Validation of Perdue Engineering Shape Benchmark Clusters by Crowdsourcing

P.Jagadeesan*, J.Wenzel#, J.R.Corney*, X.T. Yan* A.Sherlock✧, C.Torres-Sanchez*, W.Regli+
*University of Strathclyde, #University of Edinburgh, ✧Shapespace, +Drexel University

Contact: Department of Design, Manufacture & Engineering Management (DMEM), University of Strathclyde, James Weir Building, 75 Montrose St, Glasgow, G1 1XJ
Tel: +44 (0) 141 548 2254; Fax +44 (0) 141 548 4870
e-mail: jonathan.corney@strath.ac.uk

Abstract: The effective organization of CAD data archives is central to PLM and consequently content based retrieval of 2D drawings and 3D models is often seen as a "holy grail" for the industry. Given this context, it is not surprising that the vision of a "Google for shape", which enables engineers to search databases of 3D models for components similar in shape to a query part, has motivated numerous researchers to investigate algorithms for computing geometric similarity. Measuring the effectiveness of the many approaches proposed has in turn lead to the creation of benchmark datasets against which researchers can compare the performance of their search engines. However to be useful the datasets used to measure the effectiveness of 3D retrieval algorithms must not only define a collection of models, but also provide a canonical specification of their relative similarity. Because the objective of shape retrieval algorithms is (typically) to retrieve groups of objects that humans perceive as "similar" these benchmark similarity relationships have (by definition) to be manually determined through inspection.

However frequently the practical difficulty of assembling groups of volunteers willing to spend hours, manually sorting hundreds of components into families of similar shape, means that the similarity relationships associated with benchmark collections are, in practice, often defined by only a handful of individuals.

This paper reports the methodology developed to employ a commercial 'Crowdsourcing' service to distribute the task of assessing the similarity of models in a 3D dataset to anonymous workers on the Internet. To determine the effectiveness of this distributed approach it was applied to the class of "107 flat-thin wall components" within the "Purdue Engineering Shape" Benchmark's collection of 3D models. The resulting families of similar shapes (defined at varying resolutions) identified within the dataset, show close correspondence with Purdue's published clusters. This validates the use of Crowdsourcing as fast, cheap and effective method of content classification for CAD data.

Keyword: 3D Search, 3D Content Based Retrieval, CAD database, CAD data management, Shape Benchmark

1 Introduction

This paper is concerned with the combination of two approaches to problem solving:
Micro-outsourcing, or Crowdsourcing, is a neologism for the act of taking a task traditionally performed by an employee or contractor, and outsourcing it to an undefined, generally large group of people, in the form of an open call. For example, the public may be invited to develop a new technology, carry out a design task, refine an algorithm or help capture, systematize or analyze large amounts of data [1].

A **Human Intelligence Task** (HIT) is a problem that humans find simple, but computers find extremely difficult. For example a HIT related to a photograph could be: “Is there a dog in this photograph?”

The recent innovation of using Crowdsourcing to deliver HITs over the Internet has created a feasible way of providing cheap, robust, content based, analysis of digital data such as text (“translate the sentence into French”) or images (“choose the best picture of the hotel”).

The Crowdsourcing approach is exemplified by Amazon's “Mechanical Turk” [3] (mturk.com) site that provides an online marketplace enabling computer programs to co-ordinate the use of human intelligence to perform tasks which computers are unable to do. *Requesters*, the human beings that write these programs, are able to pose tasks known as HITs (Human Intelligence Tasks), such as choosing the best among several photographs of a storefront, writing product descriptions, or identifying performers on music CDs. *Workers* (called *Providers*) can then browse among existing tasks and complete them for a monetary payment set by the Requester.

To place HITs, the requesting programs use an API. Requesters can ask that Workers fulfill Qualifications before engaging in a task, and they can set up a test in order to verify the Qualification. They can also accept or reject, the result sent by the Worker, which reflects on the Worker's reputation. Workers can be anywhere in the world. Payments for completing tasks can be redeemed on Amazon.com via gift certificates or alternatively be realised as cash and transferred to a Worker's bank account. Requesters, which are typically corporations, pay 10 percent of the price of successfully completed HITs (or more for extremely cheap HITs) to Amazon.

