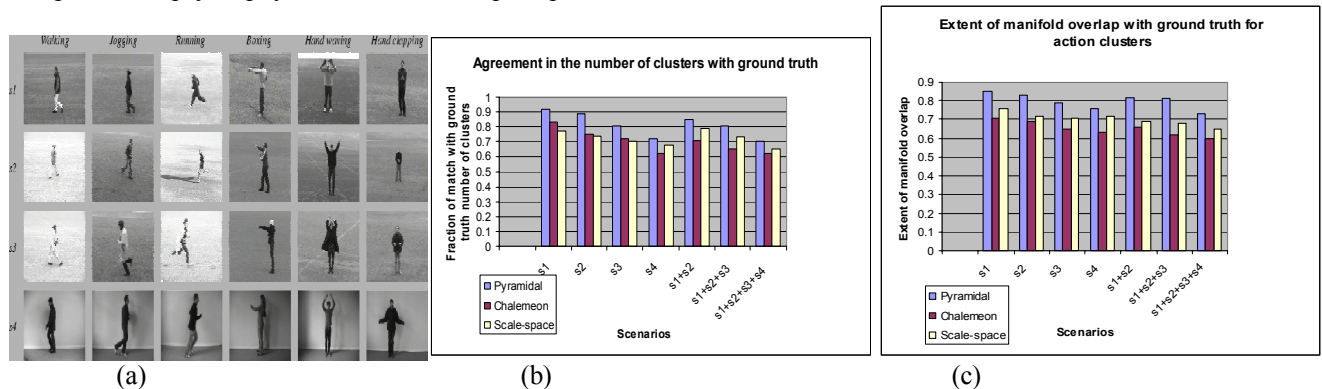


Figure 7. Illustration of performance of unsupervised clustering against a ground truth data set of action videos.

Learning Repository [21], the manifold learning datasets (see Figure 4-5), 10-dimensional time series metrics from systems monitoring scenarios, 4D optical flow maps from

cardiac echo videos, and 2D average velocity vector maps from activity videos[22]. The number of samples in the data set varied from a minimum of 20 samples to a

maximum of 40,500 samples. The number of dimensions varied from 2 to 200 (for the action videos).

#### 4.1. An Example

We first illustrate the operation of the clustering algorithm by an example. Figure 2 illustrates clustering by multi-scale grouping on a 2000 point dataset shown in Figure 1c at some of the scale levels. As can be seen, increasingly large groups are being formed at successive scales. The inner disc is cleanly separated from the outer disc matching with our perception in this case.

#### 4.2. Comparison with supervised clustering

Next, we compared pyramidal clustering to two state-of-the-art supervised clustering algorithms on the ability to extract perceptual structure from the data. Specifically, we used a reference implementation of ISOMAP [23] and Normalized cuts [16]. We ran the three clustering algorithms on the same datasets but gave hints on the correct number of clusters perceived to both ISOMAP and Normalized Cuts, whereas our algorithm ran unsupervised. The results can be seen in Figures 4, 5, and 6. First, we note that our algorithm overestimates the number of perceived clusters (some noise is still present). However, the groups produced are perceptually plausible regions as shown in these figures. In Figure 4, we show the clusters produced by pyramidal clustering and normalized cuts on the datasets shown in Figure 1 a and b. The line grouping errors are clearly evident from Figure 4b for normalized cuts. Similarly, the segmentation errors of valves with ventricular walls are more evident for normalized cuts in Figure 4d. Next, comparing to the ISOMAP algorithm in Figure 5, we also observe the structural grouping errors in both the Split Gaussian and 3D planes datasets (here the planes are split vertically). Finally, Figure 6 shows all three algorithms for two other datasets. Of the three algorithms, only pyramidal grouping could separate the inner disc from the outer disc of Figure 1c. Further, only it could isolate the noisy outliers at the edges of the dataset of Figure 6a as separate clusters while the other two merged them into one of the larger clusters.

