

Integrated Visual and Geometric Search Tools for Locating Desired Parts in a Part Database

M. V. Karnik¹, D. K. Anand², E. Eick³, S. K. Gupta⁴ R. Kavetsky⁵

¹Iktara and Associates LLC, mkarnik@iktara.com

²University of Maryland, College Park, dkanand@eng.umd.edu

³Fachhochschule Manheim, chaostrooper@chaostrooper.de

⁴University of Maryland, College Park, skgupta@eng.umd.edu

⁵Naval Surface Warfare Center, robert_kavetsky@onr.navy.mil

ABSTRACT

Manufacturing companies often need to reuse existing design information to reduce cost and time-to-market. This requires search tools that are capable of locating previously designed parts. Many geometry-based search techniques have been developed to locate parts similar to the input query part. But these techniques are not effective if the user does not have a query part to provide as input to the geometry-based search system. This paper describes an integrated search system for searching a part database both visually and geometrically for locating and retrieving parts. The visual and geometry-based search tools of this integrated system are highly interconnected and provide a seamless transition between either modes of searching. The visual search tool uses level-of-detail techniques to display a large number of parts in the scene. It has multiple navigation and sorting utilities to search the database visually. The geometry-based search tool uses gross-shape based search techniques to provide effective searching capabilities. The system described in this paper has been tested with a wide variety of parts.

Keywords: Geometric Reasoning, Information Retrieval, Visual Search, and Shape Analysis.

1. INTRODUCTION

Nowadays, manufacturing companies have large databases of CAD models of previously designed parts. Designers at these companies spend a lot of their time in searching for previously designed parts so as to reuse their information [2-3]. There are many possible decision making applications where reuse of information can be of significant value. Representative examples include part-family formation, redesign suggestion generation, supplier selection, cost estimation, tooling design, machine selection, stock selection, and design reuse. Currently, most designers search the part databases manually to locate parts that they desire. Thus a system that can search for parts in a database is expected to help companies in significantly reducing the time and cost associated compared with the manual methods of locating similar parts.

Suppose a designer would like to locate a part in the design database consisting of thousands of parts for reusing design information. There are two possible cases. In the first case the designer has a query part and wants to locate a part similar to the query. In such cases, geometry-based search techniques are useful for locating similar parts. However, in the second case the designer may not remember the exact geometric details of the part to locate it through the geometric search techniques. In such cases, the user will need to locate the desired object by browsing through all the databases. Once the designer locates a part similar to the desired part he/she can use that part as the query in geometry-based technique. Thus an integrated system that can assist the designer in locating similar parts by providing geometric as well as visual search is useful.

The following scenario illustrates the need for the geometric and visual search tools for reusing design information. Bob is a senior designer in a manufacturing company. After 10 years of service he leaves the company. His folder consists of 936 parts. Bill has taken over Bob's projects. Bill wants to design a shaft of a turbine engine. He has two options. The first is to design the shaft from scratch and go through the process and manufacturing planning steps. The second option is to select an existing shaft from one of Bob's previous projects. However, he does not know the file name of the part. The only possible way he can search is by examining all the relevant project folders and opening

each part file one-by-one until the part is found. If the 486th part file were the one he is looking for then it would have taken him, approximately, 2 hours and 45 minutes to open 486 part files in a standard CAD system. Instead if Bill had visual and geometric search tools he would be able to browse through the part database using the visual search tool select a suitable part and search for similar parts using geometry-based search tool.

The current shape-similarity assessment techniques can be broadly classified into two categories: feature-based techniques [3-5][8][12-13], and gross shape-based techniques [1][6-7][9-11][14-16]. Most modern 3D CAD/CAM systems (e.g., Pro/Engineer, Unigraphics, etc.) allow users to define features. However, feature-based representation is not always available and is mostly available in CAD system specific format. Therefore, feature-based techniques cannot be always used. Boundary representation (B-Rep) is more widely available and there are standards such as STEP, IGES that have been defined to make B-Reps CAD system independent. We have developed both feature-based as well as gross-shape based approaches. Our work on feature-based approaches is described in [3][8]. This paper describes our work on gross-shape based approach that uses information obtained from B-Rep.

