Shape-based Clustering Of Enterprise CAD Databases

Tathagata Chakraborty

Geometric Software Sol. Co. Ltd., tathagatac@geometricsoftware.com

ABSTRACT

Cluster analysis is a primary data mining method for knowledge discovery in spatial databases, where, the goal is to find 'natural' groups in a dataset based on a similarity or dissimilarity function for pairs of objects. With the number and size of spatial databases in various domains growing rapidly over the last couple of decades, methods for automated knowledge discovery in these datasets is becoming increasingly important. In the last couple of years, various similarity search methods for 3D CAD databases have been researched with the purpose of promoting engineering data reuse. However, unlike in other spatial domains, not much interest has yet been generated towards the task of automated knowledge discovery and data mining in 3D CAD databases. Moreover, most well-known clustering algorithms, have a very high computational complexity when used directly, and hence are too inefficient when applied to large spatial databases. Developing an efficient clustering technique usually requires leveraging crucial domain knowledge. This paper proposes a simple and efficient system for automatic single scan clustering of 3D CAD databases based on a shape similarity measure. The resulting system is a visual data mining tool to help the user quickly locate a seed model and search for similar models in the database. The paper also shows how the system can generate various statistical data which give new insight into the contents of the databases. The described system was implemented and evaluated on a large number of industrial CAD datasets and the results obtained were highly encouraging.

Keywords: similarity search, knowledge discovery, visual data mining, clustering

1. INTRODUCTION

The ideation, design and archival of 3D CAD engineering products is an expensive and complex task. Over the past decade there has been a rapid proliferation of 3D CAD models. This has been mainly due to automotive, aerospace and other large mechanical product manufacturing establishments adopting 3D modeling software for design, manufacturing and analysis of their products. At present, most of the 3D CAD and associated engineering data resides in PDM (Product Data Management) systems, which essentially provide methods for safe archival and controlled access to enterprise data. However, there are no efficient methods to organize this massive data in PDM systems in order to make it amenable to data mining and reuse. This is partly because PDM systems inherently do not have utilities to mine for reusable knowledge. These systems currently use metadata items as search criteria. Users can search for the existence, location and status of particular information on the basis of pre-defined classification codes and characteristics such as physical properties, manufacturing processes, and part numbers. PDM systems also use such metadata items for information browsing and navigation. The choice of metadata however is dictated by the needs of a particular business model, and is based on a multitude of industry standard, software and vendor specific keywords. Moreover, these metadata are often incomplete and difficult to update and manage for large databases.

With the increasing popularity of PLM (Product Lifecycle Management), there has been an increasing interest in integrating PDM systems, which store the bulk of the product data, with other smaller systems which help organize and discover knowledge in this data. Of these systems the one that has attracted exceptional interest in the last couple of years has been the field of similarity search in 3D CAD databases [1], [5], [9], [11], [12]. While similarity search systems help an engineer or a designer find similar models of interest, they do require the user to initially locate or build a seed model to be used as query to these systems. This entails a significant amount of time and effort on part of the user. Similarity search systems alone therefore do not provide for methods to actually mine for hidden data and discover new knowledge in existing databases. These systems can however form the basis for various cluster analysis methods, which can be used to organize large datasets.

Most well-known clustering methods have a high computational complexity, and hence cannot be directly applied to industrial CAD datasets. Moreover as these datasets change patterns over time, constantly updating the clusters becomes an expensive task. This paper proposes a simple and efficient system of clustering databases of CAD models, which is based upon the shape similarity search method proposed in [6]. The efficiency of the system results partly from the nature of the hierarchical signatures generated by the said similarity search method, which considerably reduces the search complexity. The system can also update the existing clusters efficiently when the database is updated due to additions or deletions. It is shown that clustering of enterprise CAD databases gives new insight into the nature and structure of the archived data, by generating various statistical results. Furthermore, visually analyzing these clusters and interconnections between them helps the designer or engineer quickly locate desired models. Also, as the user's familiarity with the system grows, so does his ability to quickly locate and identify potentially reusable models and associated engineering data.

The remainder of this paper is organized as follows: section 2 reviews previous research on clustering methods that have been used to cluster spatial databases in CAD and other domains. Section 3 describes in detail the basis for the non-hierarchical clustering method that is used by the described system. Section 4 describes a simple single scan algorithm used to cluster CAD databases. Section 5 describes the industrial prototype that implements the described similarity search and clustering algorithms, and show various results. Section 6 contains conclusions and avenues for future research.