Figure 1 is typical of the tasks found on mTurk. Using high resolution aerial photographs the problem of searching a vast area for signs of a missing aircraft was posted as a collection of tens of thousands of HITs that required only a simple ‘Yes or No’ answer.

The top row of three images in Figure 1 shows samples of the pictures used to instruct the “providers” in the pattern being sought. The lower picture illustrates the task presented to a worker.

Other HITs require workers to go beyond providing simple yes/no answers about a picture's content by interactively highlighting (with mouse movements) particular objects in a picture such as “the edge of a road” or the location of road signs. In this way ten's of thousands of HITs are being processes every month in a market that exemplifies how the digital economy can so effectively connect sellers and buyers with a flexibility previously unimaginable.

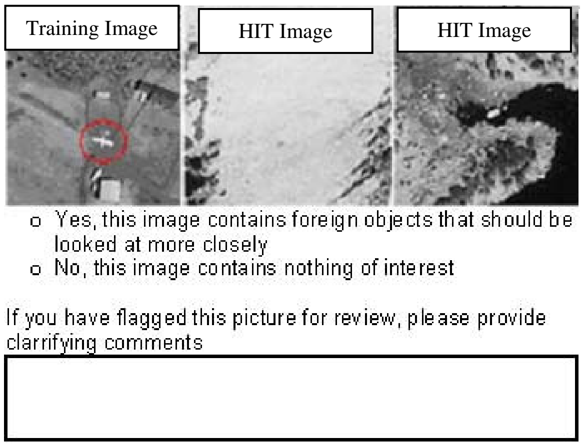
Although very little quantitative information about the effectiveness of Crowdsourcing has been published, observation of the mTurk site and the HITs being posted suggests the following:

1) Many HITs are processed very fast: the number of HITs available frequently fluctuates by several thousand over a day. Although some difficult or complex HITs persist unaccepted for weeks and ultimately go uncompleted.

2) Providers are willing to work for surprisingly low rewards, prices of $0.01 are common.

3) Graphical HITs are very popular (they never remain on the site for long).

Figure 1 Typical Content Analysis HIT



The objectives of the work reported here is to investigate if the same approach can be used to solve the geometric reasoning problems found in Mechanical CAD/CAM and, if so, quantify the performance (time, cost, accuracy) and understand the limitations (e.g. complexity of the shape or task description) of the approach.

An initial investigation into whether Crowdsourcing can be applied to these sorts of problems (i.e. if Geometric Reasoning tasks can be described clearly enough so that a culturally diverse workforce can comprehend what is required in a few seconds and deliver the result in a matter of minutes, or if the answers produced are in agreement with a consensus result, rather than being simply a broad distribution generated by random clicking) can be found in [2].

This paper reports the results of employing a commercial ‘Crowdsourcing’ service to distribute the similarity assessment of a 3D dataset to anonymous workers on the Internet. To determine the effectiveness of this approach the class of “flat thin wall components” within the “Purdue Engineering Shape” Benchmark’s collection of 3D models[9] is used to test the ability of Internet Crowdsourcing to determine families of similar parts. The results are compared against Purdue’s own published similar shapes within that group. The rest of the paper is structured as follows; the next section provides background to the technology and research literature associated with Crowdsourcing and Clustering CAD-models; section 3 details the experimental methodology and section 4 presents the results and discusses the data; lastly section 5 draws some conclusions and describes, some of the authors’ future objectives.

2 Background

The following sections provide an introduction to both Crowdsourcing and the computation of clusters. Although there is little academic work on Crowdsourcing an extensive literature exists in magazine articles, user’s blogs and application reviews.

