An Integrated 2D and 3D Shape-based Search Framework and Applications

Pu Jiantao and Karthik Ramani

Purdue University, ramani@purdue.edu

ABSTRACT

Traditional computer-aided design tools have lead to proliferation of products. In itself CAD systems have become complex to use with many features. In the light of the use of menu driven systems, this article proposes a freehand sketch driven search and reuse of designs, beautification of sketches, and a new interaction paradigm with navigation. We review our prior work in this area in an integrated fashion. The central part of this paper addresses the representation of both 2D sketches and 3D designs in a manner that is tailored for engineering. The interaction of the user with the sketch based search forms a natural interface for interaction with both 2D and 3D models. In addition, the interface is also integrated with a constraint solver for freehand sketches. The system can also suggest to the user the part-class name allowing the probing of the classes and semantics in the database. Besides, users can discover new knowledge by navigating our 2D/3D visualization interface freely based on the preliminary retrieval of classification results. Similarity search can be performed any time during the navigation process.

Keywords: Shape Analysis; Supply Chain; Sketch Based Search.

1. INTRODUCTION

It is well recognized that engineering design starts with a sketch. Sketch-based part retrieval is a more natural form for searching during early conceptual design than say example-based part retrieval. When a 3D query example is not available, a sketch is especially useful. Therefore, it is necessary to have a fast and effective system for sketch-based engineering part retrieval and interaction. Sketches provide the possibility to record abstract information and are very close to a designer's model of a product. During the design process, the sketch compensates for short-term memory limitations. At the same time the sketch supplements the engineer's cognition by depicting the visual perception in a concrete form [1]. In addition, a sketch can set up a visual dialog between the designer and other group members [2]. Although the freehand sketch allows the designer to explore new ideas with a minimal effort, the interpretation of sketches is still very difficult for computers because sketches are vague and incomplete [3]. Sketches can provide cues for knowledge searching due to the connection between the perception-based actions and design knowledge. It is clear that the knowledge reuse is important in the process of computer-aided design. However, implementing such a system is still difficult. Earlier work has not closely examined how to use sketches for knowledge retrieval and reuse. Methods for transforming the vague knowledge in a sketch to a precise representation, which can further be used for search in 3D models, 2D drawings and other sketches is reviewed. We exploit the advantages of a sketch based user interface to (1) extract the implicit constraints in a user sketch for beautification, (2) search for both 2D and 3D models, (3) obtain suggestions of part classes from the classification in the system, and (3) interact with 2D and 3D models for query modification and/or reuse. Finally the paper concludes by outlining some practical applications of this research.

2. APPROACH

During the design process, a user can sketch her ideas using a pen without turning to the traditional interaction means such as using keyboard or a mouse. The sketched result can be used in three ways: be beautified for precise input, retrieve similar design schemes available in database, or have the system suggest a class of parts to browse further. With the help of the constraint solver, a formal design can also be created. Users are allowed to switch between these modules flexibly according to their needs.

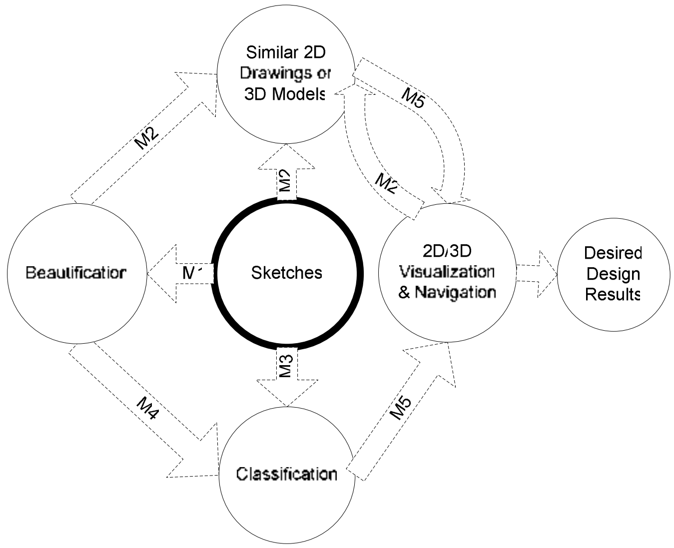


Fig. 1: Interactions between the key modules introduced in this paper: (1) M1: sketch beautification module; (2) M2: 2D and 3D shape retrieval module; (3) M3: geometric constraint satisfaction module; (4) M4: sketch to 3D classification module (5) M5: multidimensional scaling module

2.1 Sketch Acquisition

The sketch acquisition module records users' search intent using sketch. Users can employ a pen or a mouse to sketch. In our system design, the sketches are drawn through a sequence of strokes. Fig. 2 shows the visual appearance and the architecture of the sketch editor. During the sketching process, the system monitors the action of the mouse or the pen. Once the mouse cursor is moving and the left button is pressed down, it can be concluded that users have begun to draw sketches. Now the moving path of the cursor is recorded in real time with the end of the stroke indicated by the release of left button.