2. RELATED WORK

Clustering is the task of grouping the objects of a database into meaningful subclasses or groups. A number of general clustering techniques for spatial databases have been developed over the past few decades, which have been widely used in areas such as pattern recognition, data analysis and image processing. Unsupervised learning or clustering is especially useful when the database size is huge, as is the case for most manufacturing enterprises and more so as these huge databases gradually change patterns over time. A short survey of various clustering techniques and their applicability to large databases can be found in [7] and [8].

Research on automated clustering of 3D CAD databases till now has been quite limited. In [3-4], [10] Kriegel et al develop a system for top-down hierarchical clustering of CAD database based upon a clustering technique called OPTICS. The authors evaluated several secondary approaches to actually extract the meaningful clusters from the hierarchy generated by the OPTICS system. Furthermore, in [4] the authors describe three different approaches to determine a representative object for the recognized clusters. The authors also describe an industrial prototype system which implemented the described algorithms. However, the research in [3-4], [10] has been focused more on the theoretical behavior of various clustering techniques, and less stress has been given on the similarity search basis, the nature of the clustering results and cluster hierarchies obtained.

A lot of research has also been done on physical storage clustering of spatial objects, with focus on decreasing the object access time, see for example [5]. This is especially useful when range queries such as window queries are being performed on the database in order to actually retrieve the complete object descriptions. Storage clustering is however difficult to achieve when the data servers are geographically distributed, and product development occurs at multiple sites, as is the case with most large manufacturing enterprises. Needless to say, this is a very complex task and few PDM systems today have real support for distributed data. Moreover, in most situations bulk description data are rarely retrieved from CAD databases.

To summarize, most of the research that has been done on clustering spatial database has been largely theoretical in nature, and engineering concerns have not really been addressed. There has also been relatively less focus on understanding the nature and behavior of similarity search methods which form the basis for these clustering techniques. Moreover, domain knowledge and understanding has rarely been leveraged in developing an effective and efficient clustering technique. The next two sections briefly analyze these important aspects, partly in order to justify the clustering algorithm which is described later.

3. CLUSTER DISTRIBUTION IN CAD DATABASES

An exhaustive study was undertaken to understand the nature and distribution of clusters in industrial CAD databases to help choose an appropriate clustering technique. This was partly motivated by the typical behavior of various similarity search techniques, when applied to large industrial CAD databases. Moreover, it is important to understand the behavior of these similarity search techniques, as they form the basis for the clustering algorithms. This section first describes the behavior of a few shape based similarity search methods that were studied and gives probable reasons for such behavior. It then briefly presents various hypotheses on different forms of cluster hierarchies that are possible in CAD databases.

The datasets that were used in the current research consisted of several sets of between 5,000 to 6,000 CAD models from various automotive and aerospace manufacturers, and large mechanical parts manufacturing establishments. These datasets are hereafter referred to as the *Indus* datasets. Another dataset of industrial models containing around 13,000 CAD models was obtained from the test database at GSSL, hereafter referred to as the *Test* dataset. A complete dataset was also created by merging all the above datasets containing a total of approximately 35,000 CAD models, hereafter referred to as the *Full* dataset.

3.1 Behavior of Similarity Search Methods

Three existing 3D similarity search methods were chosen: (i) the feature vectors consisting of moments, (ii) the D2 shape histogram [9], and (iii) a more comprehensive multi-dimensional index based on the Fourier transform [6], to understand how these similarity criteria behave when applied to large industrial datasets. Euclidean distance measure was used for the similarity search techniques based on the moments and the D2 histogram to find a distance measure between pairs of models in the database. For the similarity search method based on the Fourier transform the L1 or Manhattan distance measure was used as it was found to be more effective over large sets of precision versus recall experiments on pre-classified datasets.

A few representative similar groups from the *Indus* and *Test* datasets were chosen as benchmarks and pre-classified. These similar groups were manually extracted from the datasets and the number of similar models present in these groups was also determined after manually browsing the database. Several representative models from each of these groups were chosen randomly as the query or seed models. Extensive automated search experiments were performed with each of these query models, to get a set of resultant similar models using all the above mentioned similarity search methods. For each search, the results within a certain 'cutoff' distance were considered valid, and the rest were neglected. For each set of query model and similarity search method, the 'cutoff' distance was varied over large ranges and plotted against the 'precision' value of the results, where,

Cutoff distance = Maximum distance, with respect to the query, allowed for a valid search result, and
Precision value = Ratio of the number of valid results to the total number of results.