While using gross-shape based techniques, it is difficult to select a distance function that matches the applications needs. Thus, each technique results in some false positives and false negatives depending upon the strictness of the comparison. If the number of search results is low, then the number of false negatives is high. This results in loss of useful data during the search. If, however, the search yields large number of results then the number of false negatives is low but the number of false positives is high. The user needs to browse through the results carefully to locate part. So a system that can help the user to navigate the data visually by providing useful filtering and navigation utilities is needed.

This paper describes an integrated search system consisting of geometric and visual search tools for locating similar parts in a database of parts. The user can specify some initial filtering criteria and load all the parts that fit the criteria in the visual search tool. While browsing the scene the user can search for parts similar to any part in the scene using a geometry-based search tool. Visual search uses filters to prune parts from database while geometry-based search uses a reference part to search for similar parts.

The search system described in this paper is easy-to-use and uses efficient techniques for performing searches quickly. The integrated system is very effective in searching as it allows the user to switch between geometry and visual search at any time. The system has been tested with a wide variety of parts and it is able to find similar parts in a matter of few seconds.

This paper is organized as follows. Section 2 gives an overview of the system and describes the system in use. Section 3 provides the capabilities of Visual search tool. Section 4 describes the signatures used in the geometry-based search tool. Section 5 provides the implementation details and results. Section 6 provides the concluding remarks.

2. OVERVIEW OF THE SYSTEM

The integrated system described in this paper consists of both geometric and visual search tools. Visual search tool is capable of loading parts that fit particular size/complexity criteria into the system and then displaying all the parts in a single scene. The user needs to specify only the high level directory. The system searches all the sub-directories recursively, loads the relevant files, and displays them in the scene. Multiple navigation and sorting utilities have been implemented in visual search to aid the user in visually locating the part that he/she is looking for. Some basic size and complexity based filters have also been implemented to reduce the size of the database. The user can select any part from the scene and perform geometry-based search on it to locate parts similar to the selected one. The results of the geometry-based search can be launched in another scene of visual search for further visual navigation of the results. Figure 1 shows the activity diagram of the integrated system. The actors in the system are the user, visual search tool and geometry search tool. The user can take any of the following courses to perform the search.

- **Course 1:** Browsing the database using only visual search. The user needs to specify only the top-most directory in which all the parts are located. The user may also specify some size and complexity criteria. The visual search tool recursively searches the sub-directories and loads all the parts that fit the criteria into the system. However, only a limited number of parts (defined by the user) are displayed in the first view of the scene. The user can browse through the parts using the mouse to closely observe all the parts that he/she is interested in. The user can change the arrangement of the parts from planar configuration to cylindrical configuration and vice-versa. If the

Actors: 1) User,
2) Visual Search,
3) Geometry Search

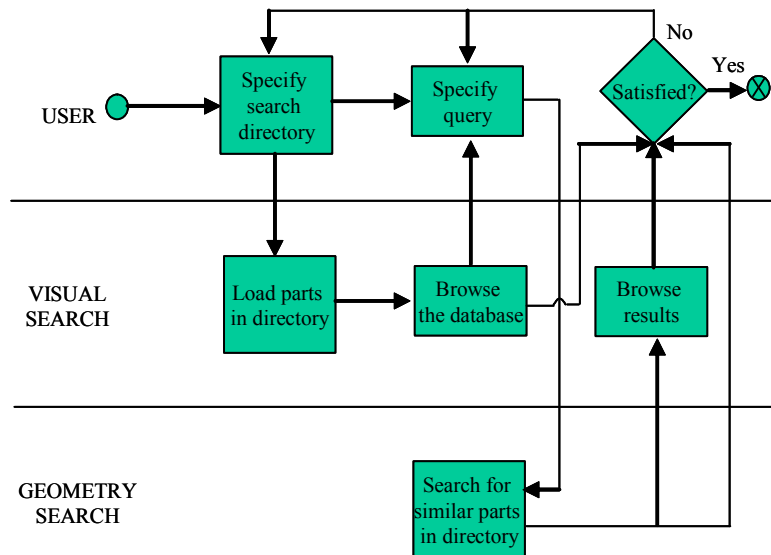


Fig. 1. Activity diagram for Integrated Search System with partition based on actors.

user wants to ensure that he has visually observed all the parts and not missed any, then he/she can switch to 'Auto-panning' mode described in the Section 3.