2.1 The 3D similarity HIT

To find out if 3D similarity could be effectively Crowdsourced, an mTurk HIT was created containing the entire class (107 parts) of “flat-thin wall components” in Purdue’s Engineering Shape Benchmark (ESB) [9]. The HIT presented a “pool” of 107 images showing isometric views of the individual CAD-models in the collection. Workers were asked to “put similar looking models together into groups” (see Figure 2) by clicking first on their image and then in one of the rows below the pool. In this way every image selected appeared below the initial pool of images (see Figure 3) in a row (i.e. cluster) of part images judged similar by the mTurk worker. The workers were asked to continue this process until there were no images left in the pool. Facilities were available for workers to edit the contents of clusters during the process, create new rows (i.e. clusters) and move parts between rows.

Figure 2 Selecting images from the pool

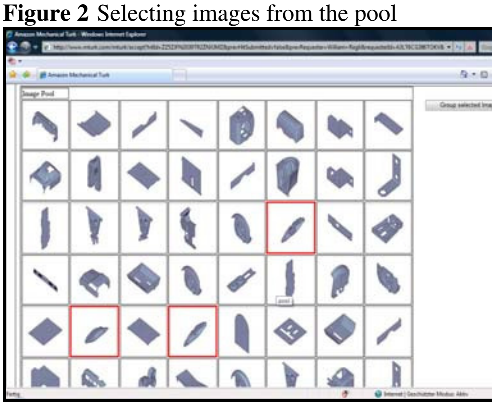
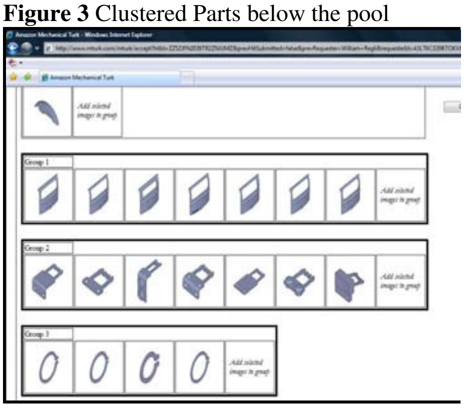


Figure 3 Clustered Parts below the pool



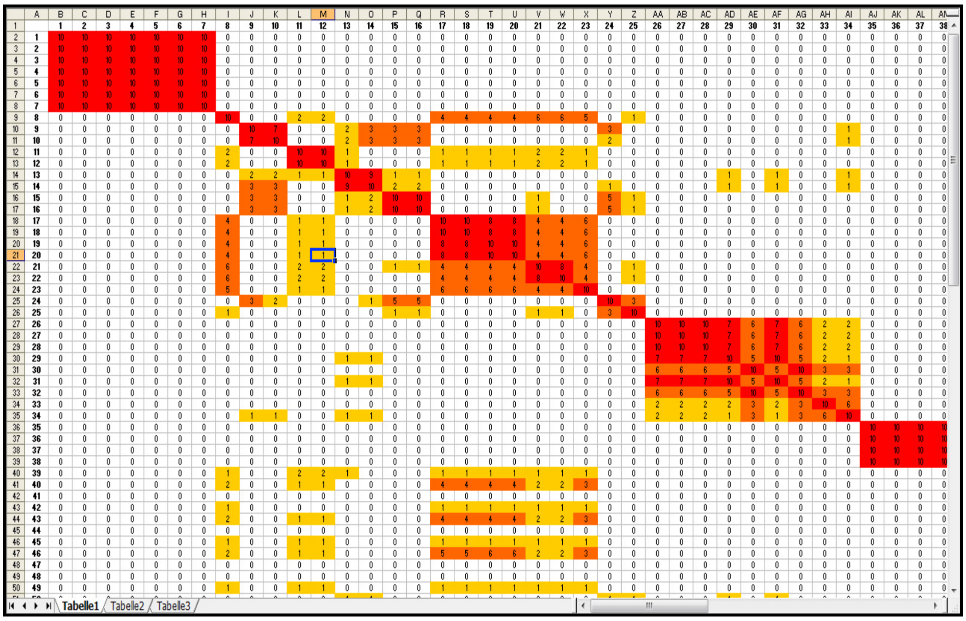
The educational background, previous exposure to 3D CAD and other personal data (such as gender, nationality and age) were also requested (although interestingly no strong correlation between previous experience and the quality of the results could be established). Details of the mTurk workers can be found in [8]. There were 14 HITs posted[1] on the MTurk site in August 2008. 10 of them were completed in a correct format, and 4 of them rejected. It should be noted that 1 of the rejected ones was due to browser compatibility problems (as task had to be carried out in Internet Explorer) and only 3 of the 14 workers actually failed to understand the task or submitted the results of apparently random clicking.