For each set of query model and similarity search method a graph was obtained, indicating the variation in precision values as the cutoff value is increased. Because of the large number of graphs that were obtained from these experiments, only the average illustrative behavior of these plots for the various similarity measures is shown in Fig. 1.

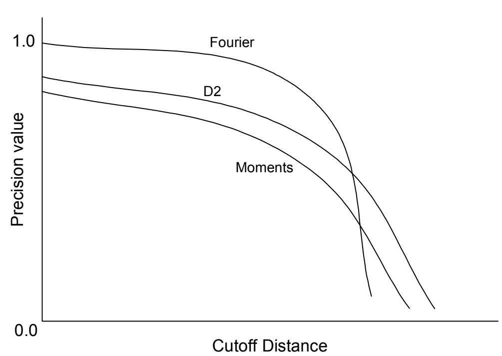

Fig. 1. Cutoff distance versus precision value plots.

As can be seen from the above illustrative plots, the curves for each of the similarity measures show a similar trend. Briefly, as the cutoff value is gradually increased from 0.0, the precision remains steady at a value of nearly 1.0 or decreases gradually up to a certain point where there is a sudden drop in the precision values. After this inflexion point the drop in precision values keeps getting steeper with increasing cutoff. Note that the cutoff point at which the curves drop suddenly varies with the similarity search method and the distance measure used. It should also be noted that the cutoff inflexion point also varies slightly depending upon the chosen query part but for most cases this range of variation is negligible for a particular similarity search method.

From the nature of the graphs in Fig. 1., it can be readily observed that after a certain value of 'cutoff' distance, the precision value decreases sharply indicating spurious results, which are not similar to the query model. It may also be observed that the graphs for similarity search methods based on moments and D2 shape histogram do not fall as sharply as that of the method based on Fourier Transform. This can be attributed to the fact that the method based on the Fourier transform generates a multi-dimensional and more comprehensive feature vector as compared to the other methods. Further interpretation of these plots requires that test results with the cutoff values greater that the inflexion point in the graph are analyzed manually. The ordered results for two fairly complex industrial query parts at a couple of cutoff points greater than the inflexion point are shown in Fig. 2.. In both cases the last 2 results are beyond the cutoff value at the inflexion point. Note that not all the results that came up are shown; due to lack of space only representative from the long list of results are shown.

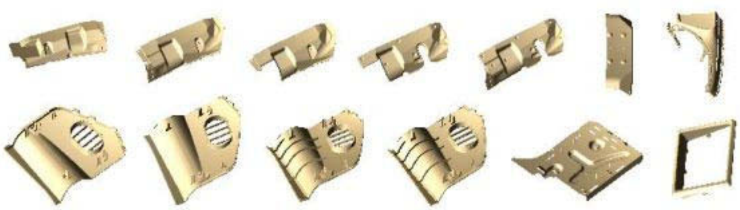

Fig. 2. Sample search results with high cutoff distances.

As can be seen from Fig. 2., the additional results at the end that are appended when the cutoff value is increased beyond the inflexion point are mostly random in nature and are hardly similar to the original query model. This is a very significant fact, because it indicates that these similarity measures do not behave in a manner where relaxing the cutoff values of these measures will bring in a hierarchy of results. This is believed to be true for most other 3D shape similarity measures as well. This is partly because the extensive manual studies and surveys suggest that industrial 3D CAD databases do not have an intuitive hierarchical representation, when the only criterion for similarity is the geometric shape of the 3D models. This is considered in more detail in the next section, where we hypothesize on the various possible forms of CAD database hierarchies.

3.2 Cluster Hierarchy in CAD Databases

Except for some very trivial cases it is difficult to find shape based hierarchy in industrial CAD databases. This might be because we can intuitively guess the hierarchy in simpler cases, but not in complicated models. In any case, imposing a rigid mathematical hierarchy where none exists, for example by merging CAD models as in [3] and [10], will only hinder the visual data mining process. Some trivial cases where a shape based hierarchies do seem to exist are shown in Fig. 3.. Note that even in these cases detecting a hierarchy is largely a subjective matter.