The user can sort the scene based on size such that the parts are arranged in increasing order of size. The large sized parts are shown in red. The user can select multiple parts from the scene and delete them. This reduces the number of parts that needs to be scanned. Suppose the user knows the approximate size of the part that he is interested in. Then, the user can also filter the remaining parts based on size/complexity. The visual search tool keeps the parts within that range and deletes the remaining parts.

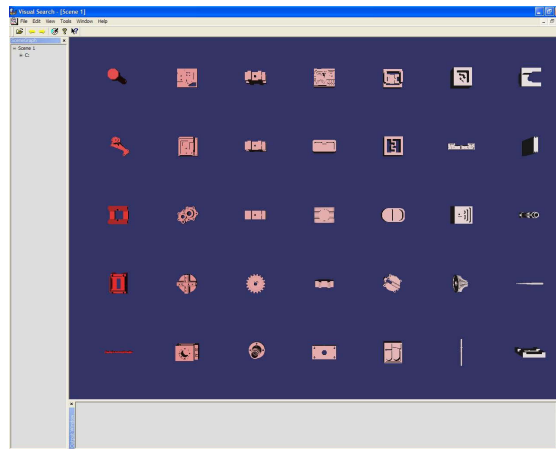


Fig. 2. Visual Search tool with parts sorted based on size.

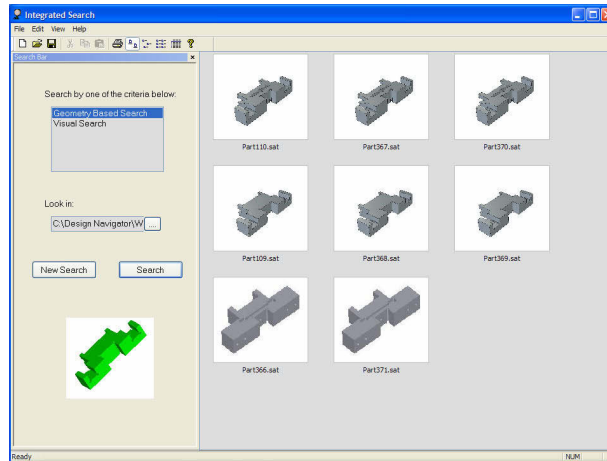


Fig. 3. Geometry-Based Search tool.

While browsing through the database, the user can mark a single/multiple parts and launch them in another scene for closer observation. Once the user has exhausted all parts in one view of the scene he can go to the next view to search the next group parts visually using the different utilities described above. Figure 2 shows the visual search tool with large parts that have red color. The parts have been arranged in a planar configuration.

- **Course 2:** Searching the database using only geometry-based search. The user needs to specify the directory containing the previously designed parts and a query part. When the user click search the geometry-based search tool searches for parts similar to the query by using some gross-shape and detailed shape complexity statistics signatures described above. Figure 3 shows the results of the geometry-based search tool.
- **Course 3:** Using a both visual and geometry-based search. The user can select a part from the visual search tool and launch it in geometry-based search tool to look for parts similar to the selected part. Figure 4 shows the selected part that has been launched in the geometry-based search module. The database parts are shown in the right window. These parts are loaded from the same directory that has been specified in the visual search tool. The parts that had been filtered/deleted from visual search are not loaded in geometry-based search. The user can then perform geometry-based search on the database. If the number of search results is high the user can launch the parts in visual search tool and navigate through them to locate the part that he/she is interested in.

3. VISUAL SEARCH CAPABILITIES

Visual Search tool uses a scene graph to display all the part files in a single scene. A scene graph is a directed acyclic graph consisting of heterogeneous nodes that represent the elements of a scene. The scene graph helps in representing the elements of the scene as objects and promotes object oriented programming. The typical nodes of a scene graph include geometry, material/property, camera, light and some other nodes. The scene graph in Visual Search is used to perform the following operations.