The clusters identified by the workers were summed in a single similarity matrix, S, where the number of times each pair of models was clustered together in the same family of similar shapes (see Figure 4) was recorded. Given *n* different CAD-models, the matrix contained $n*(n-1)/2$ values [7]. The cells of the matrix held a similarity measure (ranging from 0 to 10) for each pair of CAD-models. In other words two parts could reach a maximum similarity of 10 in case where all the mTurk workers grouped them together and zero when they had never been associated.

One hundred and seven models in the ESB's "flat-thin wall components" collection are divided into nine sub-groups or families. To enable comparison with these nine families the next section describes how "clusters" were generated from the similarity matrix (S) defined by mTurk workers

[1] The 14 HITs were made available in two rounds: for the first one, 10 HITs were posted and all were accepted within 8min 48sec. The completion time for all the workers to submit their results was 37min 18sec. 7 results were valid, 1 technical problem and 2 invalid entries. For the second round, 4HITs were offered to workers who accepted the task within 2min 30sec. The completion time was 24min 18sec. There were 3 results valid and 1 invalid. The payment reward per valid HIT was $4.

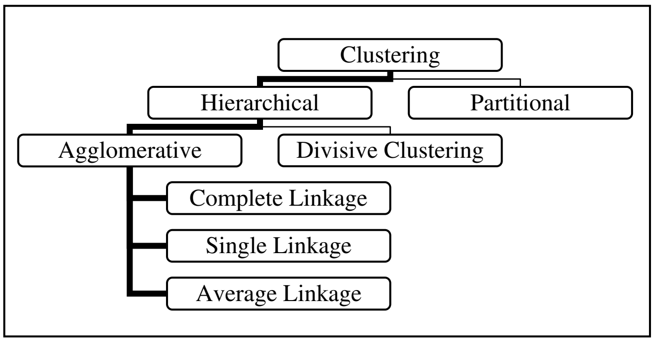
Figure 4 Excerpt from the similarity matrix



2.2 *Overview of Clustering Methods*

Clustering can be defined as "the classification of objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait" [4].

Figure 5 Classification of clustering methods after [5]

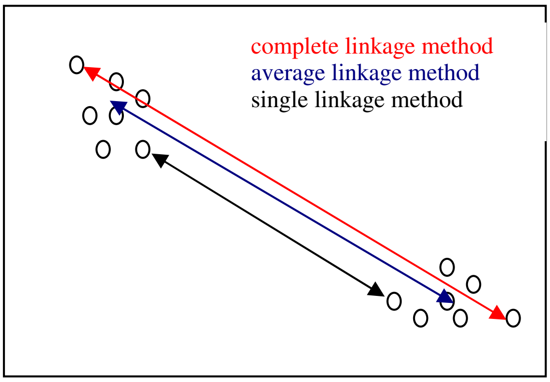


Broadly speaking there are two kinds of clustering algorithms used in data analysis 1) hierarchical clustering approaches which produce a nested series of partitions; 2) partitional methods which generate binary divisions (e.g. k-means algorithm) [5] of data. Such divisive (or "top-down") methods start with all items together in one single cluster and perform binary splitting operations until a stopping criterion is met [5]. This study is concerned with the first type, the so called agglomerative method of the hierarchical approach. This "bottom-up" method [4] starts with each element as a single cluster (i.e. containing only one item), and successively merges clusters together until a stopping criterion is satisfied [5]. There are different approaches to determining exactly how individual items or clusters are selected for incorporation into clusters [6]. Figure 5 illustrates the hierarchy of clustering methods (bold line indicates the methods used in this work).

The basis for all the different clustering algorithms is a distance measure that determines how similar or dissimilar the different elements are to each other [5].

