# Overview of Geometry Based Indexing and Search Tool

Satyandra K. Gupta

Director, Advanced Manufacturing Lab

University of Maryland, College Park, MD 20742

Technical Report, Advanced Manufacturing Lab, University of Maryland, College Park, MD 20742, December 2008

## 1. Motivation

3D CAD systems have become very popular in the industry. These systems are being used to generate 3D models of parts and assemblies. These models are used as a basis for analysis and generating process plans. The use of 3D models also allows virtual prototyping and hence reduces the need for physical prototyping. Nowadays, organizations routinely set up databases of CAD models to enable all participants in the product development process to have access to 3D data to support their individual functions. In addition to designers, manufacturing and service engineers are expected to greatly benefit from these databases. These databases are updated with the latest versions of parts and assemblies and hence significantly improve information dissemination. CAD databases for even moderate size companies are expected to be large in size. A product assembly can contain many subassemblies and each subassembly can contain many parts. Therefore, even a small organization that has multiple product lines may add hundreds of assemblies and thousands of parts to their database every year.

Manufacturing companies are constantly looking for ways to reduce costs and the time-to-market. Intuitively, if two products are similar, it is possible to reuse information about one product to derive corresponding information about the other one. In addition to supporting downstream manufacturing and service operations, the part and assembly databases can be very useful during the design phase as well. There are many possible applications where reuse of information can be of significant value.

The following paragraphs describe several scenarios where geometry based search techniques can be useful for product development companies.

- **Design Reuse:** Reusing design/manufacturing information stored would result in a faster and more efficient product realization process. While designing a new part/assembly the designer can refer to existing designs and utilize the components used previously. Let us consider the design of the shaft of a turbine engine. Usually the designer has two options. The first option is to design the shaft from scratch and go through the process and manufacturing planning. The second option is to refer to the database of existing designs, and select an existing shaft and either use it as it is or make minor modifications to it. For example, drill a few holes or cut a few slots in it.

  Reuse of existing designs is beneficial from many different perspectives. It reduces design time by eliminating the need for modeling and analysis for the assembly being reused. Furthermore, the existing assembly is already tested and has an established manufacturing plan. This further reduces the product development time and cost. Sharing assemblies across multiple product lines also allows a company to take advantage of the economy of scale.

- **Cost Estimation Based on Similar Parts:** Accurate cost estimation can take any where from few minutes to few hours depending upon the expertise of the cost estimator and the complexity of the part. Based on our conversations with human cost estimators, it appears that many of them implicitly use estimates from previously completed tasks to generate new quotes.

Manual cost estimation is inefficient especially when the designer submits the 3D model over the Internet for getting quotes. One way to achieve this is to search a database of previously manufactured parts and locate parts similar to the newly designed part automatically, so that the manufacturing cost of the retrieved parts can be used to estimate the cost of the new part. Thus, the ability to quickly find previously manufactured parts similar to query part can be used to facilitate cost estimation.

- **Tooling Manufacturing:** Manufacturing of many parts is a two-step process. During the first step the tool is designed and constructed for making the parts. During the next step, parts are produced using the tool. Often tool makers and manufacturers are two different organizations. Let us consider molding of plastic parts. Selecting a tool maker is an important step in this process. Many different kinds of tools exist that can be used to create plastic parts depending upon the shape of the part. Different tool makers specialize in different kinds of toolings. Therefore, one has to analyze the shape of the part to determine the most appropriate tool maker based on the type of tool needed for the part. Internet-based tool ordering systems give an organization an opportunity to contact a wide variety of tool makers (many of them located in different geographical locations) to solicit quotes from them in order to get the best deal. However, contacting a very large number of tool makers to get quotes is not practical due to the time needed to send the data and analyze the quotes. Therefore, designers and manufacturers often rely on their prior experience to contact the tool makers that have capabilities to handle the new part. This model worked well when designers and manufacturers were dealing with a small number of local tool makers. In the era of global operations and access to a large number of tool makers, designers and manufactures can benefit from software support to help them in identifying potential tool makers.

  A possible way to identify potential tool makers is to find similar parts to the given part and identify tool makers based on the tool makers for these similar parts. The same toolmaker that fabricated the mold for the retrieved plastic part can be approached to provide a mold for the new plastic part. This methodology is currently being practiced by experienced part designers. However, they currently rely on their memory to locate the similar parts. This approach does not work well when the part libraries are very large.

- **Accessing DFMA Knowledge:** A possible use of content based part and assembly search technologies is to provide access to existing design knowledge. Designing assemblies requires considerable effort. Creating good assembly designs require thoughtful analysis and careful application of Design for Manufacturing and Assembly (DFMA) principles. Part and assemblies tools provides two benefits. First, new designers can adopt and copy successful design templates found in the search.

Second, designers can access associated data such as cost, reliability and failure reports.

Archived redesign projects that are driven by DFMA principles can provide meaningful suggestions on how to carry out the redesign in a new project. This way redesign cost will be reduced by exploiting past redesign experiences. We expect future design repositories to include the models of both the initial design and redesign.

We have built a tool that allows searching for parts based on geometric attributes to support the above described applications [1-10]. This report describes main functional capabilities of the tool that performs search based on geometric attributes.

## 2. Part Indexing to Support Geometry-Based Search

This capability involves extracting information (called signatures) from a set of STL files. These signatures are extracted based on the segmentation of part geometry constructed from STL files. We have broken down the signatures into the following categories: overall shape concavity, feature count, edge concavity, shape aspect ratio, large planar features count and surface area, face classifications, and principal moments. Signatures from STL files are extracted off-line to create a repository. The repository management functions include:

- **Repository Creation**: This capability enables users to create repository for a given directory. All STL files in the given directory and its sub-directories are processed. Signatures extracted are placed into a repository file in the given directory. Users have the option to choose whether to extract 2D images from STL files during the process of extracting signatures. These 2D images are used to display search results later on. The process of extracting signatures and images of the 1000 test STL files takes about 3 hours. Users need to provide a directory from which STL files will be loaded, name of a file where the signatures will be saved and the option whether to generate 2D images.

- **Insertion Parts into Repository**: Since extracting signatures from STL files takes a lot of time, it is necessary to extract signatures from a few given parts and insert them into a given repository file. This way, we can avoid rebuilding the entire repository. Users have to provide an existing repository file, the STL files that need to be inserted and the option of whether to generate 2D images for the given parts.

- **Building Part Clusters in the Repository**: In many cases, a lot of parts in the repository will be very similar to each other. We have adopted techniques from database clustering to identify representative parts in the repositories.