Each track of a stroke S is composed of a sequence of small line segments rather than image bitmaps: $S = \{((x_i, y_i), (x_{i+1}, y_{i+1}), t_i) \mid 0 \le i \le n\}$ where n is the total number of line segments included in a single stroke S, (x_i, y_i) and (x_{i+1}, y_{i+1}) are the two ending points of a small line segment at time t_i. Consequently, sketching activity A is usually formed by a sequence of stroke $A = \{S_i \mid 0 \le i < m\}$ where m is the number of strokes. In the end, the desired shape descriptor will be extracted from these strokes [4].

In the sketch process, it is inevitable that the user will make some mistakes. Therefore, besides the sketch operations, some editing operations are also provided to users. Some basic operations, such as erase, trim, move, rotate, zoom, and view copy are included. More operations can be added into this system, although only a few basic operations are provided in this system.

2.2 Sketch Beautification and Implicit Constraints

Sketch beautification is then integrated with the current system to regulate the freehand sketch. Sketch beautification boosts the users sketch representation and disambiguates many implicit constraints. To take advantage of freehand sketch-based interaction, many methods have been proposed to parse and recognize the sketches. For parsing purpose, the work outlined in Refs [5] and [6] tried to explore the interactive nature of sketching such as the stroke direction and speed. However, the sketching activity has to be recorded in real time. The perturbation of a user's hand will lead to obvious changes in curvature and speed, and thus result in incorrect segmentation. To overcome such limitations, many other approaches were proposed such as the template-based method [7] and the Bayesian-based statistical model [8]. Unfortunately, these methods are still not capable of handling the parsing problem robustly. Then some other approaches [9-10] imposed constraints on users' behaviors. For example, a user was required to explicitly indicate the intended portioning of the ink, i.e., each pen stroke must represent a single shape, such as a single line segment or arc segment. Despite their simplicity, the single-stroke requirement usually resulted in a less natural interaction. Following sketch parsing, the next important step is the recognition of the parsed result. Approaches interpreting individual geometric shapes include HMM-based (Hidden Markov Model) algorithm [11], Zernike moment features [12] and multi-layer image recognition scheme [13]. However, most recognizers are either hand-coded or require large sets of training data to reliably learn new symbols. They still could not assure an ideal accuracy. In contrast, we proposed a new algorithm [14] to parse and recognize sketches. The parsing procedure is independent of the sketching speed and curvature, stroke-order, -number, and -direction, as well as invariant to rotation, scaling, and translation of strokes. It can also be used to recognize composite shapes.

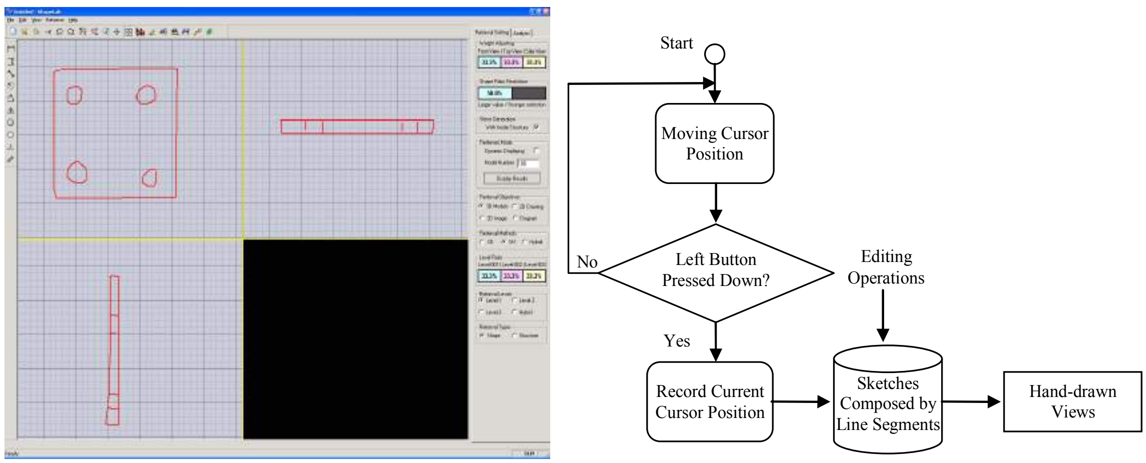


Fig. 2: Sketch acquisition module: (left) sketch editor, (right) information workflow.

To parse a stroke into independent primitives, we propose a circle-scanning strategy [14], in which multiple circles are used to scan the sketches by changing the radius of each circle progressively. When two neighbor intersected points between a scanning circle and the sketch are close enough, the shared point between the two respective intersected line segments is regarded as the critical point or segment point. As for the detailed explanation, please refer to our paper [14]. Fig. 3 shows the scanning circle approach for parsing 2D sketches. The method uses circles of increasing radius to scan a sketch progressively and the intersecting points are analyzed. In practice we use scanning circles in multiple directions and the Figure below is only for illustrative purposes.