Fig. 3. Shape hierarchies in simple cases.

Instead of only the overall geometric shape, even if one considers only the topology or the skeleton of the CAD models, hierarchies found would still be largely limited to simple models. This is because the topology and skeleton structures for non-trivial mechanical CAD models tend to be much more disorganized as compared to say molecular models. Functionality based hierarchies however do exist in CAD databases. For example, a collection of bearings might not all be of the same shape, but do belong to the same class of models. Also the fact that this class of bearings may have been sub-grouped into further class types like, roller-ball, inner-slide etc. creates a functionality based hierarchy. These hierarchies are however industry specific and usually limited to only a portion of complete database which contain these standard parts. Also, theoretically if these hierarchical relations are already available then they can easily be superimposed after a shape based flat partitioning into clusters is achieved.

One very intuitive form of hierarchy however does exist in industrial CAD databases. This hierarchy is primarily based on the 'composed of' and 'where used' concepts. For example, a typical industrial dataset consists of a variety of differently sized models, and often most of the smaller models in turn form parts of larger models and assemblies. This creates complex hierarchies between various objects in the database. Also several key components or features of a model tend to occur repeatedly across a variety of significantly different models while essentially performing the same function, e.g. a collection of supporting rib features. Similarity search based on the 'where used' concept and key features of a component will greatly increase the usefulness of such a system. These complex relationships and pseudo-hierarchies therein are promising avenues for future research.

4. CLUSTERING TECHNIQUE

This section describes a simple single scan clustering technique for CAD databases which incorporates some graph-theoretic heuristics, for establishing links between clusters. The similarity search technique used as the basis for this clustering technique was the multi-dimensional Discrete Fourier Transform (DFT) based technique described in [6], as it outperformed most other similarity search procedures by a large margin in precision versus recall experiments carried out over the various datasets. Moreover, the hierarchical nature of the multi-dimensional feature vector generated by the DFT based method allows each search query to be performed in sub-linear time with respect to the database size. The next sub-sections briefly describe the shape similarity search technique based of the DFT and some modification that were applied to the original technique in [6], and then describe the clustering technique in detail.

4.1 DFT-based Shape Similarity

The shape similarity technique proposed in [6] consists of various well defined steps:

- Point cloud approximation.
- Model Orientation.
- 2D Depth-Map Projection.
- 2D Depth-Map Transforms.

Briefly, the method consists of transforming the 3D CAD model into a canonical orientation so as to make it invariant to the set of rigid transformations. The point could approximation step facilitates the step of model orientation, which uses a modified version of the Principal Component Analysis (PCA) method. It also helps in the pseudo-voxellization of the model surface, used for 2D depth-map projections. Model orientation using PCA was not found to be too robust across a diverse set of models, mostly because of its sensitivity to outliers. The current implementation therefore use an alternative algorithm, known as the Minimum Volume Bounding Box (MVBB) algorithm [2] to orient the model invariantly in three-dimensional space. This method was found to be superior to PCA for most mechanical CAD models, which are usually prismatic in nature.

After pseudo-voxellization of the model surface, the voxellized model is projected along X, Y and Z axes to create 2D depth-maps. A significantly large raster resolution of 128 is used to voxellize the model, which gives on projection depth-map images of 128x128 resolution. To create a representative 2D projection, several techniques may be used to appropriately weigh the voxels. It was found that a simple weighing of the voxels based upon distance from the plane of projection produced reasonably good depth-maps. The goal here was to get depth-map projections with a good mixture of flat filled regions and sharp edges, which are quite representative of the 3D model and capture its depth features along the projection direction. The current implementation uses only the DFT and does not use the Harr Wavelet Transform (HWT) as mentioned in [6] since, on an average, the HWT, when used in conjunction with the DFT, only marginally contributed positively to either the precision or the recall values for the given datasets. A small number of low-frequency phase and magnitude coefficients of the DFT were taken as the signature for the model. Fig. 4. shows the transformed model states during extraction of the shape measures for a sample CAD model. (i) Shows the original model, (ii) the MVBB for the model, (iii) shows the 2D depth-map image projected along the Z axis (128x128 pixels), (iv) normalized Fourier Transform (128x128 pixels) and, (v) shows the reconstructed depth-map image shown in (iii), but using only 144 (12x12) Fourier coefficients from the center of the transform values shown in (iv). From (v) it is apparent that even a hundredth fraction of the hierarchical DFT coefficients gives a fair enough representation of the original model, while allowing for minor differences between similar models.