## 3. Quick Search

We have implemented a geometry based part search tool to work with STL files. The current version of the implementation is able to perform segmentation on STL files and use the segmented facet sets to derive information about part faces. The ability to work with STL files ensures that our system can now work with every CAD system that is able to export STL files.

Statistics have been collected from part signatures such as angles between adjacent facets in each patch and areas of different types of patches such as planar, rounded, cylindrical etc. In order to account for similar parts with different dimensions, these filters are mostly using ratios of signatures. For example, we use the ratio between convex hull volume and part volume instead of convex hull volume or part volume alone. When computing the distance using these filters, we also normalize the difference between the query part and parts in the database to eliminate the factor of absolute difference. The system uses a hybrid approach to compare a query part to the parts in the database. It first applies sequential filtering. Some of the filters have a cut-off threshold. So if any distance component of those filters exceeds the cut-off threshold, then the part is pruned. If a part is not pruned, then individual distance components are combined into a weighted average. A cut-off value is also used on the overall weighted average to perform additional pruning.

## 4. Run-Time Customization of Quick Search

We have implemented run time customization capabilities in the Geometry-Based Search system to customize quick searches. These capabilities allow the users to tailor the search performance based on their needs. Performance of the search can be customized via three adjustments:

- Search criteria weight adjustment: allows the user to change the importance of some of the filters by increasing or decreasing their weights.

- Filter cutoff adjustment: allows the user to increase or decrease the number of results returned. This also influences the search time.

- Distance correction: allows the user to change the distance threshold for each filter. Only those distances larger than the threshold are considered in the overall average distance.

Users can use these customization methods to guide the search. For example, if shape aspect ratio is a big concern, users can increase the weight of corresponding measures to make parts with similar shape aspect ratio more favorable than other parts in the database. Figure 1 shows the quick search results for one of the query parts. Notice the less than ideal placement of relatively similar parts P531.stl and P571.stl. Figure 2 shows the results for the same query after changing the face classification filter correction value from 0 to 11. Figure 3 shows the quick search results for a second query part. Notice that only a few parts are produced. Figure 4 shows the results after changing the average distance cutoff value from 40% to 45%. Finally, Figure 5 shows the quick search results for the first query part. Notice again the less than ideal placement of similar parts P531.stl and P571.stl. Figure 6 shows the results after changing the weight of the principle moments filter from 50% to 30%. Parts P531.stl and P571.stl are now given much better ranking.