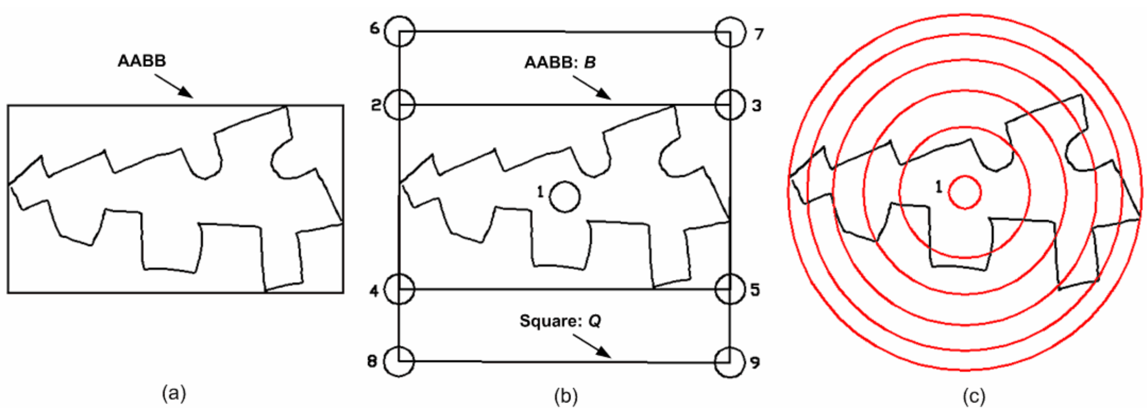


Fig. 3: Scanning Circle Approach.

To recognize a sketched shape, we use a template-based shape similarity method. Given a parsed segment, we compute the similarities between this segment and all predefined geometric primitive such as line, circle and arc. The most similar primitive can tell what kind of primitive the parsed segment represents. As for the introduction of the shape similarity computation, refer to Ref. [14]. In practice, this approach can also be used to recognize a sketched composite.

Once the primitives in the sketches are recognized, they can be beautified. However, during the sketching process, it is difficult for users to sketch precisely, and over-sketching or under-sketching is unavoidable. Simply converting the segments into the corresponding regular primitives is not enough. To handle these problems, we assume that users usually start and end their sketches at a certain geometric entity. In this way, we can use the nearest-neighbor principle to handle these drawbacks: find the nearest entity on which the starting point or the ending point of a sketch are located and the intersecting point between the two entities is the right starting or ending point. Another problem is that an arc can be sketched differently. Frequently the proposed segment method will parse a sketched arc into several parts. To handle this issue, we propose a rule: if two neighboring parsed segments are recognized as arcs and their angles have the same direction, then they will be merged as one arc.

Finally, during the beautification, we have to determine the parameters for each geometric entity. For a line, we need to set the starting point and ending point. We can adopt the starting and ending points of the sketch segment as the initial starting and ending points of a line. Later the two points can be adjusted after over- and under-stroke process. For an arc or circle, we select three points on the sketch segment and use them to compute the radius and the center of the arc or circle since three points determine an arc or circle. The three points can be selected in a simple way: the two ending points and the mid point of the parsed primitive.

2.3 2D and 3D Shape Representation

Sketch beautification is then integrated with the current system to regulate the freehand sketch. In our previous work15 we have proposed a method to compute the pose of a 3D model by finding the orthogonal orientations with maximum virtual contact area (VCA). The basic idea stems from the fact that the orthogonal directions that have the maximum VCAs are the major principal axes of the 3D model and provide a stable pose. The key step in obtaining the principal axes is to determine the polygons of a 3D object that have the same normal and lie in the same plane. VCA is defined as the bounding area formed by polygons that have the same distance from a predefined point and have the same normal. To obtain the direction along which the VCA is the maximum, we need to find all polygons that have the same normal direction and the same distance to the mass center. The direction that gives the maximum VCA is the first principal axis $\mathbf{b}^u$ of the 3D object orientation. To get the next principal axis $\mathbf{b}^v$ of an object orientation, we find the normal that satisfies two conditions: (a) is orthogonal to the first principal axis; and (b) has maximum area. The third axis can then be obtained by performing the cross product between $\mathbf{b}^u$ and $\mathbf{b}^v$:

$$\mathbf{b}^w = \mathbf{b}^u \times \mathbf{b}^v \tag{1}$$

To depict a 3D model precisely using 2D views, we abstract the 3D shape as multiple levels of detail as illustrated in Fig. 4. It can be seen that the contour level reflects its global shape by which a user can "guess" the true object to some extent. The silhouette level conveys more shape details using a few more simple sketches compared to the contour level. When the detailed shape information is not important, the silhouettes are enough to differentiate two similar objects more confidently as compared with the contours. The third level contains the complete information, including the visual appearance and the occluded structure, by which a user can figure out its shape precisely. In practice, especially in engineering fields, there are a lot of models that look similar from visual appearance but still have different inner structures. Therefore, we need to consider the complete details of the object. However, the projected shapes at the three levels are different from different viewpoints. With the help of the pose determination method, we project a 3D shape onto the six faces along the principal axes to represent the Multi-level Detail (MLD). At the contour level, there are three different views along the principal axes; at the silhouette level, there are six different views; and at the full level, we use the traditional drawing-like views to represent the drawing level, and there are three different views along the principal axes.