- **Culling:** The scene graph helps in reducing the number of objects being drawn in the scene by identifying those objects that are being occluded by other elements of the scene. This is termed as visibility culling and reduces the amount of data that needs to be rendered.
- **Level-of-Detail:** The scene graph provides tools for implementing level-of-detail techniques. Objects in the scene that are close to the view-point are rendered in detail while the objects that are far away from the view-point are rendered with less detail.
- **Rendering:** The scene graph describes the state of the objects in the scene and helps in ordering the rendering correctly.

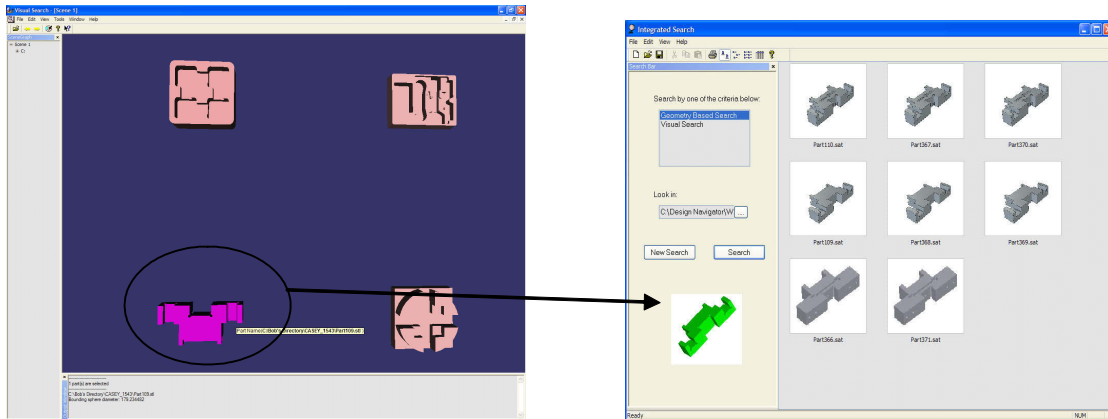


Fig. 4. Part selected in visual search (pink) and launched in geometry-based search.

The description of each capability of Visual Search tool has been described in this section. The following capabilities have been implemented.

- **Examining mode:** Figure 5 shows Visual Search's mainframe displaying several parts in cylindrical configuration. The user can navigate through the scene using trackball manipulator. The user can change the orientation of viewpoint by dragging the mouse while the left-button is down. She/He can zoom and zoom out by dragging the mouse with right button down. Once the user locates the part she/he is interested in, she/he can select and isolate it by double-click of the left-button of the mouse.
- **Auto panning:** Auto panning provides a convenient way to navigate through the parts without any interaction from the user. The user only has to wait until the part being searched appears on the screen. Auto panning can be started and paused by pressing the space bar. In cylindrical organization the shell rotates about its axis and simultaneously translates along the axis. This results in a spiral motion. The direction of translation is changed when the all parts in the view are traversed. The translation speed can be adjusted by the factor "levels per rotation". The rotation speed can be adjusted with the mouse wheel.

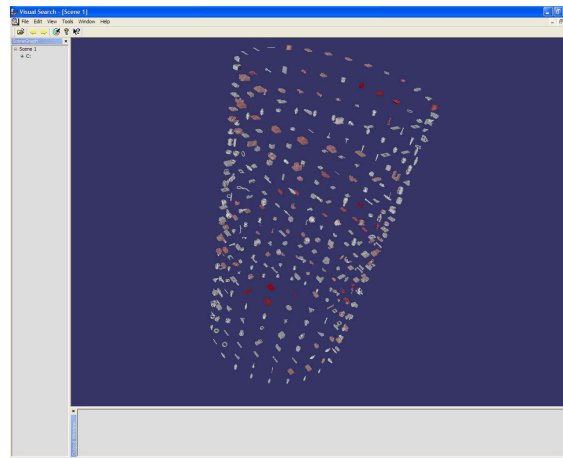


Fig. 5. Visual Search with cylindrical view of parts.