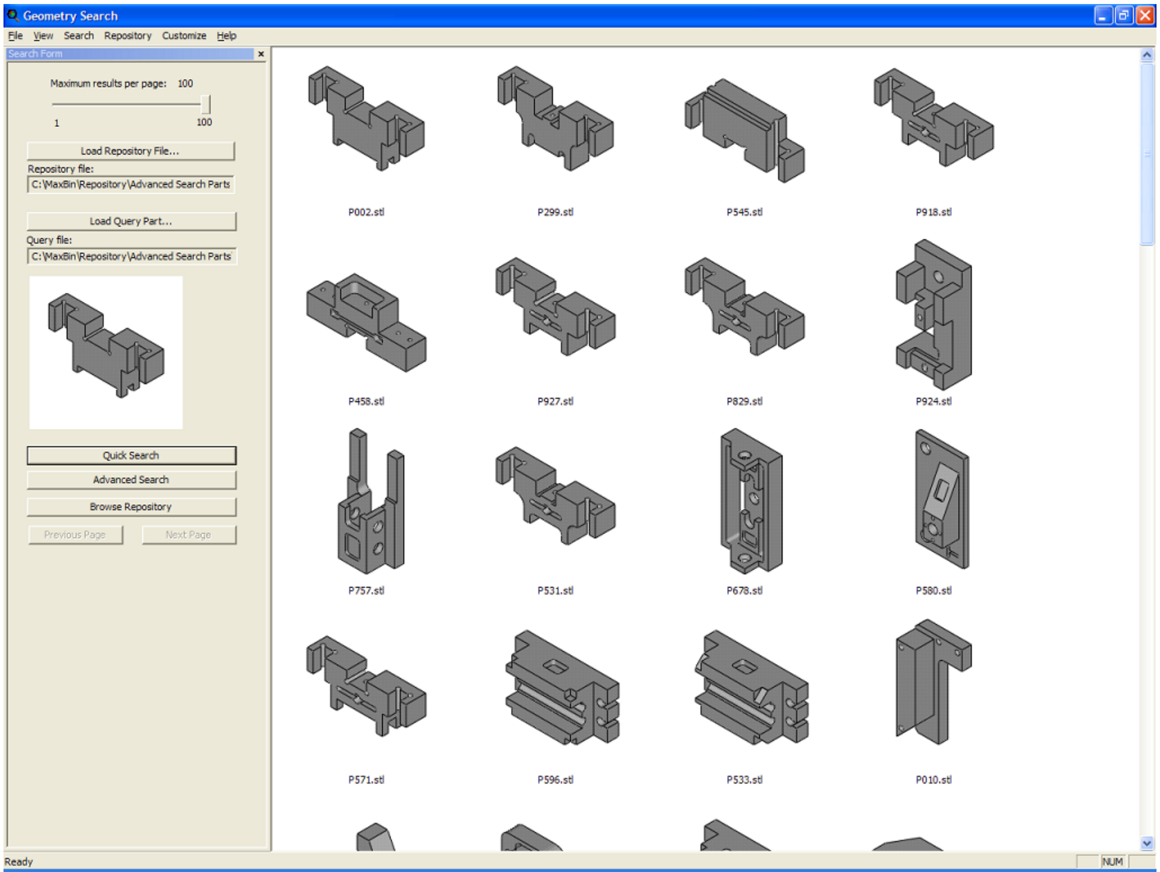


Figure 1: Quick Search Results before threshold adjustment

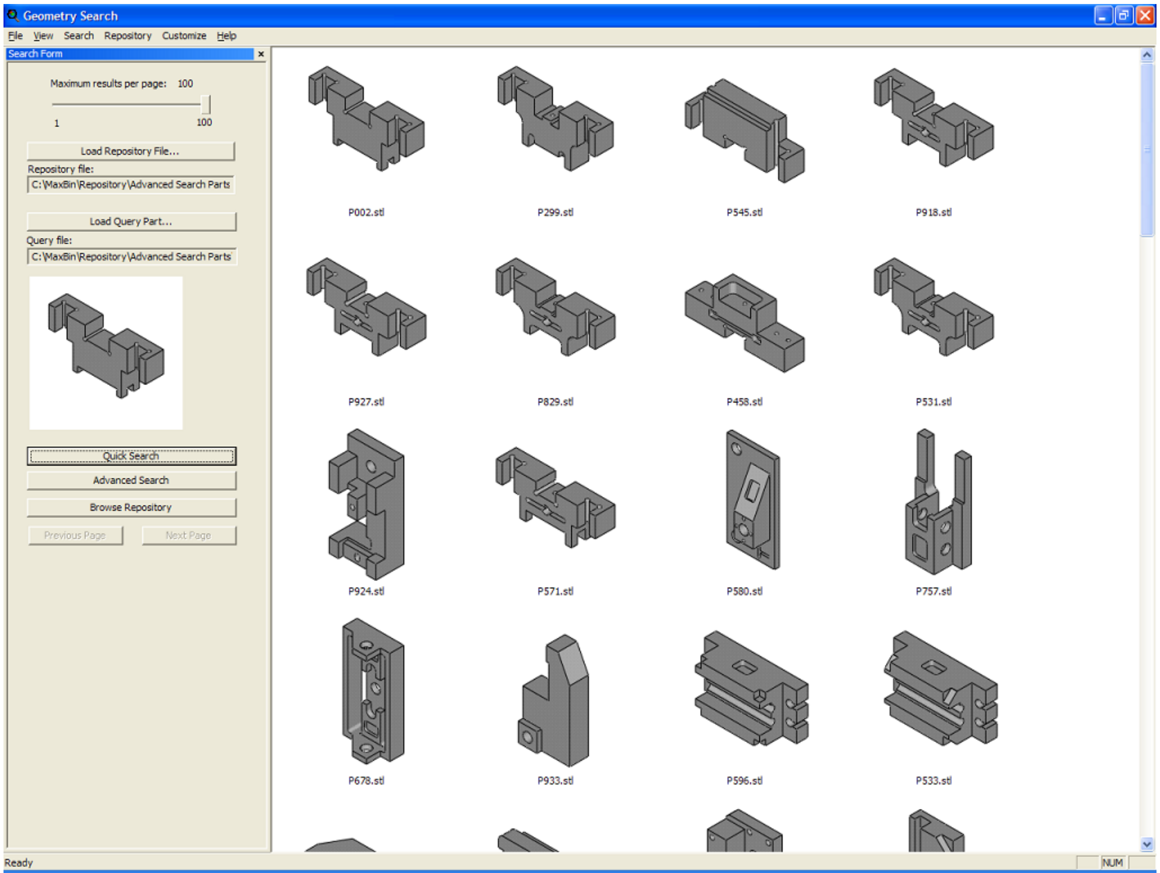


Figure 2: Quick Search Results after threshold adjustment

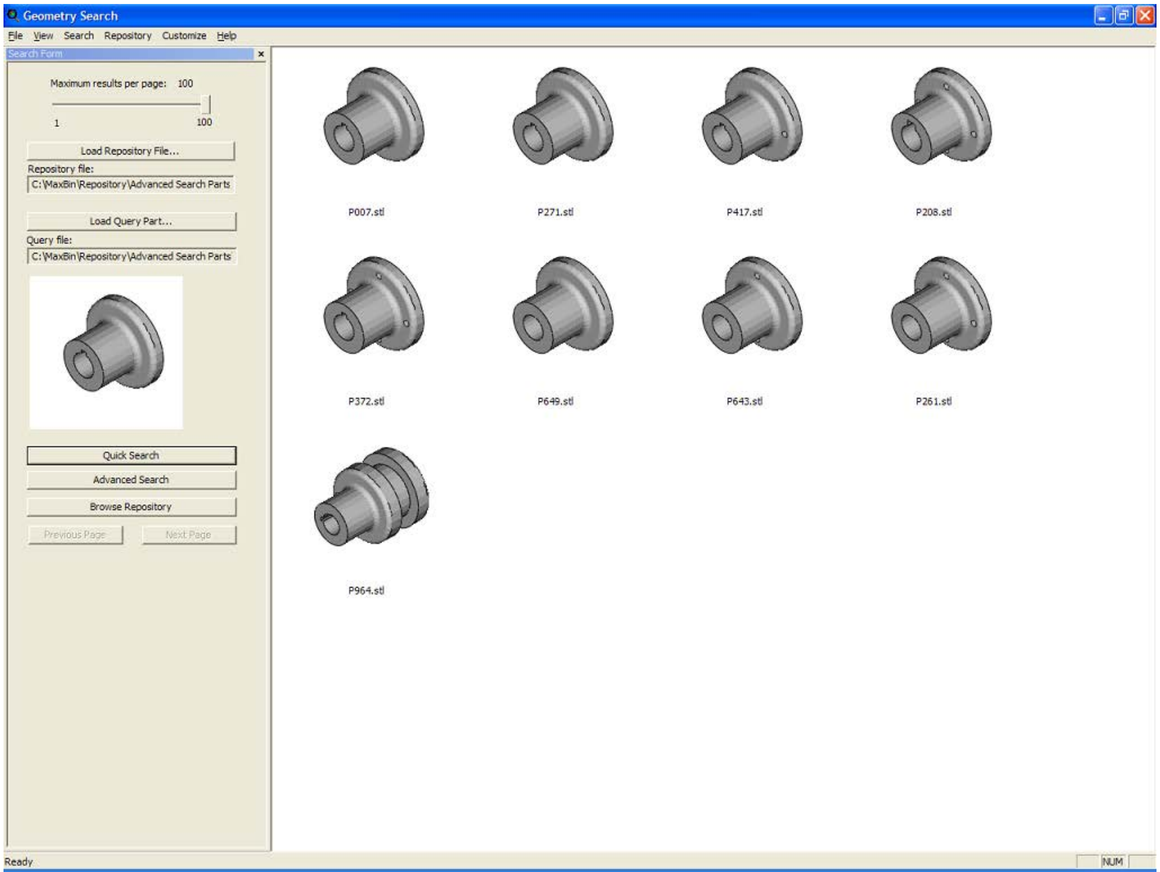


Figure 3: Quick Search Results before average cutoff adjustment

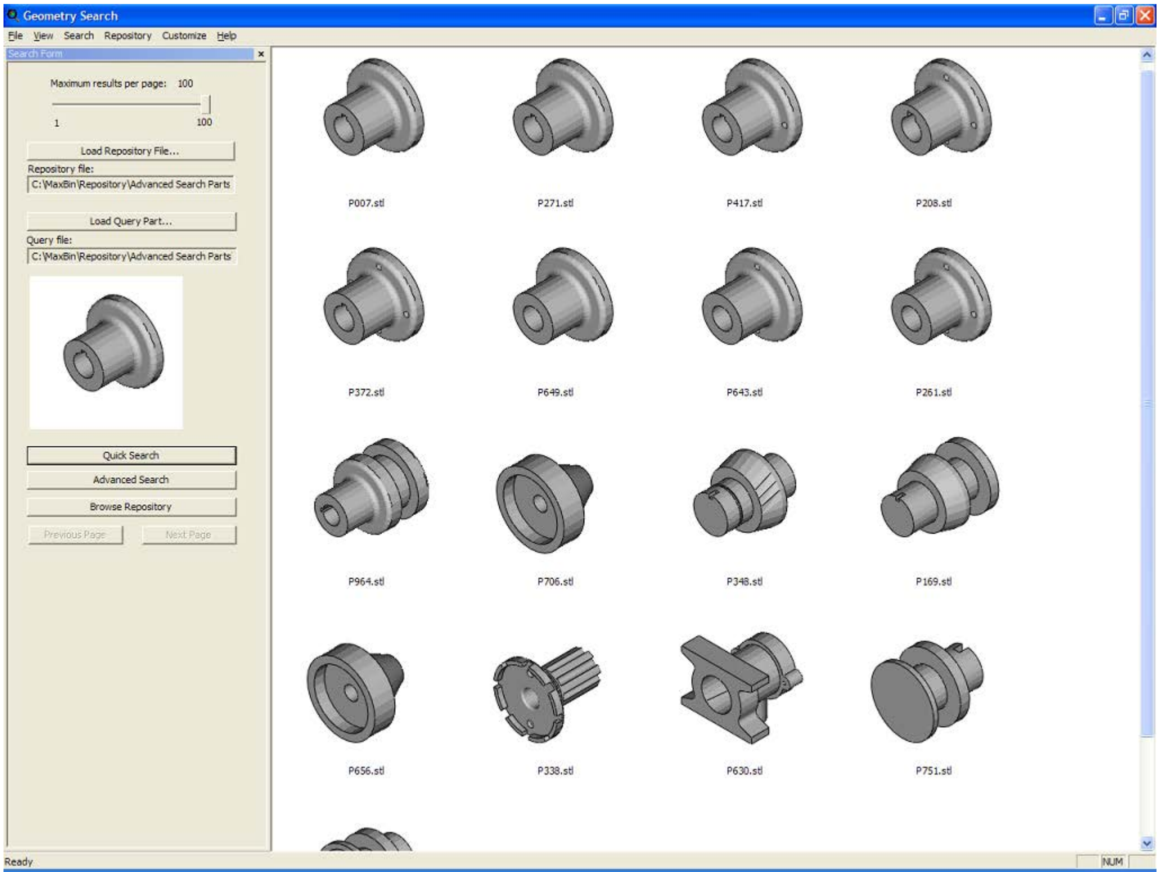


Figure 4: Quick Search Results after average cutoff adjustment

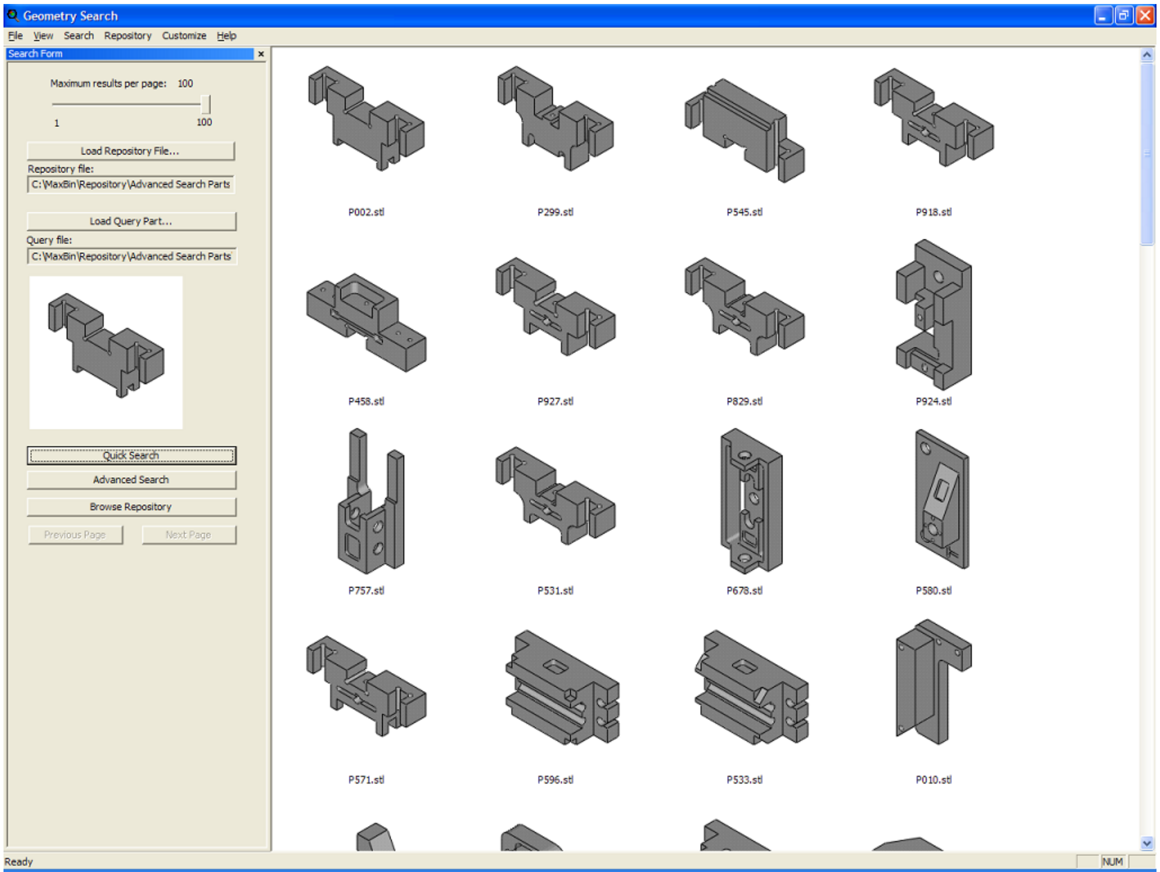


Figure 5: Quick Search Results before filter weight adjustment

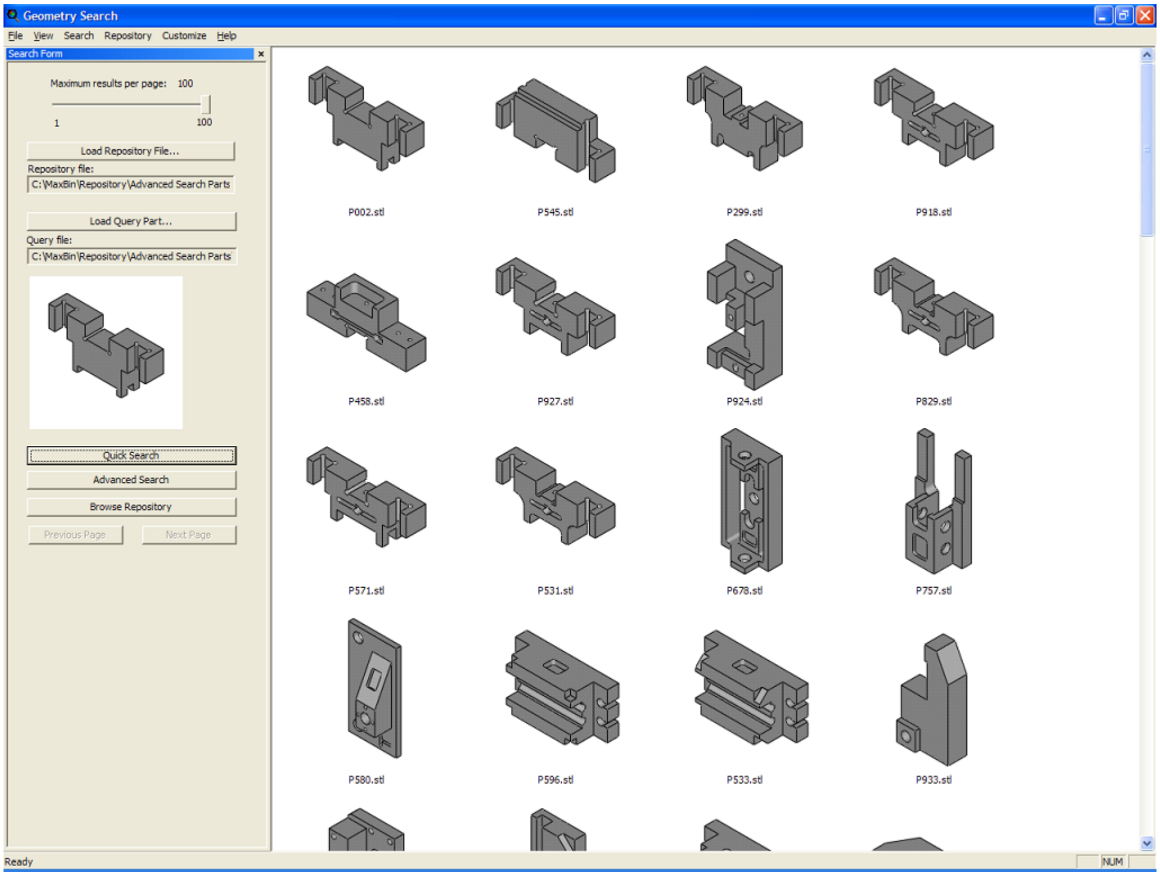


Figure 7: Quick Search Results after filter weight adjustment

## 5. Query by Example Capability

We have implemented a capability to support query by example over large part repositories. The rationale behind developing this capability is as follows. In many part retrieval situations, the user may not have a 3D model of a query part. An example of such a scenario is design reuse. In such a scenario, a user may be interested in using an existing design with minor modification to reduce the 3D modeling time. In this scenario, the user will not have a detailed part model to begin the search. Requiring the user to create a detailed part will defeat the purpose of the design reuse. So in this scenario, a better alternative for the user is to browse the repository and locate the parts by visual inspection. This alternative is expected to perform better than sketch based search functionality. However, browsing a large repository is likely to present its own set of challenges. For example if a repository contains fifty thousand parts, then browsing the entire repository can take many hours. In such cases, a better alternative is to first create a repository of representative parts and allow the user to browse the representative repository first. Once the user finds a part close to his/her needs in the representative repository, then the user can use the selected part as a query part and use the entire repository to locate all similar parts. This two step process will allow the user to navigate large repositories effectively.