Next, we recorded the running time of the three algorithms as a function of the size of the dataset on a T1300 Pentium 4, 1.6GHz, 1GB RAM machine. The performance of these algorithms is indicated in Figure 3c on a logarithmic time scale. It can be observed that ISOMAP runs much slower than N-cut for the dataset tested. For larger than 3000 samples, both ISOMAP and Normalized Cut had memory problems and often crashed. From this we conclude that pyramidal grouping is computationally as efficient as normalized cuts while still being unsupervised in automatically determining the number of clusters.

#### 4.3. Clustering performance

To assess the performance of pyramidal grouping in automatically indicating the number of clusters, we used the action video dataset provided by KTH [22]. This collection depicts six types of human actions as shown in Figure 7a, (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in four different scenarios: outdoors s1, outdoors with scale variation s2, outdoors with different clothes s3 and indoors s4. Currently the database contains 2391 sequences at 25fps frame rate each of which is 4 seconds long. We processed each video to extract the moving objects and their motion was described using average velocity curves using the method in [20]. This gave rise to a  $2 \times 4 \times 25 = 200$  dimensional feature space for the 1200 videos. Of the 2391 sequences available from the KTH database, we retained 1200 sequences for our study. Unsupervised clustering was performed to generate a hierarchy of clusters for the 1200 training videos.

To evaluate the correctness of clustering, we selected combinations of scenarios at a time such as s1 alone, s2 alone, s1+s2, s1+s2+s3, s1+s2+s3+s4, etc. and collected all videos of individuals depicting the 6 different action types. The ground truth action type was used to label the 1200 sample data sets in groups of scenarios mentioned above.

We measured the correctness of clustering by observing two parameters, namely, the match in the number of detected clusters, and the overlap of the cluster regions. The match in the number of detected clusters was given by the fraction of ground truth clusters that were indicated by the top level of multi-scale grouping. The percentage overlap of the cluster manifolds was taken as the difference in the fraction of grid occupied by the ground truth versus the clusters given by pyramidal clustering.

The data from each action type was averaged to generate the summary results in Figure 7. Figure 7b shows the agreement of the pyramidal clustering with the ground truth data in the clustering of actions. In all cases, we noted at least 70% agreement in the number of clusters by pyramidal clustering and the ground truth clusters, although the number of clusters was usually higher with pyramidal clustering. More surprisingly, there was a large region overlap of the pyramidal clusters with ground truth clusters. This indicates that the seed regions suggested by pyramidal clustering closely agreed with the ground truth perceptual grouping of the actions. As expected, the agreement worsened as the scenarios became mixed such as s1+s2+s3+s4 over individual scenarios s1. The scenario of s4 depicting indoor situations was the most difficult due to the existence of large background clutter.

#### 4.4. Comparison with unsupervised clustering

Finally, we compared pyramidal clustering to two other hierarchical clustering schemes namely, Chameleon [4]

and scale-space clustering [7]. These two methods were chosen as they are both hierarchical, and perform a hard clustering of data. Further, they agree with pyramidal clustering in the use of proximity and/or density cues, although the orientation cue is not used. We cut the clustering dendrogram at points indicated by the two algorithms in their papers [4][7] to make the comparison meaningful. The fraction of overlap with the number of clusters and the area overlap with the ground truth clusters is shown in Figure 7b and 7c. As can be seen, pyramidal clustering shows better agreement in both the number and the area overlap with the ground truth clusters. Furthermore, the time performance of our algorithm was at least orders of magnitude faster as shown in Figure 3c.

## 5. Conclusions

In this paper, we have presented a novel algorithm for unsupervised clustering of multi-dimensional data using the principles of multi-resolution perceptual grouping. We have shown that the clustering algorithm is successful at automatically determining the number of clusters and results in groups that match our perception. Extensive comparisons with state-of-the-art clustering algorithms have shown the superiority of this method in extracting higher-level structures in the data.