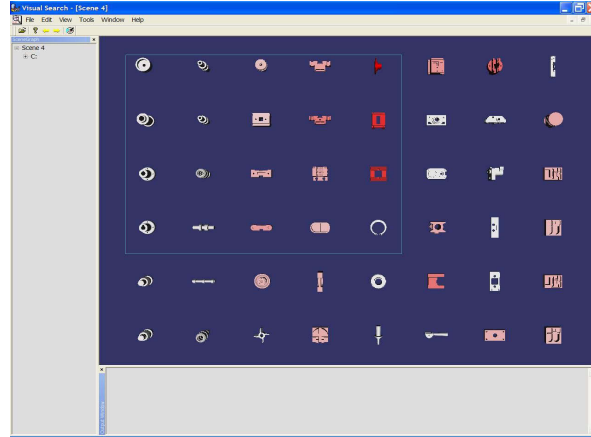


Fig. 6. Rectangular region for selecting parts.

- **Arrangement of the parts:** There are two different ways of organizing the parts on the screen. This should give the user freedom to choose the best form of displaying the files. This depends on factors like size of the screen, the number of parts and the preferences of the user.
 1. **Planar scene:** The parts are organized like the fields of a check board. The distance between a part and its three neighbors is four times the size of the part. Because of its 2D structure this organization is used best for viewing all parts on one sight, only if the number of parts is not too high.
 2. **Cylindrical scene:** In this configuration all parts are placed on the shell of a cylinder. The user can define the radius r and the number of parts n on each level. The number of levels is then given by the n/N , where N is the total number of parts being loaded. The height is then calculated using a suitable separation between the levels. The angular separation between each part is $\theta=2\pi/n$. The viewpoint is set in the center of the cylinder with orientation towards the shell.

All parts are rescaled to the same size after loading. This prevents overlapping when displaying them and also ensures that large differences in size do not affect the visualization.

- **Part management:** The user can load single or multiple files or a whole directory. The user can use the “drag and drop” functionality by selecting files in the file manager and dragging them with the mouse over the empty Visual Search mainframe. Parts can be added to an already existing document by using the “Add...” function from the file menu or simply by dropping a selection from the file manager.

Searching for distinct parts in the view can be simplified by minimizing the number of shown parts. To do this the user can select parts, which definitely do not meet his criteria and delete them by pressing the “Del”-key.

To ensure that Visual Search works efficiently on machines having both low as well as high graphics processing power, a user-defined limit on the number of parts in a scene has been implemented. If the user loads more parts than the limit, only the initial parts up to the limit get displayed. To display the remaining parts, the user needs to click on the ‘Next View’ button. Then, the parts in the current view are deleted and the parts in the next view get displayed. The user can view the previous set of parts by clicking on the ‘Previous View’ button.

- **Part selection:** To mark a part it has to be selected by double clicking it with the left mouse button. When the “Ctrl”-key is pressed while picking parts, the selection can be expanded or reduced. To indicate the selection the part is highlighted and a window, with additional information pops up. The user can unselect everything by double clicking the left mouse button in the empty background of the scene. For further examination of parts the selection can be opened in a new window by pressing the “Enter”-key.

A utility for selecting more than one part in the scene has been implemented. The user can select multiple parts at a time and launch them in a new window or delete them from the scene. To select multiple parts the user needs to keep SHIFT button pressed and move the mouse to create a rectangular region. Once the user releases the SHIFT button the parts within the rectangular region are selected. Figure 6 shows rectangular region for selecting parts. The user can also invert selection of parts.

- **Sorting and filtering:** A sorting function has been implemented to sort the parts based on size i.e. radius of the bounding sphere. For display purposes, all parts have the same size. However, to distinguish between the sizes, larger parts are colored in red while smaller parts are colored in white. The parts can be restored to their original size by selecting a menu option. The scene then contains parts having their original size. A filtering utility has also been implemented to filter out parts based on size and complexity. The size criteria helps in filtering the parts based on volume while the complexity criteria helps in filtering the parts based on number of faces.
- **Display windows:** An output window has been implemented in the system. For parts, the window displays all the files that the user has selected and their respective sizes. In addition to the output window, a scenegraph window has been implemented in the system. This window displays the contents (parts) of each scene as a tree-view similar to the tree-view in windows explorer. Each node of the tree is a part. If a part in the scene is selected then the corresponding nodes in the tree-view are highlighted.