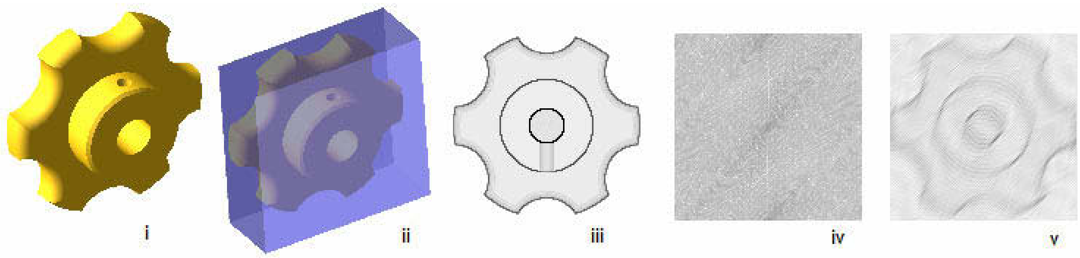

Fig. 4. Transformed model states.

Several performance results were generated during the process of extracting the signatures and storing them in the database. All tests were done and results were generated on a 2.0 GHz AMD Opteron™ machine with 1.0 GB of RAM. The process of generating the signatures for any model takes nearly constant time, irrespective of the size and complexity of the model. On an average the process of extracting the signatures and storing them into a database took around 2.0 seconds per model. Due to the hierarchical nature of the DFT magnitude coefficients, the search time complexity is sub-linear with respect to the database size. The sub-linear behavior is achieved by performing repeated range queries on an increasing number of DFT magnitude coefficients, until a reasonably small number of query results are obtained on which the actual L1 distance computation between the signatures are performed. The histogram of search time for a given *Indus* dataset containing approximately 5,000 models is shown in Fig. 5. Average search time for this dataset was around 0.25 seconds.

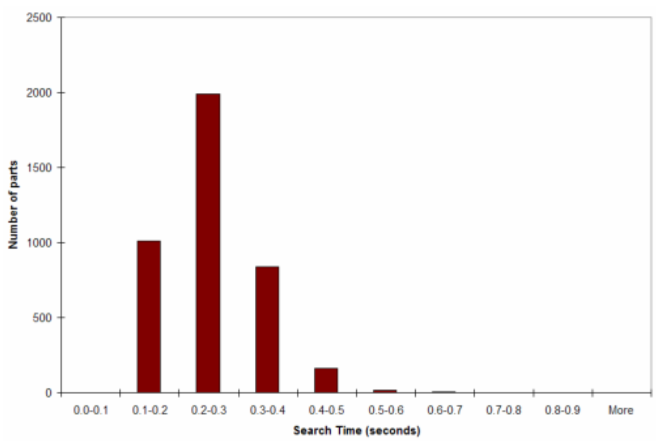

Fig. 5. Search time histogram.

Signatures for all models in a given dataset are generated by this method, and the resultant part names and signatures are stored in a flat database. A few of the magnitude coefficients of the DFT are stored in separate columns, to facilitate fast range queries. The rest of the signature is stored as a BLOB (Binary Large OBject) data. These databases are then passed on to the clustering algorithm to determine the clusters.

4.2 Single Scan Clustering

The clustering technique used was based on the single scan algorithm. The basic idea of a single scan algorithm is to group objects of the database into clusters based on a local similarity condition thus performing only one scan through the database. The result is a single level partitioning or flat clustering of the database into clusters. If the average runtime complexity of the sub-linear similarity search query is assumed to be O(*log n*), then the overall runtime complexity of the single scan algorithm is only O(*n log n*). The algorithmic schema for the basic single scan clustering algorithm is as follows:

Procedure SingleScanClustering (Database *D*)
    For each object *o* in *D* ***do***
        If *o* is not yet member of some cluster ***then***
            Create a new cluster *C*
            While neighboring objects satisfy the cluster condition ***do***
                Add them to *C*
            End While
        End If
    End For