2.4 Shape Descriptors

In Ref. [4], we have proposed two methods to retrieve 2D drawings by measuring their shape similarity. The first approach represents a drawing as a spherical function by transforming it from 2D space into 3D space and then employs a fast spherical harmonics transformation to get a rotation invariant descriptor. The second method represents the shape of a 2D drawing from the statistics perspective as a distance distribution between pairs of randomly sampled points. Both the representations have many valuable advantages: invariant to affine transformation, insensitive to noise or cracks, simple, and fast.

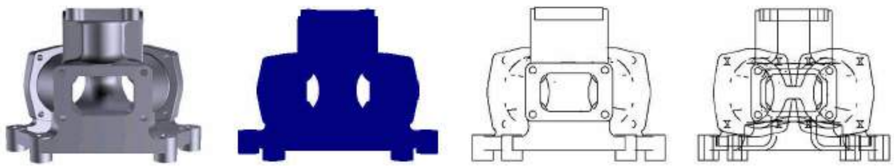


Fig. 4: Multiple levels of detail: (a) a 3D model: (b) contour level; (c) silhouette level; (d) drawing level.

A 2D analog of the spherical harmonics was used in Ref. [16] to extract a series of rotation invariant signatures by dividing a 2D silhouette shape into multiple circular regions. However, this method has two major limitations as mentioned in Ref. [4]: one-to-multiple correspondence and instability caused by shape perturbation. In order to overcome these limitations and thus obtain a set of robust rotation invariant signatures for a 2D shape, we propose a strategy called 2.5D spherical harmonics representation [4], which can extract a series of rotation invariants by transforming a 2D shape from 2D space into 3D space uniquely. The name “2.5D” arises from the fact that a 2D shape is represented in a 3D space. The key idea of using a 2.5 D spherical harmonics is to create a more unique one-to-one mapping between the shape in 2D and its descriptor overcoming problems in earlier methods (see Ref. [5]).

The basic steps can be described as follows. First, given a 2D shape, a sphere is determined and its equator plane encloses the 2D shape. Second, a set of rays located in the equator plane of the sphere is uniformly shoot out from the sphere center. Thus the interested points between the rays and the 2D shape are regarded as an approximation of the 2D shape. Third, to represent the intersected points using a one-to-one mapping spherical function, these points are projected onto a cylinder whose basis plane is the same as the equator plane of the sphere. Each point has a different height that is equal the distance between the sphere center and this point. Consequently, a 2D shape is uniquely transformed into a 3D spherical representation. We name this process a 2.5D transformation. Fig. 5 shows an example of this transformation for a 2D shape example. From this example, we notice that the geometric information is represented clearly in 3D space along the surface of a cylinder. Finally, to obtain the rotation-invariant, we use the fast spherical harmonics transformation method [17] in which a spherical function of bandwidth B is sampled on the 2B-many Chebyshev points. These sampled points form a 2B×2B equiangular grid along the longitude and latitude of a sphere. In our implementation, B is equal to 64. It means that a given 2D shape is depicted by 64 float numbers.

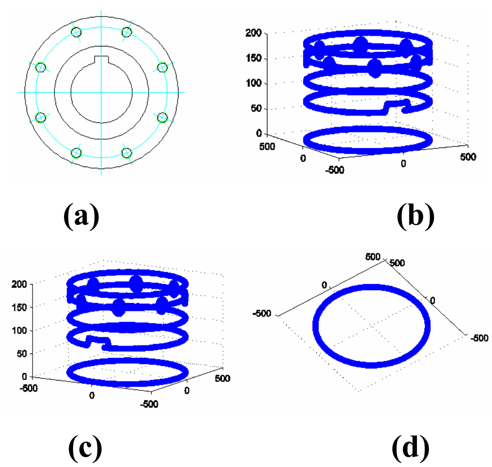
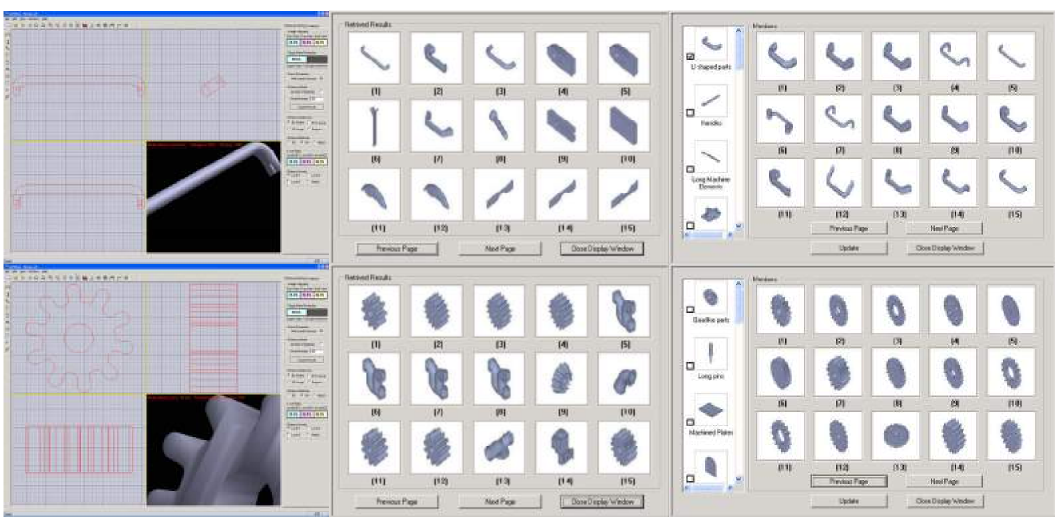