A key step in deploying the above described strategy is the development of techniques for identifying the representative repository. We have adopted techniques from database clustering to build representative repositories. We use pair-wise distance between parts as a basis to begin the clustering process. Parts that have a small distance between them are added to the same cluster. The cluster sizes are controlled by specifying a distance cut-off value. The maximum of the pair-wise part distances in a single cluster cannot exceed a user specified distance cut-off value. Once the clusters have been built, a representative part is selected from each cluster. This is done by selecting the part in a cluster that has the minimum average distance from the other parts in the cluster. Representative parts from each cluster are collected together to form the representative repository. The distance used in building the clusters is customizable and can be adjusted by users based on their needs.

## 6. Advanced Search

Distance measures based on coordinate invariant signatures usually produce large distance between dissimilar parts and small distance between similar parts. However, there are cases when coordinate invariant abstractions may produce small distance between dissimilar parts. Hence these are quite useful for part pruning. However, we need new techniques with better discrimination capabilities for getting accurate search results. Distance measures that utilize features' position and orientation usually provide better discrimination capabilities. However, they are dependent on coordinate systems in which parts are represented. Hence, an alignment step is needed that transforms the parts to poses that lead to the minimum distance. After alignment, dissimilar parts always result in large distances while similar parts always result in small distances. Hence this approach leads to a better discrimination capability. Due to the computational nature of alignment, alignment based search is slower than search without alignment. Thus we use it as the second level search on the results returned by other search filters. For