This study has investigated three different methods known as "single-linkage", "complete linkage" and "average linkage" clustering. All linkage methods have in common that the clustering starts with the two most similar elements [4]. As soon as two elements (or later on groups of elements) are clustered together the columns and rows of a similarity matrix associated with them are merged and the distances are updated [4]. The linkage methods differ in the way the distance or similarity between the clusters is calculated (compare Figure 3):

Figure 6 Comparison of linkage methods



<u>1.) Single linkage clustering</u> (also called 'minimum method'): the distance between two clusters is the minimum distance between two elements of the different clusters [7]. In terms of similarity, the minimum distance is equivalent to maximum similarity.

<u>2.) Complete linkage clustering</u> (also 'maximum method'): the distance between clusters is equal to the maximum distance between two members of the different clusters. In terms of similarity, the minimum similarity between a pair of elements from different cluster is taken as the representative similarity between the clusters [7].

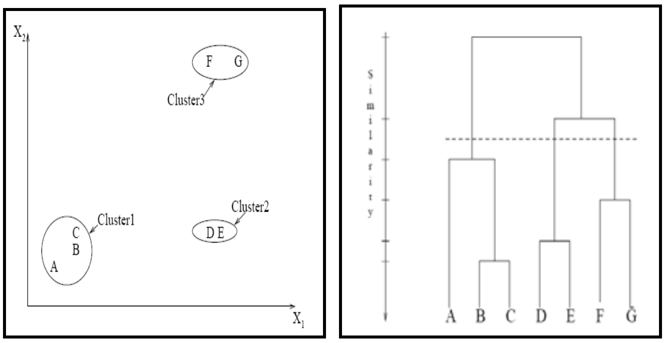
<u>3.) Average linkage clustering</u>: The average linkage method is a compromise between the two previous methods. The distance d between two different clusters A (having |A| elements) and B (having |B| elements) is the average distance taking all elements x and y of the clusters into consideration [4]:

$$d(\mathrm{A},\mathrm{B}) = \frac{1}{|A| * |B|} * \sum_{x \in A} \sum_{y \in \mathrm{B}} d(x, y) \qquad (1)$$

A common way of presenting the different stages of agglomerative clustering processes is *dendrograms*, which show the order in which elements or groups of elements are clustered into a tree structure. Usually dendrograms illustrate the complete clustering process until all elements are merged to one, all consuming, cluster [5]. Figure 7 shows an example of a clustering process with its associated dendrogram. The process started with the clustering of the elements B and C. The dotted line shows the current stage of clustering.

Generally the clustering process can be halted at any stage and there are two different criteria, commonly, used to stop clustering processes; 1) the distance criterion, when clustering is stopped if there are no more clusters within a certain distance; 2) the number criterion, where the process of merging elements ends if a certain number of clusters is reached. However, because the dataset had only 107 items, the clustering process was simply halted when all shapes had been added (i.e. the root cluster).

Figure 7 Clustering process and the according dendrogram, after [5]



3 Implementation and Results

The “Statistics Toolbox” of MATLAB was used to generate clusters. As the MATLAB clustering algorithms require a distance matrix instead of a similarity matrix as input, the similarity matrix M was normalized to a distance matrix D by equation (2):

$$\underline{D} = \frac{1}{10} * \left(\begin{bmatrix} 10 & 0 & \dots & 0 \\ 0 & 10 & & \dots \\ \dots & & \dots & 0 \\ 0 & \dots & 0 & 10 \end{bmatrix} - \underline{S} \right) \tag{2}$$

As described in section 2.2, the three linkage methods applied each differs in the way the distance between the clusters is calculated. Dendrograms were plotted for each of the different methods (see figures in Appendix-A available at http://www.strath.ac.uk/dmem/research/crowdsourcing/plm-appendix-A.pdf) and the influence of the linkage methods on the clustering process analysed. When the results of the “average” to the “complete” methods are compared the differences are surprisingly small. For example at the level where the parts are clustered into 20 groups (i.e. Appendix A[1]) the major clusters differ by only a single part highlighted in Figure 8.