A modified version of single scan algorithm has been used where, apart from each individual cluster there is also associated with each cluster a peripheral cluster. Intuitively, each loose cluster consists of a core cluster with densely placed models and a peripheral cluster with sparsely placed models, see (i) in Fig. 1.. The peripheral clusters are obtained by using a heuristic where objects which occur in at least half the similarity search results with each object in the loose cluster as the seed object, are added to the core cluster whereas the rest are assigned to the peripheral cluster. Once all the clusters and peripheral clusters are created, links between pairs of clusters are created where the peripheral clusters overlap. Linking clusters in this manner lends the algorithm a slight graph-theoretic nature and eases navigation and usability. The overall algorithm schema may be summarized as follows:

Procedure SingleScanClustering2 (Database *D*)
    For each object *o* in *D*
        If *o* is not yet a member of some cluster or peripheral cluster ***do***
            With *o* as a seed part, perform a search to get a collection of similar parts *P*.
            For each part *p* in *P* ***do***
                With *p* as a seed object, perform a search to get a collection of similar parts *P'*.
                Add the objects of *P'* to *P*, repeating the common objects, and thus increasing their occurrence count.
            End For
            For all objects in *P* with occurrence count greater than or equal to *n/2,* form a cluster *C*.
            For all objects in *P* with occurrence count less than *n/2,* form an associated peripheral cluster *PC*.
        End If
        For each peripheral cluster ***do***
            Find intersection with all other peripheral clusters.
            If an intersection is non-empty then create a linkage between the corresponding clusters.
        End For
    End For

Intuitively the clustering process can be visualized as shown in Fig. 6.. (i) Shows a single core cluster and its peripheral cluster, and (ii) shows the formation of linkages between clusters with intersecting peripheral clusters.

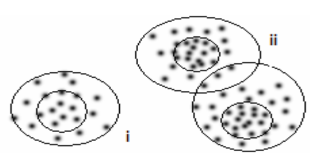

Fig. 6. Formation of clusters and linkages.

5. EXPERIMENTAL RESULTS

An evaluation application was developed to extract the signatures from a 3D model and store it in a flat database. Apart from the signatures the application also stores an isometric projected JPEG image of the models in a separate table of the database for visualization purposes. Indexing of various tessellated 3D file formats like VRML and STL are supported by the evaluation application. Apart from this most other formats from various industrial 3D CAD modelers are also supported in a commercialized version. The evaluation application also included a database browser from where a seed part can be selected as query and search can be invoked. The search results are visible as static JPEG images in the application area. A screen shot of the application and that of the database browser is shown in Fig. 7.

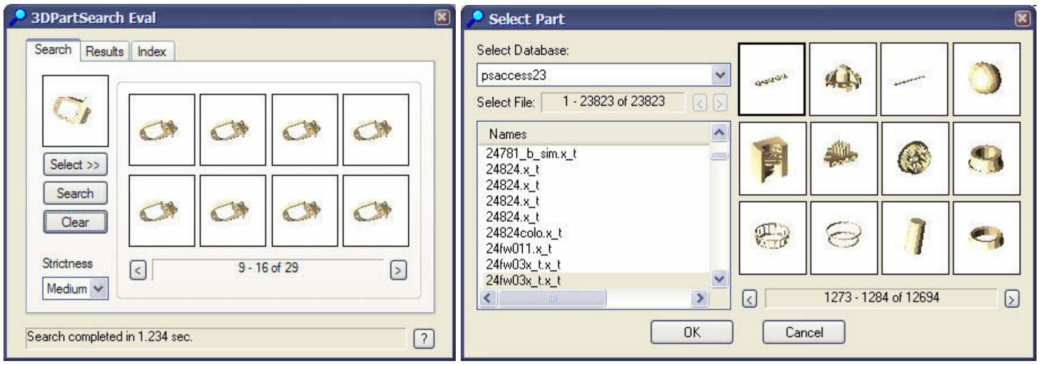

Fig 7. Screenshot of evaluation application and the database browser.

The next sub-sections first briefly look at the kind of statistical results that can be generated during the process of indexing, and then outline some clustering results and cluster visualization through the evaluation application.