Fig. 5: An example of 2.5D spherical harmonics representation: (a) is a 2D shape; and (b), (c), and (d) show the 3D representation of the drawing from different perspectives.

2.5 View-based 3D Shape Classification

In our system, three 2D orthogonal views by pose determination and projection are automatically generated from each 3D triangulated model [15].Therefore, given a shape description from views of a query, the system can find similar 3D models. Compared to most other existing 3D shape descriptors, which capture the form from 3D models directly, shape signatures generated from views perform well and can be applied to view-based 3D model retrieval directly [18]. Similarly, it is intuitive to accept the idea of sketch-based 3D model classification given the fact that sketches are the most natural form for shape expression. Sketch-based 2D symbol classification/recognition has progressed extensively in the past decades. Most classifiers/recognizers either use a coded template for matching [19-23] or require sets of training data to reliably learn new symbols.[12], [13], [24-27] Among them, methods using statistical learning for symbol classification share a similar background with this paper even though we mainly focus on sketch-based 3D part classification. A classifier combination is applied to sketch symbol recognition using user-defined training examples [18]. This method can reach higher classification accuracy because the sketch query is consistent with the training data. However, the idea is not applicable for 3D engineering parts because the engineering part classification scheme and training data are hard to define on the fly. Besides, engineering parts are difficult to sketch formally for training purposes. Therefore, we motivate the user to help the system obtain the best retrieval with the classification engine defined by real engineering models.

For our work, we rely on shape descriptors as feature vectors for the classification problem. Our experience with sketching has shown that users prefer to draw a model at a higher level, thus closer to the contour level of the view generated from the 3D model, which captures an outer boundary and internal boundaries from a specific view of a 3D model [28]. In this paper, two criteria are needed to meet the shape descriptor selection. First, it has to be applicable to both 2D views and the sketch. Second, it is rotation invariant so that optimal alignment identification can be saved. Three shape descriptors are chosen to represent the shape content from the sketches/views in this context: 2.5D Spherical Harmonics (SH) from the contours [28], Fourier Transform (FT) from the outer boundary [29], and the Zernike moments (ZM) from the region inside the outer boundary.30 These three shape descriptors have been shown empirically to perform well in the task of shape matching. Although our framework is independent of the shape descriptor selected, we choose 2.5D SH, FT and ZM because they complement the shape description from different perspectives using dissimilar techniques. For example, 2.5D SH includes the internal boundaries in addition to the outer boundary considered by FT, while ZM reflects more of the internal details by describing the distribution inside the region. Therefore, it is expected that classifier combination can achieve a better performance. For the current work, we concatenate shape signatures generated from three views to form a single feature vector $x \in \Re^n$. Data produced at this stage is also employed for classifier recognition.



Query Shape Search Only Unified Search

Fig. 6: GUI of part class browsing and retrieval with examples.

Fig. 6 shows the Graphical User Interface (GUI) of the implementation. On the left-hand side of the GUI are the query input. The shape search only results are shown in the middle. Models of the selected class are shown in the order of shape similarity to the query on the right-hand side. The default images of the models will be the ones belonging to the class that has the highest possibility. The user is then able to browse groups of models based on the selection that he/she thinks as the right classes. A comparison of the shape only search and unified class based search shows the clear advantages of class suggestion. The proposed framework can not only improve the search effectiveness and efficiency, but also enhance user interaction by involving only a limited number of highly possible choices. This design will be especially useful for a large database with a large number of classes

2.6 2D and 3D Shape Navigation

Formally, each 2D view, *V*, is described by an aspect ratio and histogram vector. For 2.5D Spherical Harmonics representation where *N*=64. We can now describe two 3D objects *U* and *V* by the set of the three orthogonal views, i.e., *U* (= {*U*1, *U*2, *U*3}) and *V* (= {*V*1, *V*2, *V*3}).