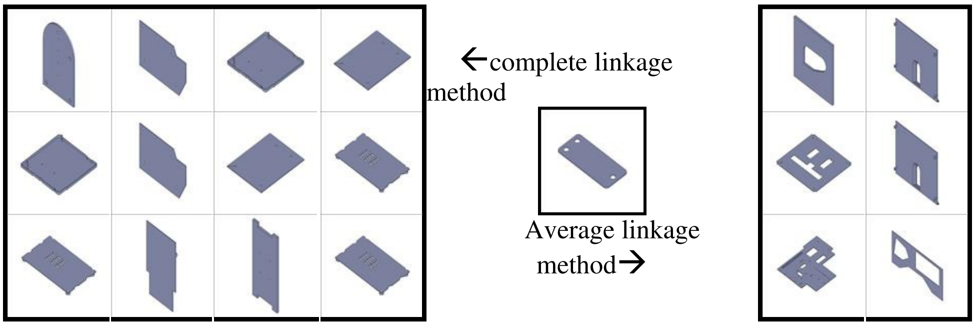
However the differences between these two linkage methods can be clearly seen in the structure of the higher levels of the dendrogram tree (i.e. near the root) in (appendix-A) Figures A.3. The average linkage method leads to a much richer structure in which clusters merge into larger clusters. By contrast the complete linkage dendrogram (appendix-A - Figure A.2) shows twelve clusters that are merged simultaneously at the root, this means for all combinations of clusters it is possible to find two parts that were never clustered together.

While the results of the “average” and the “complete” linkage methods are comparable, the results for the “single linkage” method are significantly different. Figure A.3 (appendix A) shows that the relative range of cluster sizes is much more variable for the

[1] Because of the paper length constraints Appendix-A (containing figure A.1, A.2 and A.3 is located at http://www.strath.ac.uk/dmem/research/crowdsourcing/plm-appendix-A.pdf

single linkage method. In "single linkage" the distances between the clusters are determined by the shortest distance between any two parts of the different clusters. This results in more and more parts being merged into already big clusters, while smaller clusters stay isolated until the late stages of the linkage process. As a result of these comparisons the authors decided to focus on the "average" linkage method because it resulted in dendrogram structures of clusters that allowed discrimination between families of similar models

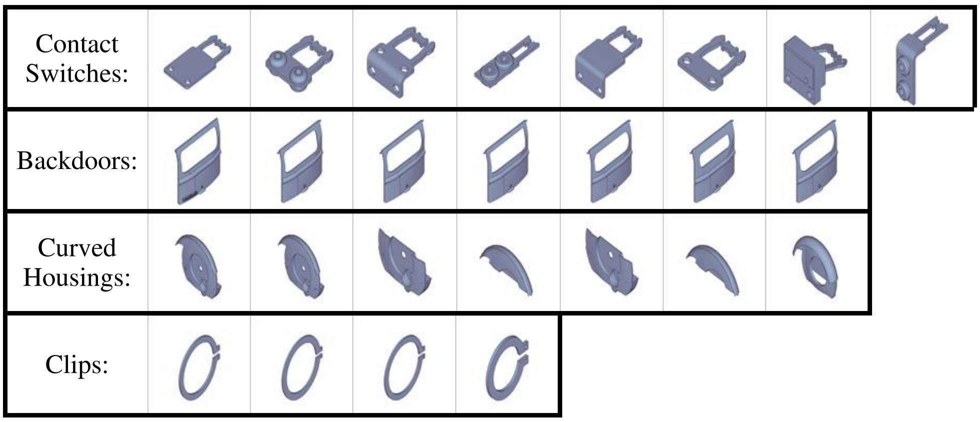
Figure 8 Example for different assignment using different linkage methods