5.1 Statistical Results

As was mentioned earlier various statistical results are obtained by running the clustering algorithm under different cutoff values of similarity search. For example, when the clustering algorithm is run at a very low cutoff value, slightly greater than 0.0, then the clusters generated are clusters of duplicate or near duplicate models. Such a statistic is very useful when the goal is to trim the dataset by removing redundant groups of very similar models. Similar statistics for general clusters can be obtained by running the clustering algorithm at various cutoff values somewhat lower than the cutoff value at the inflexion point. Results of running the clustering algorithm with a cutoff value nearly equal to 0.0 (*Exact*), and at two other higher cutoff values of 0.5(*Medium*) and 0.58(*Low*) on a sample *Indus* dataset with around 6,000 models is shown as histograms in Fig. 8..

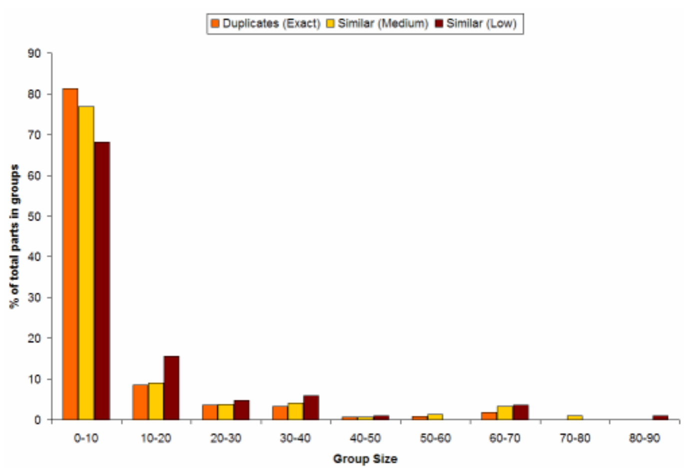

Fig. 8. Histogram of clusters containing duplicate and similar models.

As can be seen from the histogram, most duplicates occurred in clusters where the cluster size was small i.e. in the range of 2-10. The maximum duplicate cluster size was in the range of 60-70, which indicates that there are a few small groups of duplicate parts which contain a large number of duplicates. Note that the histogram for clusters of similar models is very similar in nature to the histogram of duplicate models, except for a perceptible drop in the number of clusters in the range 0-10, possibly because some clusters in this range got merged with other clusters. This indicates that relaxing the cutoff value from 0.0 to near the inflexion point did not drastically affect the clusters size or distribution, which in turn indicates that the clusters in this dataset almost always contain models with little or no difference between them.

All the results from other datasets have not been presented, partly because most of the histograms are usually similar in nature to the one shown from the sample database in Fig. 8.. However some interesting trends were found that are currently being investigated and will be reported in future publications.

5.2 Clustering Results

Cluster visualization is as difficult and intriguing a task as clustering itself. Currently, we use a crude technique for cluster visualization, where the clusters are simply listed in no particular order. In a manner similar to how the seed part is chosen in the evaluation application using the database browser dialog shown in Fig. 7., another dialog enables the user to visualize the clusters. However, due to design restrictions linkages between clusters cannot be visualized through the current application. A sample screenshot of the dialog for visualizing clusters through the evaluation application is shown in Fig. 9.

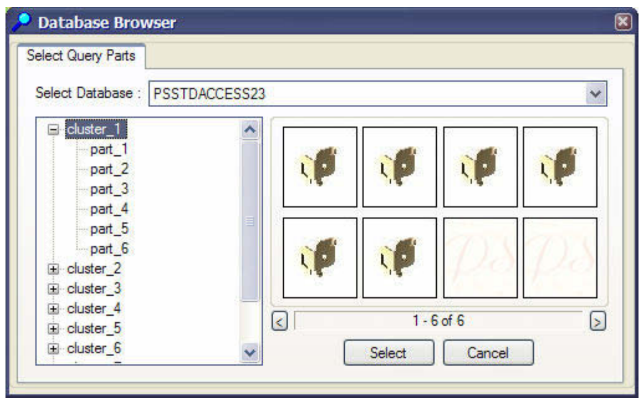

Fig. 9. Cluster visualization dialog.

The clustering results were manually analyzed and in most cases found to be quite accurate. It is difficult to accurately measure the effectiveness of the clustering method on such large industrial datasets, as pre-classification would require enormous effort. To measure the effectiveness approximately some representative groups were found by manually scanning the database and compared with the clustering results to determine the clustering effectiveness. The results were quite encouraging, but cannot be illustrated due to limitations on space.