$$D(U,V) = w_a \sum_{m=1}^{3} dist(u_m^{'}, v_m^{'})$$

where is the *m*th principal view (e.g., *m* = 1 means first principal view) as described below and is a weight based on normalized aspect ratios. We refer to *D*(*U*,*V*) as the global similarity as it accounts for similarity of all the three corresponding view pairs. The non-metric nature of the distance function used in the shape matching algorithm poses a special problem in reducing the high dimensional feature vectors into lower dimensional vectors. We overcome this problem with non-linear multidimensional scaling (MDS) [11], [32], to convert the database objects from similarity space to Euclidean space. The input to the non-linear MDS is the dissimilarity matrix obtained from the distance metric proposed above. From the MDS analysis we found that the optimum dimensions for this dataset was 3, confirming the fact that a 3D visualization is sufficient to represent the whole database, without disturbing the original distances significantly.

A user is allowed to navigate through 2D or 3D space instead of browsing the traditional scroll list like results based on a query (1D space) for similar models. With the help of the proposed interaction paradigm (a) 3D CAD models which look similar from a particular view, or (b) have a very high overall similarity on the whole can be easily found in a natural way. Towards this goal: (1) orthogonal main views of a 3D shape commonly used in design are obtained; (2) effective visualization of large amount of 3D models by navigating in the 2D & 3D similarity space is enabled; and (3) natural searching/browsing mechanisms are used. Similarity search can be performed any time during the navigation process. Users are allowed to switch between the sketch-beautification, search in 2D and 3D, class suggestion and navigation modules flexibly according to their needs. At the end, users can pinpoint their desired design results assisted by our interactive navigation system [33].

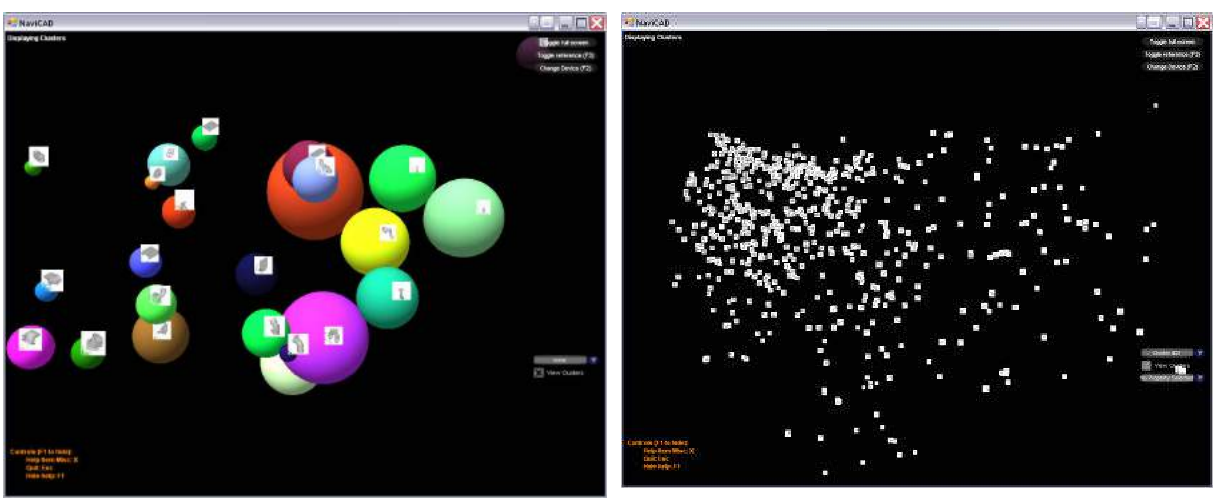


Fig. 7: Interface for product search using 3D navigation.

2.7 Applications Development

Today most of the organizations in the supply chain have become very specialized in engineering functions such as product design, industrial design, tooling, and manufacturing processes. This increased specialization and the growing economic pressures on OEM companies have resulted in an increased trend towards outsourcing manufacturing and some other support functions. A key search application related to this research helped launch the worlds first shape based search engine for the global supply chain by our commercial development partner (see Fig. 8) [34].