4. GEOMETRY-BASED SEARCH CAPABILITIES

The geometry-based search tool locates existing parts similar to the new part based on some geometric attributes. It creates signatures for each of the parts in the database and stores the signatures along with the solid model of the part. A signature is a list of geometric attributes that describe the part and depends on the application. These pre-computed signatures reduce the time required for comparison and, thus, improve the speed of comparison. The search tool then uses the signatures to compare the signature of the query part with each of the signatures of the database parts to determine if the parts are similar.

The geometry-based search tool uses four signatures to identify existing parts similar to the query part. The signatures are applied sequentially to improve the efficiency and accuracy of the comparison. The following four signatures are used to compare the query part with database part to assess similarity.

- **Part Volume and Surface Area:** The part volume and surface area are used as a first signature to remove all those parts that are considerably smaller/ larger than the query part. If V_Q is the volume of the query part Q and V_P is the volume of an existing part P , then according to the current system settings part P is pruned out if $V_P < (V_Q/3)$ or, $V_P > (3 \times V_Q)$.

These system settings can be changed based on the user requirement. A similar criterion is used to prune out parts based on the surface area. The volume and surface area of the part are computed using standard ACIS functions.

- **Basic Shape Statistic:** The next signature uses face type information to prune out those parts that have different geometry of faces than the query part. We have identified five types of face geometries. These include: planar, cylindrical, conical, spherical, and NURBS. The second signature stores the number of faces of each type in a part. Thus, the signature is a vector having five elements, each corresponding to one of the five surface types mentioned above. The query part vector is compared with the vector of the database part and if the number of mismatches exceeds a particular value then the database part is pruned. This value can be set by the user depending upon his requirement. The number of faces of part of particular type is computed using standard ACIS functions.
- **Gross Shape Complexity:** The gross shape complexity is the third signature used to prune database parts. It uses convex hull of the part to determine the complexity of the gross shape. It is a vector having two elements. The first element is the ratio of the volume of the convex hull of the part to the volume of the part and the second element is the ratio of the surface area of the convex hull of the part to the surface area of the part. The vectors for the database and query parts are compared using L_2 norm. If the value of the norm exceeds a particular preset

value then the database part is pruned. This value can be set by the user. A standard incremental algorithm has been implemented to compute the convex hull of a part.

- **Detailed Shape Complexity:** The detailed shape complexity is expressed using normal direction area distribution. Normal direction area distribution is a graph for each surface type α obtained by plotting the total fractional area in each normal direction starting from the direction having the largest fractional area until the direction having the tenth largest area.

Normal direction area distribution vector = $[a_{\alpha,n_1} \ a_{\alpha,n_2} \ \dots \ a_{\alpha,n_{10}}]$. The comparison is carried out using L_2 norm.

The following approach is used to extract information for constructing detailed shape complexity signature.


1. Convert a solid model of an engineering part to a form acceptable for analysis by ACIS (a commercially available geometric kernel from Spatial Technology).
2. From the ACIS model of Step 1, the following information is retrieved:
 - a. The total number N_f of planar, cylindrical, conical and quadratic faces in the solid model.
 - b. The area of the k th face of the solid model, A_k , where $1 \leq k \leq N_f$
 - c. Indicator of the surface type α for each surface: planar ($\alpha = 1$), cylindrical ($\alpha = 2$), conical ($\alpha = 3$), spherical ($\alpha = 4$), and NURBS ($\alpha = 5$).
 - d. The normal direction or the direction of the axes of rotation of the k th face, $n_{\alpha k}$.
3. From the information obtained in Step 2, and after obtaining A_k and $n_{\alpha k}$ for each surface of the solid, the surfaces are sorted based on the normal directions or axes of rotation $n_{\alpha k}$ to obtain the following additional information:
 - a. The number N_α of unique normal directions or axes of rotation for each type of surface α .
 - b. The i th unique direction of a unit normal to a planar surface or the direction of the axis of rotation of a conical, cylindrical, or quadratic surface, $n_{\alpha i}$.
 - c. The number $M_{\alpha i}$ of planar surfaces in the normal direction $n_{\alpha i}$ and the number $M_{\alpha i}$, $\alpha = 2, 3, 4$, of cylindrical, conical, and quadratic surfaces having $n_{\alpha i}$ as the direction of the axis of rotation.