each candidate part, we select large planar patches of the query part and the part in the pool. Next, we generate the combinations of possible alignments for the patches from different parts. For each possible alignment, we compute the minimum difference between all planar patches of the query part and the candidate part. Finally, we pick the minimum distance among all combinations as the minimum distance between the two parts. A similar method is used to perform alignment of cylindrical patches. By using alignment of both planar and cylindrical patches we are able to produce very good discrimination between similar parts and dissimilar parts.

In order to align patches, a part mesh must first be processed and its various patches extracted and classified. Our current patch extraction method extracts and labels two types of patches: planar and curved. No further processing is needed to perform alignment of planar patches, since a single normal can be extracted from a planar surface to be used in alignment. Additional processing must be carried out, however, to find cylindrical surfaces and extract a single vector representing the surface. Our algorithm for finding cylindrical surfaces involves going through each patch labeled as curved and ascertaining whether the facets that make up the curved patch have a mostly cylindrical arrangement. That algorithm first visits every facet and places the area of each facet into a bucket that matches the facet's normal. The buckets are sorted in descending order based on surface area. The algorithm then chooses the largest bucket and uses it as a reference. It then visits every other bucket and computes the cross product between the normal of the reference bucket and the normal of each visited bucket. The results of the cross products (resulting vector and the surface area of visited bucket) are placed in a separate set of buckets which are sorted by surface area in descending order. If the largest bucket contains 90 percent of the surface area, then the patch is mostly cylindrical and the vector associated with the biggest bucket is used for alignment.