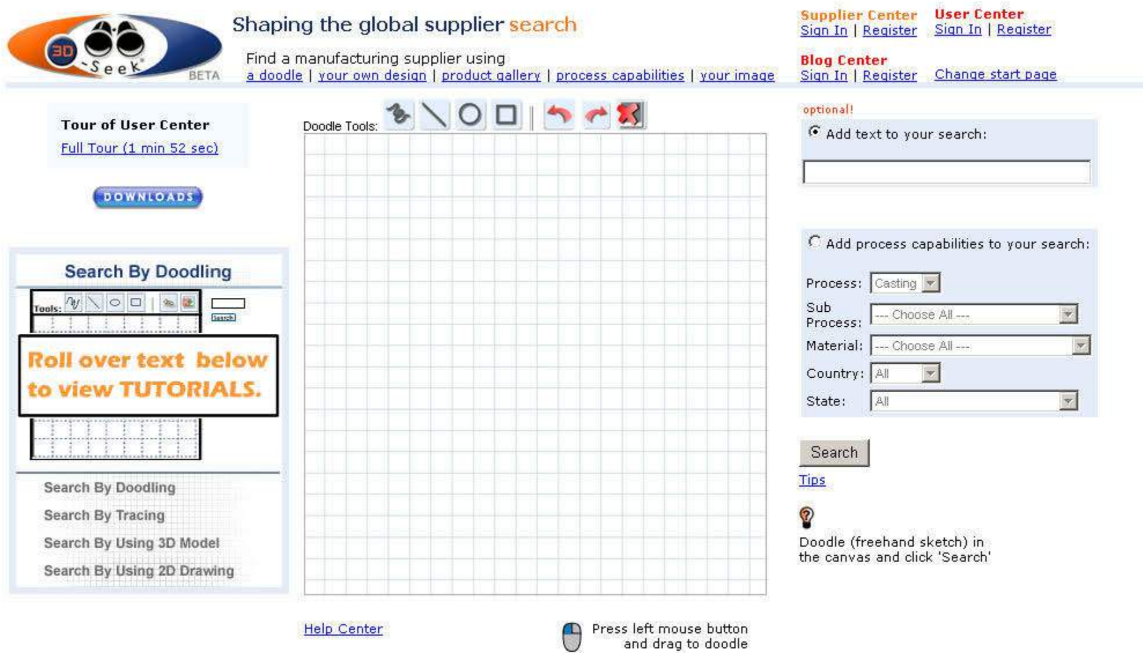


Fig. 8: Interface for product search connecting to supply chain (Courtesy Imaginestics).

3. CONCLUSIONS

In this paper, we introduced a new design paradigm which is supported by four key modules: a sketch beautification module, a 2D/3D search engine, a sketch-based user interface, sketch-based classification and 2D/3D navigation modules. The motivation is to overcome the limitations of traditional computer-aided design that are not natural and require heavy training, and furthermore is inefficient. In addition the users can search, retrieve and learn through browsing from large repositories in natural ways. Current CAD systems have many problems and natural searching is becoming more serious as versions change and become complex.

The proposed design paradigm has two distinct characteristics: (1) freehand sketch is used in the whole design cycle; and (2) the design knowledge implied in 2D legacy drawings and 3D models is reused conveniently by performing search operations. Usability evaluation demonstrated that this design paradigm is more natural and efficient than the traditional (Window, Icon, Menu, and Pointer) WIMP paradigm. Finally this paradigm has implications in design advisory systems as well as supply chain connection of buyers and sellers.

4. REFERENCES

[1] Beryl, P.; Mark, A.: Computer-aided Sketching to Capture Preliminary Design, In Proceedings of ACM International Conference Proceeding Series on Third Australasian Conference on User Interfaces, Melbourne (Australia), 2002,7, 2002, 9-12.

[2] Tversky, B.: What Does Drawing Reveal about Thinking, In Proceedings of Visual and Spatial Reasoning in Design, Cambridge (USA), 1999.

[3] Van Dijk, C. G. C.: New Insights in Computer-aided Conceptual Design, Design Studies, 16(1), 1995, 62-80.

[4] Pu, J.; Ramani, K.: On Visual Similarity Based 2d Drawing Retrieval, Journal of Computer Aided Design, 38(3), 2006, 249-259.

[5] Sezgin, T. M.; Stahovich, T.; Davis, R.: Sketch Based Interfaces: Early Processing for Sketch Understanding, In Proceedings of the 2001 Workshop on Perceptive User Interfaces, Orlando (USA), 2001, 1-8.

[6] Stahovich, T. F.: Segmentation of Pen Strokes using Pen Speed, In Proceedings of 2004 AAAI Fall Symposium on Making Pen-based Interaction Intelligent and Natural, Washington D.C. (USA), 2004, 152-158.

[7] Hse, H.; Shilman, M.; Newton, A. R.: Robust Sketched Symbol Fragmentation Using Templates, In Proceedings of the 9th International Conference on Intelligent User Interface, Funchal (Portugal), 2004, 156-160.

[8] Shilman, M., Pasula, H., Russell, S., and Newton, R.: Statistical Visual Language Models for Ink Parsing, In Proceedings of AAAI Spring Symposium on Sketch Understanding, Palo Alto (USA), 2002, 126–132.

[9] Igarashi, A.; Matsuoka, S.; Kawachiya, S.; Tanaka, H.: Interactive Beautification: a Technique for Rapid Geometric Design, In Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology, Banff (Canada), 1997, 105-114.