6. CONCLUSIONS

This paper has proposed a simple and automatic flat clustering or partitioning method for large CAD databases. The clustering method is based on the single scan algorithm and partly graph-theoretic in nature. It has been empirically shown how a hierarchical clustering method is unsuitable for CAD databases due to the inherent non-hierarchical nature of industrial CAD databases. The proposed clustering method was applied to significantly large industrial CAD datasets and results obtained were highly encouraging. It was also shown how several statistical results could be generated during the process of clustering, which gives new insight into the contents of the datasets. The paper also illustrated a compact application for indexing, searching and visualizing the clusters.

Future directions for research include looking into better visualization techniques for these clusters, like dynamic 3D visualization frameworks. Another area of work will involve exploring hierarchical clustering techniques based on the 'where used' and 'key features' based concept. This would also entail creating similarity search methods which exploit these concepts efficiently.

7. ACKNOWLEDGEMENTS

The author would like to thank Geometric Software Solutions Ltd. (GSSL) for extending cooperation in promoting the research. Special thanks to Anil Risbud and Dr. Vinay Kulkarni for their insight and encouragement throughout the project. Thanks are also due to colleagues in GSSL for their direct and indirect participation in the various studies conducted under this project.

8. REFERENCES

[1] Ankerst, M., Kastenmüller, G., Kriegel, H.-P., and Seidl, T.: 3D Shape Histograms for Similarity Search and Classification in Spatial Databases. In Proc. 6th International Symposium on Spatial Databases (SSD'99), Hong, Kong, China, July 1999.

[2] Barequet, G., Har-Peled, S.: Efficiently Approximating the Minimum-Volume Bounding Box of a Point Set in Three Dimensions. J. Algorithms 38(1): 91-109 (2001).

[3] Brecheisen, S., Kriegel, H.-P.,Kröger, P., Pfeifle, M.: Visually Mining Through Cluster Hierarchies, Proc. SIAM Int. Conf. on Data Mining (SDM'04), Lake Buena Vista, FL, 2004, pp. 400-412.

[4] Brecheisen, S., Kriegel, H.-P., Kröger, P., Pfeifle, M., Viermetz M.: Representatives for Visually Analyzing Cluster Hierarchies, Proc. 4th Int. Workshop on Multimedia Data Mining (MDM/KDD'03), Washington, DC, 2003, pp. 64-71.

[5] Brinkhoff T., Kriegel H.-P.: The Impact of Global Clustering on Spatial Database Systems, Proceedings 20th International Conference on Very Large Data Bases, Santiago, Chile, 1994, 168-179.

[6] Chakraborty, T., Venkataraman, S., and Sohoni, M.: A Fast 3D Shape Search Technique for 3D CAx/PDM Repositories, Technical Paper, Society of Manufacturing Engineers (SME), Aug. 16, 2004.

[7] Ester M., Kriegel H.-P., Sander J., Xu X.: Clustering for Mining in Large Spatial Databases, KI (Artificial Intelligence), Special Issue on Data Mining, ScienTec Publishing, March 1998, pp. 18-24.

[8] Ester M., Kriegel H.-P., Xu X.: A Database Interface for Clustering in Large Spatial Databases, Proc. 1st Int. Conf. on Knowledge Discovery & Data Mining (KDD '95), Montreal, Canada, 1995, pp. 94-99.

[9] Funkhouser, T., Min, P., Kazhdan M., Chen, J., Halderman, A., Dobkin, D., and Jacobs D.: A Search Engine for 3D Models. ACM Transactions on Graphics, 22(1), pp. 83-105, January 2003.

[10] Kriegel, H.-P., Brecheisen, S., Januzaj, E., Kröger, P., Pfeifle, M.: Visual Mining of Cluster Hierarchies, Proc. 3rd Int. Workshop on Visual Data Mining (VDM@ICDM2003), Melbourne, FL, 2003, pp. 151-165.

[11] Kriegel, H.-P., Brecheisen, S., Kröger, P., Pfeifle, M., Schubert, M.: Using Sets of Feature Vectors for Similarity Search on Voxelized CAD Objects, Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'03), San Diego, CA, 2003, pp. 587-598.

[12] Vranic, D.V., and Saupe, D.: 3D Shape Descriptor Based on 3D Fourier Transform. In: Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001) (editor Fazekas, K.), Budapest, Hungary, September 2001, pp. 271-274.