The information obtained in Steps 2 and 3 is used to compute the normal area distribution histogram.

The four signatures are sequentially compared and at each stage of the comparison those database parts that exceed the preset value are pruned. The final distance value is the sum of the L_2 norms of the four signatures.

5. IMPLEMENTATION AND EXAMPLES

The geometry-based search has been implemented based on the algorithms described above using Microsoft Foundation Classes (MFC), ACIS 7.0, and OpenGL on a Windows platform. To locate a set of parts that contains the query part, the designer must supply the part geometry of the query part (ACIS files .sat, .sab) and the directory of the database to search. The output of the system is the ACIS models of all the parts that are similar to the query part.



| Subject | Part 1 | Part 2 | Part 3 |
|---------|--------|--------|--------|
| 1 | 0:46 | 2:52 | 2:12 |
| 2 | 7:22 | 3:05 | 2:10 |
| 3 | 7:23 | 5:44 | 3:12 |
| 4 | 4:05 | 3:30 | 3:46 |

Fig. 7. Results obtained from the experiment performed using Visual Search system.

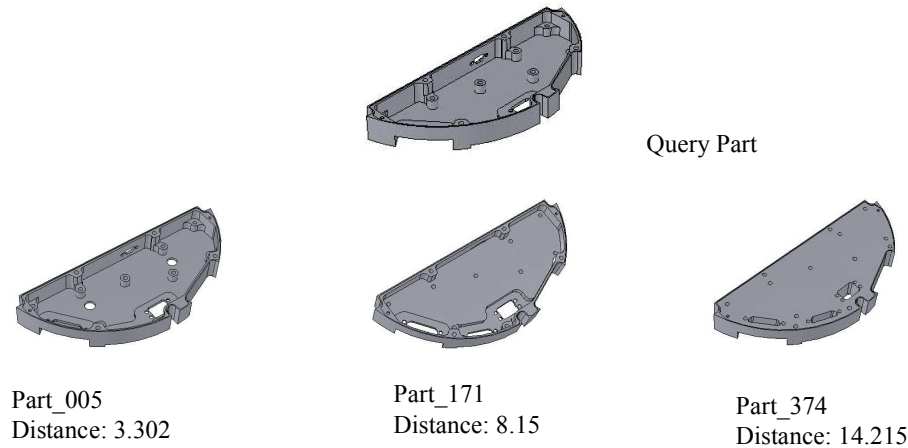


Fig. 8. Results obtained from Geometry-Based Search system.

Visual Search is written in C++ under Microsoft Windows using the MFC-library for drawing the windows. For displaying the parts the OpenGL-library is used. This is a portable, high-level graphic library on the top of OpenGL, which was developed for high performance applications like simulators, virtual reality, games and scientific visualizations.

An informal case-study was performed to evaluate visual search tool. Four subjects, none of whom ever used the visual search tool before, were invited to search for three different geometries in a directory containing 400 parts. The test procedure was conducted as follows.

1. Each subject was introduced to basic functionality of visual search tool in about five minutes.
2. The subject was shown a printout on a paper with a picture of the geometry he was supposed to search for. She/He was also shown the location of the top-most directory containing the part.
3. The aim was that the subject should be able find the part quickly, using all the tools of the system.
4. The time was started as soon as the candidate got the paper in his hands and it was stopped at the moment he found the right part.

The parts are shown at the top of the table in Figure 7. The first geometry did not have any similar one in the database, while the second and the third geometries had several ones similar to them. In the last test the size of the part was provided as additional information to help the search. As seen from the table, locating the last part always took less time than locating the first two. In Figure 7 the times taken are given in minutes and seconds. So this informal study shows that once the subject performs the first part search using Visual Search, the next part is found very fast. Providing additional information (part size) speeds up the searching progress significantly.

Figure 8 shows the part used as input to the geometry-based search tool and three parts that have been retrieved from the part database. The retrieved parts have gross-shape similar to the query part. The time taken for the search was 0.2s. Thus, the geometry-based search tool can retrieve parts for database quickly.