3.1 Comparison between ESB results and 'Crowdsourced' clusters

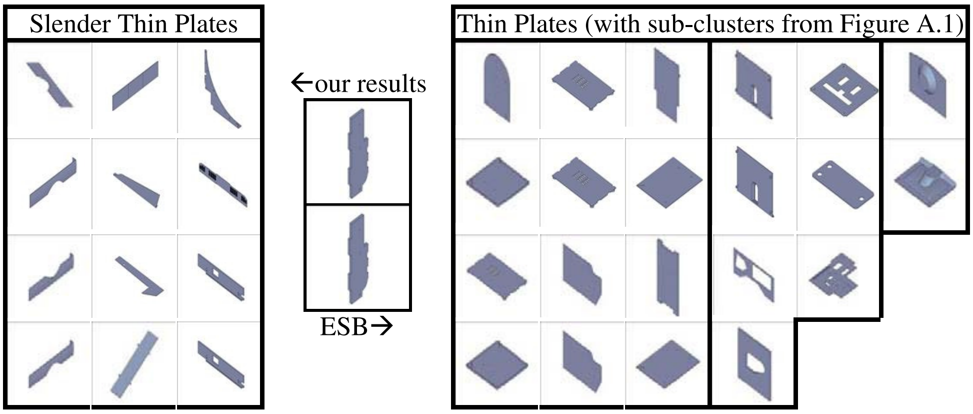
Despite the fact that all the 107 parts can be described as "flat-thin wall components", the ESB identifies 9 distinct families (or clusters) of models that can be identified within the overall classification. Focusing on the eight central clusters of ESB (and disregarding the "Miscellaneous" group) exact matches for half of them (Figure 9) can be identified at the clustering level illustrated in (appendix-A) Figure A.1.

Figure 9 Clusters identical to the ESB



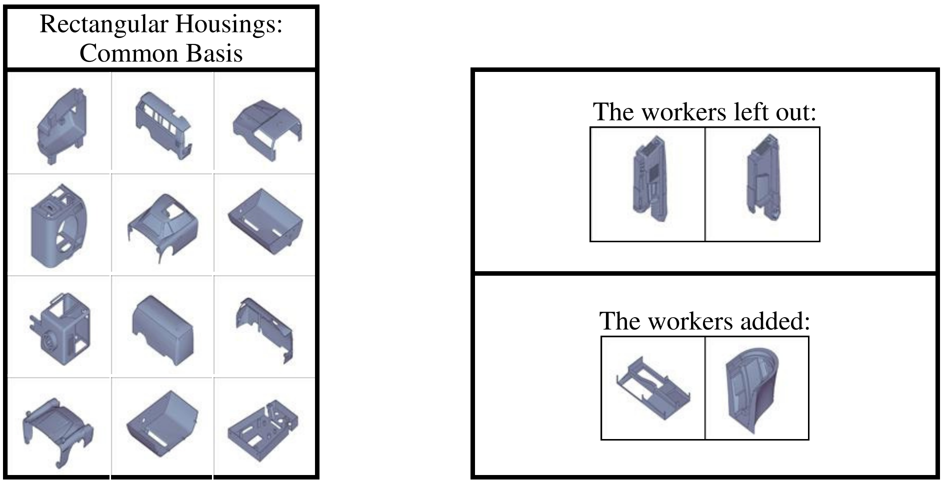
Close correlations can also be identified with two of the other ESB-clusters known as "thin plates" and "slender thin plates". At the clustering level shown in Figure A.1 (columns 3, 4 and 5 from the right), the "thin plates" cluster is spread across three separated groupings. But at higher levels of tree, these clusters are merged into a single cluster whose only difference to the ESB group is in the classification of the two "circuit boards". While "average" clustering assigns these "circuit boards" to the "slender thin plates", they were assigned to the "thin plates" group in the ESB (see Figure 10).

Figure 10 Different assignment of circuit boards in ESB and our results



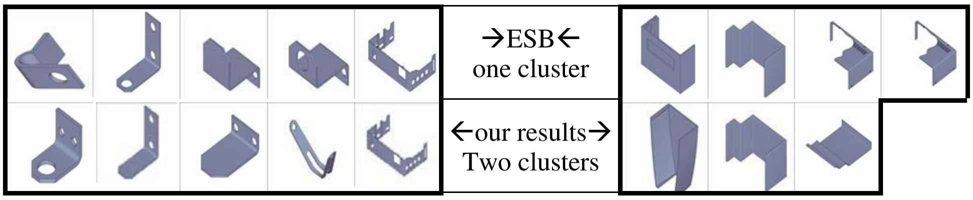
Even the group with probably the most complex and inhomogeneous parts in the ESB ("rectangular housings") can be recognized in the HIT results. In the ESB this group consists of 14 CAD-models. Although the mTurk workers left out two of the 14 parts, they added two other parts (from the "Miscellaneous" group in the ESB) see Figure 11.