Figure 7 shows the quick search results for one of the query parts. The results are sorted from left to right and from top to bottom in the order of similarity to the query part. Notice the two parts P022.stl and P066.stl come before (to the left of) part P792.stl even though they are clearly less similar to the query than P792.stl. Quick search is designed to act on a large set of parts and produce results quickly. It is not powerful enough to very accurately discriminate between similar and dissimilar parts. Advanced search is slower and it is designed to act on a smaller set of parts. Its accuracy, however, is much better. Figure 8 shows the results of the advanced search for the same query. All less than ideal parts have been eliminated from the previous list by advanced search. Advanced search has, in fact, such good discrimination that when it processes all the results of the quick search and sorts them by similarity, there is usually a very significant distance gap between the last similar part and the first dissimilar one. Our system takes advantage of this characteristic of advanced search by dynamically searching for such a gap and establishing a cutoff point just after the last similar part. This way advanced search very often successfully eliminates all the dissimilar parts and keeps all the similar ones, in effect avoiding being too aggressive and eliminating any of the similar parts. If the gap cannot be found, then the system uses a user specified cutoff point. Figures 9 through 11 demonstrate the performance of the advanced search on few other parts.

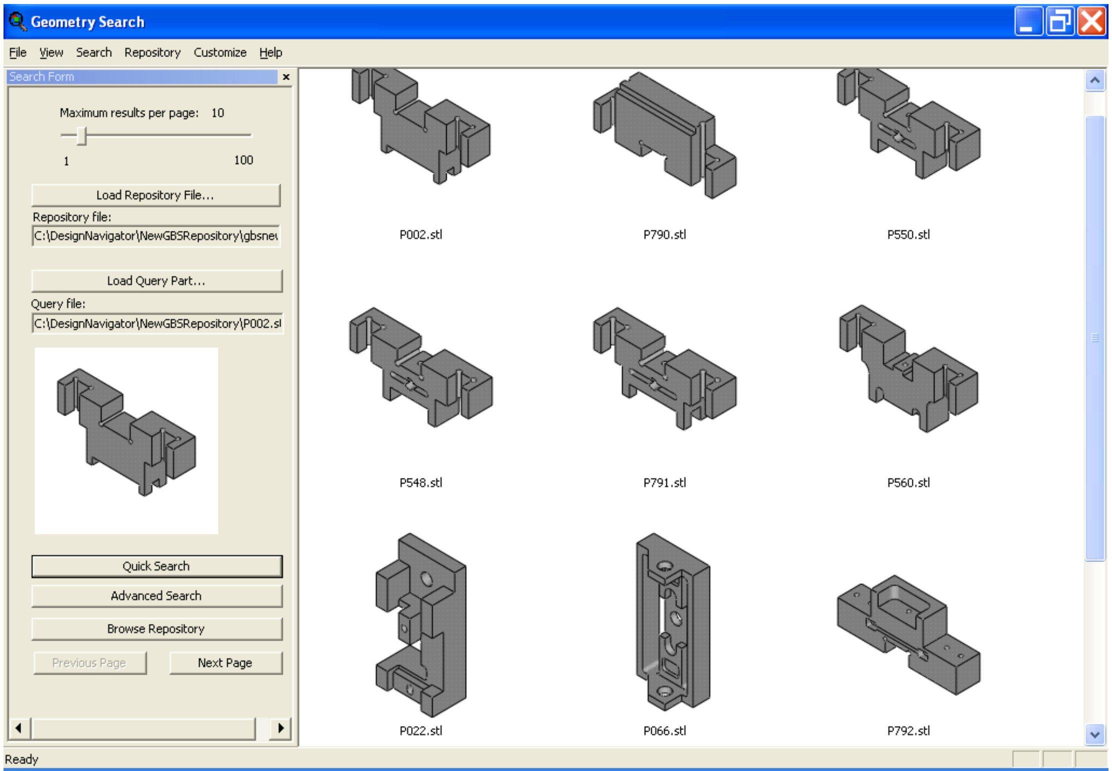


Figure 7: Quick search results

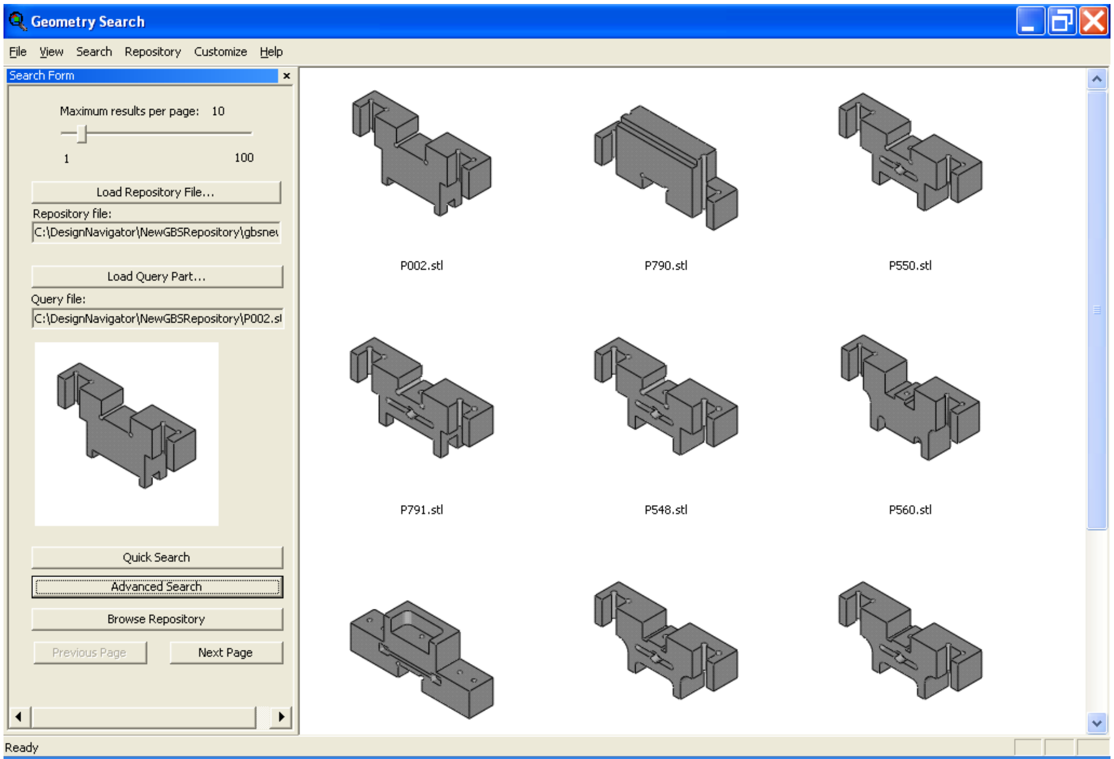


Figure 8: Advanced search results

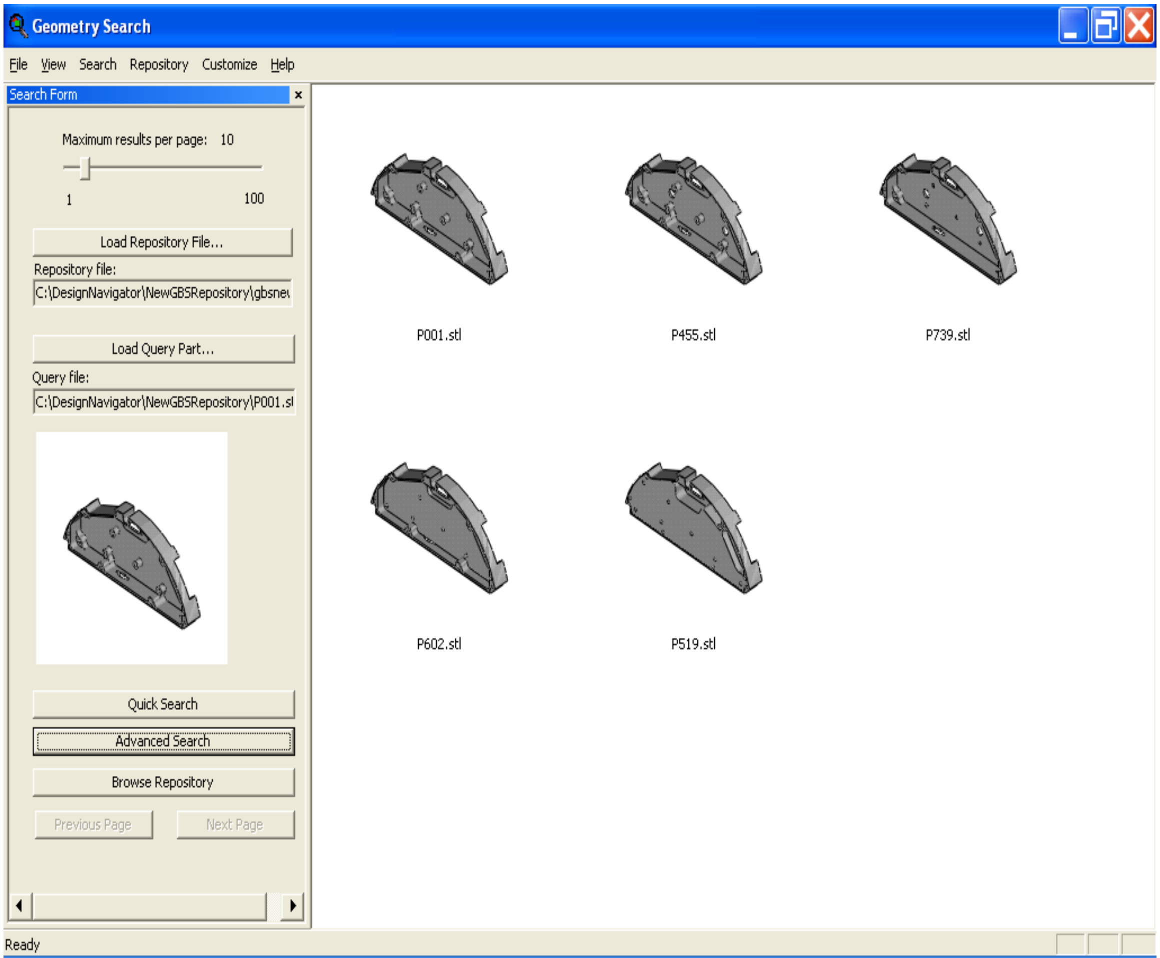


Figure 9: Advanced search results

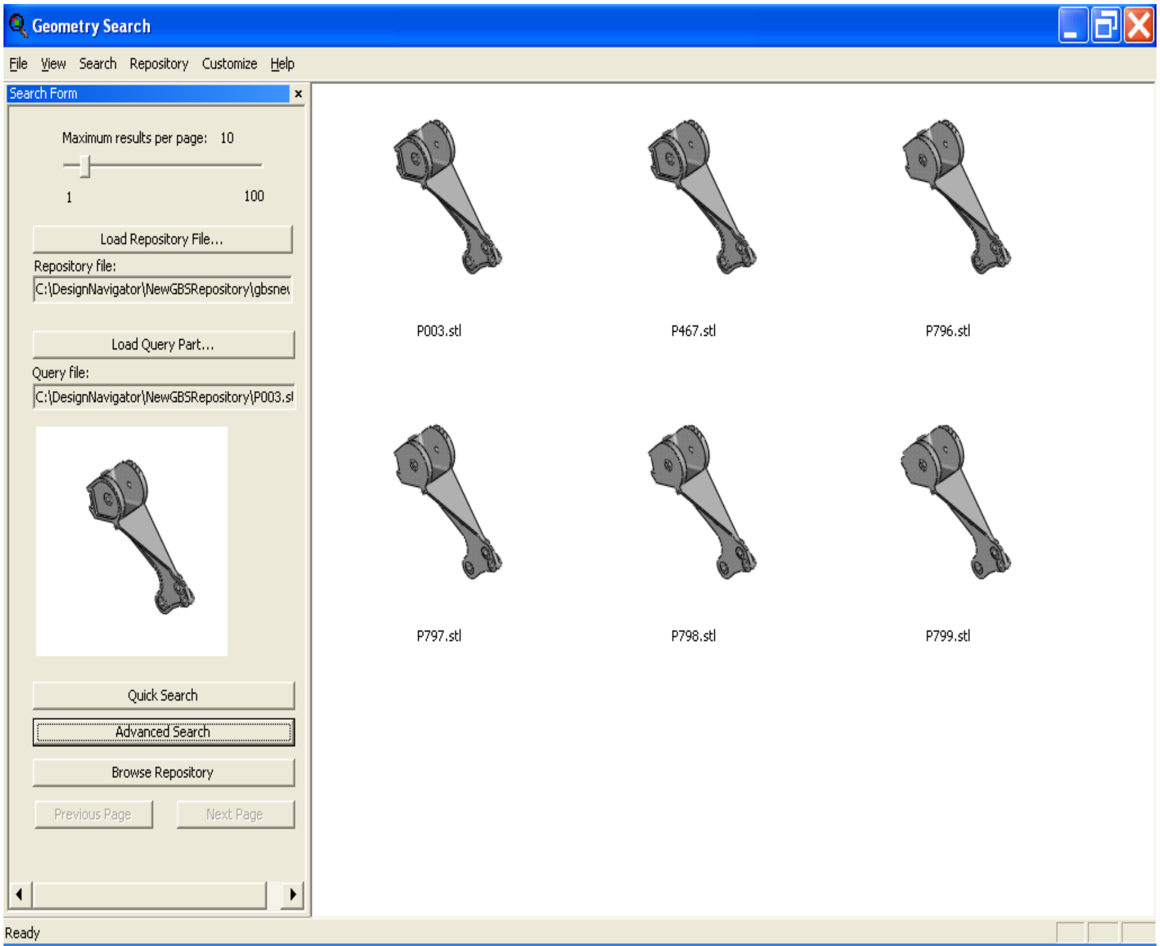


Figure 10: Advanced search results

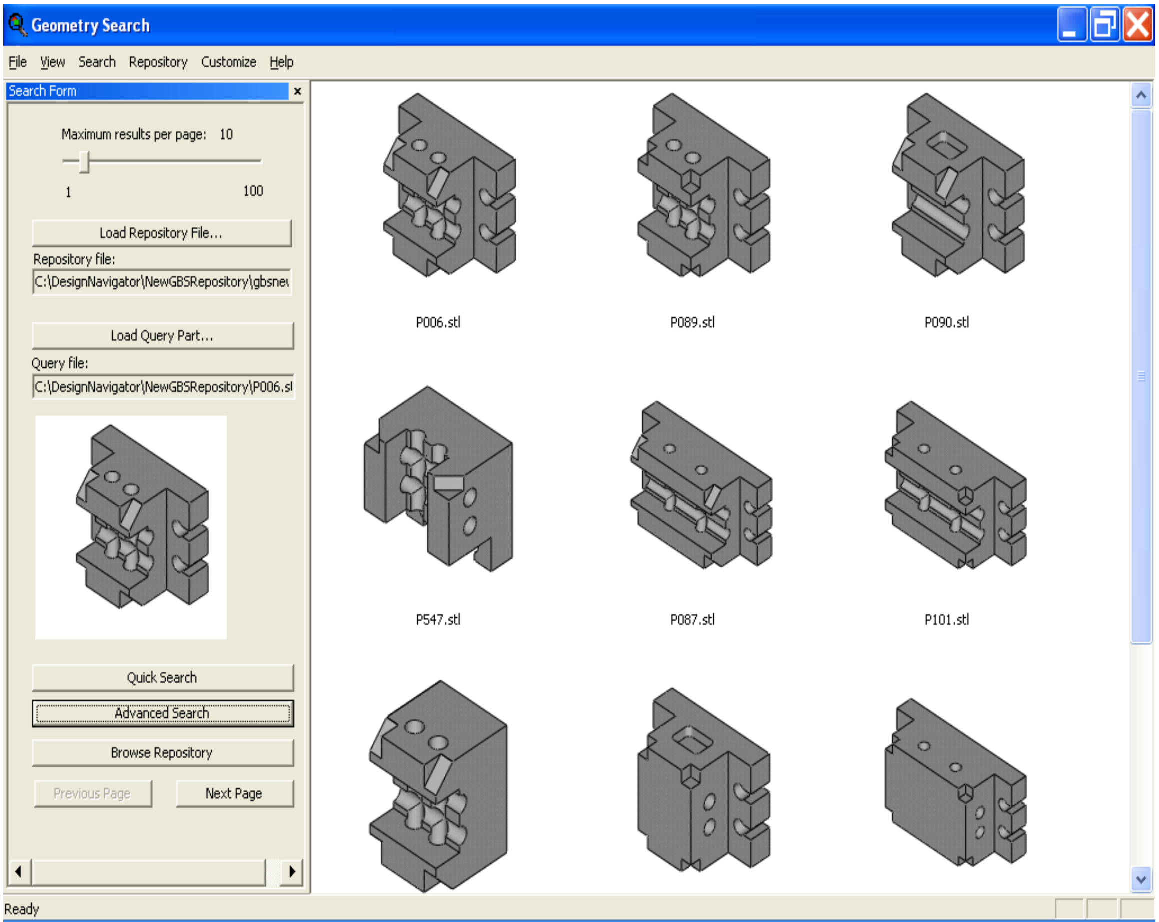


Figure 11: Advanced search results

## References

[1] Cardone, S.K. Gupta, and M. Karnik. A survey of shape similarity assessment algorithms for product design and manufacturing applications. *ASME Journal of Computing and Information Science in Engineering*, 3(2):109-118, 2003.

[2] Cardone, S.K. Gupta, and M.V. Karnik. Identifying similar parts for assisting cost estimation of prismatic machined parts. *ASME Design for Manufacturing Conference*, Salt Lake City, UT, September 2004.

[3] Cardone and S.K. Gupta. Similarity assessment based on face alignment using attributed vectors. *CAD Conference*, Phuket Island, Thailand, June 2006.