[10] Forbus, K.; Lockwood, K.; Klenk, M.; Tomai, E.; Usher, J.: Open-domain Sketch Understanding: The nuSketch Approach, In Proceedings of the AAAI Fall Symposium on Making Pen-based Interaction Intelligent and Natural, Washington, D.C. (USA), 2004, 58-63.

[11] Sezgin, T. M.; Davis, R.: HMM-Based Efficient Sketch Recognition, In the Proceedings of the 10th International Conference on Intelligent User Interfaces, San Diego (USA), 2005, 281-283.

[12] Hse, H.; Newton, A. R.: Sketched Symbol Recognition Using Zernike Moments, In the Proceedings of 17th International Conference on Pattern Recognition (ICPR'04), Cambridge (UK), 2004, 367-370.

[13] Kara, L. B.; Stahovich, T. F.: An Image-Based Trainable Symbol Recognizer for Sketch-Based Interfaces, In Proceedings of AAAI Fall Symposium Series 2004: Making Pen-Based Interaction Intelligent and Natural, Arlington (USA), 2004, 99-105.

[14] Pu, J.; Hou, S.; Ramani, K.: Toward Freehand Sketch Beautification Driven By Geometric Constraint, Submitted to ACM Transactions on Graphics, 2006.

[15] Pu, J.; Ramani, K.: A 3D Model Retrieval Method Using 2d Freehand Sketches, Lecture Notes in Computer Science, Springer Berlin, Vol. 3515, 2005, 343-347.

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

[17] Healy, D.; Kostelec, P.; Moore, S.: FFTs for the 2-Sphere—Improvements and Variations, Journal of Fourier Analysis and Applications, 9(4), 2003, 341-385.

[18] Chen, D-Y.; Ouhyoung, M.; Tian, X-P.; Shen, Y-T.: On Visual Similarity based 3D Model Retrieval, Computer Graphics Forum, 2003, 223–232.

[19] Costagliola, G.; Deufemia, V.; Polese, G.; Risi, M.: A Parsing Technique for Sketch Recognition Systems, In Proceedings of 2004 IEEE Symposium on Visual Languages-Human Centric Computing (VLHCC'04), 2004, 19-26.

[20] Fonseca, M. J.; Pimentel, J.; Jorge, J. A.: Cali-an Online Scribble Recognizer for Calligraphic Interfaces, In Proceedings of the 2002 AAAI Spring Symposium on Sketch Understanding, 2002, 51–58.

[21] Fonseca, M.J.; Jorge, J. A.: Using Fuzzy Logic to Recognize Geometric Shapes Interactively, In Proceedings of the 9th IEEE Conf. on Fuzzy Systems, 1, 2006, 291-296.

[22] Valois, J. P.; Côté, M.; Cheriet, M.: Online Recognition of Sketched Electrical Diagrams, In Proceedings of Sixth International Conference on Document Analysis and Recognition, 2001, 460-464.

[23] Alvarado, C.; David, R.: SketchREAD: a Multi-domain Sketch Recognition Engine, In Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology, 2004, 23-32.

[24] Liu, W. Y.; Qian, W. J.; Xiao, R.: Smart Sketchpad - An On-Line Graphics Recognition System, In Proceedings of Sixth International Conference on Document Analysis and Recognition, 2001, 1050-1054.

[25] Sezgin, T. M.; Davis, R.: HMM-based Efficient Sketch Recognition, In Proceedings of the 10th International Conference on Intelligent User Interfaces, 2005, 281-283.

[26] Kara, L. B.; Stahovich, T. F.: An Image-Based, Trainable Symbol Recognizer for Hand-Drawn Sketches, University of Massachusetts, Boston, MA, 1997.

[27] Rubine, B.: Specifying Gestures by Example, ACM Transaction on Computer Graphics, 25(4), 1991, 329-337.

[28] Pu, J.; Hou, S.: Jayanti, S.; Ramani, K.: 3D CAD model retrieval based on multiple levels of detail, pp. 103-112, The 14th Pacific Conference on Computer Graphics and Applications, October 2006, Taipei, Taiwan.

[29] Zhang, D. S.; Lu. G. J.: Shape Retrieval Using Fourier Descriptors, IEEE International Conference on Multimedia and Expo (ICME), 2001, 1139- 1142.

[30] Khotanzad, A.; Hong, Y.: Invariant Image Recognition by Zernike Moments, IEEE Trans. Pattern Analysis and Machine Intelligence, 12(5), 1990, 489-497.

[31] Torgerson, W. S.: Multidimensional scaling: I. theory and method, Psychometrika, 1952, 401-419.

[32] Kruskal, J. B.; Wish. M.: Multidimensional Scaling, Sage Publications, Beverly Hills, CA, 1978.

[33] Pu, J.; Kalyanaraman,Y.; Jayanti, S.; Ramani, K.; Pizlo, Z.: Navigation and Discovery of 3D models in a CAD Repository, Accepted for publication in IEEE Computer Graphics and Applications (Special issue on 3D Documents).

[34] http://www.imaginestics.com