Figure 11 Cluster "rectangular housings" in ESB and our results



The only major difference between the ESB and the HIT generated clusters appears in the group called "bracket like parts" where workers split this group up in two separate clusters (see Figure 12). It appears that the MTurk workers made a distinction between the presence, or absence, of holes, however despite this, the two branches of the dendrogram merge into a single group at a higher level.

Figure 12 Separation of “bracket like parts”



4 Conclusions and Future Work

The results of the Crowdsourced similarity clustering provide surprisingly good correlation with the ESB’s published groupings with differences in membership being attributable to inherently ambiguous parts whose correct assignment is arguable

Future work will investigate if similar performance is observed when HITs are used to classify the reset of the ESB. The result suggests that Internet Crowdsourcing can be used as an effective method of content based classification.

References

1 Surowiecki J., (2005), ‘The wisdom of crowds’, Random House, New York. ISBN: 978-0-385-72170-7

2 Jagadeesan P., Wenzel J., Corney J.R., Yan X.T., Sherlock A., Regli W., (2008), ‘Geometric Reasoning With a Virtual Workforce (Crowdsourcing for CAD/CAM)’, *2nd International Workshop Virtual Manufacturing VirMan 08 as part of the 5th INTUITION International Conference: Virtual Reality in Industry and Society: From Research to Application*, October 6-8, 2008, Torino, Italy, ISBN 978-960-89028-7-9

3 Amazon MTurk: http://en.wikipedia.org/wiki/Amazon_Mechanical_Turk (Accessed 30th July 2008)

4 Data clustering: http://en.wikipedia.org/wiki/Data_clustering (Accessed 1st Sept 2008)

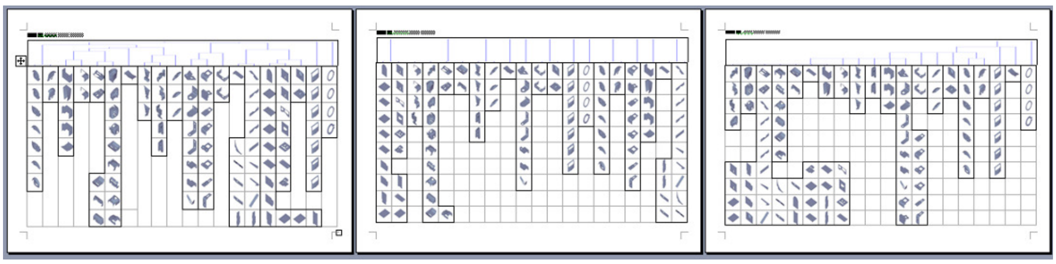
5 Jain A.K., Murty M.N., Flynn P.N, (1999), ‘Data clustering: A review’, *ACM Computing Surveys*, Vol. 31, No. 3, pp. 264-323

6 Kuiper F.K., Fisher L., (1975), ‘A Monte Carlo comparison of six clustering procedures’, *Biometrics*, Vol. 31, No. 3, pp. 777-783

7 Johnson S.C. (1967), ‘Hierarchical clustering schemes’, *Psychometrika*, Vol. 32, No. 3, pp. 241—254

8 Wenzel J., (2009), ‘Feasibility study for the “crowdsourcing” of geometric reasoning in mechanical CAD’, Institut fuer Fertigungstechnik und Werkzeugmaschinen, Leibniz Universitaet Hannover (Institution).

9 Jayanti, S. , Kalyanaraman, Y. , Iyer, N. & Ramani, K. *Developing An Engineering Shape Benchmark For CAD Models*, CAD, Vol 38: 9, Sept 2006, pp 939-953.



Appendix-A: Download from:
www.strath.ac.uk/dmem/research/crowdsourcing/plm-appendix-A.pdf