[4] Cardone, S.K. Gupta, A. Deshmukh, and M. Karnik. *Machining feature-based similarity assessment algorithms for prismatic machined parts*. Computer Aided Design, 38(9):954-972, 2006.

[5] A.S. Deshmukh, S.K. Gupta, M.V. Karnik, and R. Sriram. A system for performing content-based searches on a database of mechanical assemblies. *ASME International Mechanical Engineering Congress & Exposition*, Orlando, FL, November 2005.

[6] A.S. Deshmukh, A.G. Banerjee, S.K. Gupta, and R. Sriram. Content-based assembly search: A step towards assembly reuse. *Computer Aided Design*, 40(2):244-261, 2008.

[7] S.K. Gupta, A. Cardone, and A. Deshmukh. Content-based search techniques for searching CAD databases. *CAD Conference,* Phuket Island, Thailand, June 2006.

[8] M.V. Karnik, S.K. Gupta, and E.B. Magrab. Geometric algorithms for containment analysis of rotational parts. *Computer Aided Design*, 37(2):213-230, 2005.

[9] M.V. Karnik, D.K. Anand, E. Eick, S.K. Gupta, and R. Kavetsky. Integrated visual and geometric search tools for locating desired parts in a part database. *CAD Conference*, Bangkok, Thailand, June 2005.

[10] M.V. Karnik, S.K. Gupta, E.B. Magrab. Geometric containment analysis for rotational parts. *ASME Design Automation Conference*, Salt Lake City, UT, September 2004.