## References

- [1] Y. Gdalyahu, D. Weinshall, and M. Weman. Self-organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization, *IEEE Trans. PAMI*, vol. 23. no.10, pp.1053-1074, 2001.
- [2] Y. Sadahiro. Perception of Spatial Dispersion in Point Distributions, *Cartography and Geographic Information Science*, 27 (1), 51-64, 2000.
- [3] Y. Sadahiro. Cluster Perception in the Distribution of Point Objects, *Cartographica*, 34 (1), 49-61, 1997.
- [4] G. Karypis, E-H. Han, and V. Kumar. Chameleon: A hierarchical clustering algorithm using dynamic modeling, in *IEEE Computer*, vol. 32, no. 8, pp. 68-75, 1999.
- [5] A. K. Jain, M.N. Murthy, and P.J. Flynn. Data clustering: A survey. *ACM Computing Survey*, 31(3):264--323, 1999.
- [6] P.J. Burt, and E. Adelson. The Laplacian Pyramid as a compact image code, in *IEEE Transactions on Communications*, vol.31, no. 4, pp.532-540, 1983.
- [7] Y. Leung, J-S. Zhang, Z-B. Xu, Clustering by scale-space filtering, in *IEEE Trans. PAMI*, vol.22, no.12, pp.1396-1410, 2000.
- [8] M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. Intl. Conf. on Knowledge discovery and data mining (KDD-96)*.
- [9] M-L. Shyu, S-C. Chen, M. Chen, and C. Zhang. A Unified Framework for Image Database Clustering and Content-based Retrieval, *Proc. 2nd ACM international workshop on Multimedia databases*, pp. 19 - 27 2004.
- [10] J.C. Platt, M. Czerwinski, and B. A. Field. PhotoTOC: Automatic clustering for browsing personal photographs, in *Proc. Fourth IEEE Pacific Rim Conference on Multimedia*, 2003.
- [11] D.W. Jacobs. Robust and Efficient Detection of Salient Convex Groups, in *IEEE Trans. PAMI*, vol. (18) No. 1, January 1996, pp. 23-37.
- [12] J. Wang, W.J. Yang, and R. Acharya. Color clustering techniques for color content-based image retrieval from image databases. *Proc. Int'l Conference on Multimedia Computing and Systems*, 442-449, 1997.
- [13] L. Klimek, B. Wooley, S. Bridges, J. Hodges, A. Watkins, and S. Smolensky. Comparison of the performances of a Bayesian algorithm and a Kohonen map for clustering texture data, vol.9, pp.777-784, 1999.
- [14] R. Souvenir, and R. Pless. Manifold clustering, in *Proc. 10th International Conference on Computer Vision (ICCV)*, pp.648-653, 2005.
- [15] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm, *Jl. Of Royal Statistics Society*, vol.39, no.1, pp.1-21, 1977.
- [16] S.X. Yu, and J. Shi. Multiclass spectral clustering, in *Intl. Conf. on Computer Vision*, 2003.
- [17] L. Ye, and M. E. Spetsakis. Clustering on Unobserved Data using Mixture of Gaussians, *Technical Report CS-2003-08*, York University, Oct. 2003.
- [18] D. Heeger, and J. Bergen. Pyramid-based Texture Analysis/Synthesis. *Proceedings, ACM Siggraph*, August, 1995.
- [19] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*, MIT Press: Cambridge, MA 1990.
- [20] T. Syeda-Mahmood. Segmenting actions in velocity curve space, in *Proc. ICPR*. Vol. 4, pp. 170- 175, 2002.
- [21] <http://www.ics.uci.edu/~mlern/MLRepository.html>
- [22] <http://www.nada.kth.se/cvap/actions/>
- [23] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, vol. 290, pp. 2319--2323, 2000.
- [24] Y. Zhu, and D. Shasha. StatStream: Statistical monitoring of thousands of data streams in real time. In *Proc. VLDB*, pp.358-369, 2002.
- [25] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications, in *Proc. ACM SIGMOD, Intl. Conf.on Management of Data*, pp.94-105, 1998.
- [26] B. Raytchev, and H. Murase. Unsupervised recognition of multi-view face sequences based on pairwise clustering with attraction and repulsion, *Computer Vision and Image Understanding (CVIU)*, pp.22-52, 2003.