6. CONCLUSIONS

This paper shows the usefulness of integrated search strategies consisting of geometric and visual search tools for locating similar parts in a database of parts. The search tools described in this paper are easy-to-use and use efficient techniques for performing searches quickly. The integrated system is very effective in searching as it allows the user to switch between geometry and visual search at any time. The system has been tested with a wide variety of parts and provides the user with the option of using both visual as well as geometric search tools to improve search results. If the query part is not available the user can visually scan the database to identify a suitable part that can be used as it is or it can be used as query for geometry-based search tool.

We expect that the integrated search system will help the designer in searching for the parts effectively for reusing the design information. While designing a new part the designer can refer to existing parts and utilize the components used previously. Reusing design/manufacturing information stored would result in a faster and more efficient design process.

The current version of visual and geometry-based search tools works with ACIS file formats (.sat, .sab). This will be extended to work standard file formats such as STEP, IGES.

Acknowledgements

This research is supported in part by the Center for Energetic Concepts Development at the University of Maryland, Naval Surface Warfare Center at India Head, and Iktara and Associates.

7. REFERENCES

- [1] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In 8th IEEE International Conference on Computer Vision, Vancouver, Canada, July 2001.
- [2] A. Cardone, S.K. Gupta, and M.V. Karnik. A survey of shape similarity assessment algorithms for product design and manufacturing applications. *Journal of Computing and Information Science in Engineering*, 3(2):109-118, June 2003.
- [3] A. Cardone, S.K. Gupta, and M.V. Karnik. Identifying similar parts for assisting cost estimation of prismatic machined parts. In ASME Design for Manufacturing Conference, Salt Lake City, Utah, September 2004.
- [4] V.A. Cicirello and W.C. Regli. Machining feature-based comparisons of mechanical parts. In International Conference on Shape Modeling and Applications, Genova, Italy, May 2001.
- [5] A. Elinson, D.S. Nau, and W.C. Regli. Feature-based similarity assessment of solid models. In ACM Symposium on Solid Modeling and Applications, Atlanta, GA, 1997.
- [6] M. Hilaga, Y. Shinagawa, T. Kohmura, and T.L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In SIGGRAPH, 203-212, August 2001.
- [7] N. Iyer, Y. Kalyanaraman, K. Lou, S. Jayanti, and K. Ramani. A Reconfigurable, Intelligent 3d Engineering Share Search System Part I: Shape Representation. In 23rd ASME DETC Computers and Information in Engineering (CIE) Conference, Chicago, IL, 2003.
- [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, February 2005.
- [9] D. McWherter, M. Peabody, A. Shokoufandeh, and W.C. Regli. Database techniques for indexing and clustering of solid models. In 6th ACM Symposium on Solid Modeling and Applications, June 4-8, Ann Arbor, MI 2001
- [10] D. McWherter, M. Peabody, A. Shokoufandeh, and W.C. Regli. Transformation invariant similarity assessment of solid models. In ASME DETC, Pittsburgh, PA, September 2001.
- [11] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In International Conference on Shape Modeling and Applications, Genova, Italy, May 2001.
- [12] M.M. Ramesh, D.Y. Hoi, and D. Dutta. Feature-based shape similarity measurement for retrieval of mechanical parts. *Journal of Computing and Information Science in Engineering*, 1(3):245-256, September 2001.
- [13] G. Srinivas, D.E. Fasse, and M.M. Marefat. Retrieval of similarly shaped parts from a CAD database. In IEEE International Conference on Systems, Man, and Cybernetics, San Diego, CA, October, 1998.
- [14] T. L. Sun, C. J. Su, R. J. Mayer, and R. A. Wysk. Shape similarity assessment of mechanical parts based on solid models. In ASME Design for Manufacturing Conference, Boston, MA, September, 1995.
- [15] R. Sung, H. J. Rea, J. R. Corney, D. E. R. Clark, J. Pritchard, M.L. Breaks, and R.A. MacLeod. Assessing the effectiveness of filters for shape matching. In 2002 ASME International Mechanical Engineering Congress & Exposition, IMECE '02. New Orleans, LA, November 2002.
- [16] V. Tuzikov, J.B.T.M. Roerdink, and H.J.A.M. Heijmans. Similarity measures for convex polyhedra based on Minkowski addition. *Pattern Recognition*, 33(6):979-995, June 2000.