EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2004)
John F. Hughes and Joaquim A. Jorge (Editors)

Towards 3D Modeling using Sketches and Retrieval

Manuel J. Fonseca   Alfredo Ferreira   Joaquim A. Jorge

Department of Information Systems and Computer Engineering
INESC-ID/IST/Technical University of Lisbon
R. Alves Redol, 9, 1000-029 Lisboa, Portugal

Abstract

Retrieving 2D and 3D drawings by content is not an easy task. Automatic feature extraction, indexing and matching are some of the problems raised by these approaches. We have developed a generic method to classify, index and retrieve drawings using sketches, based on spatial relationships, shape geometry and high-dimensional indexing mechanisms. This approach has been applied with success to retrieving clip-art and complex technical drawings from large databases. In this paper we give a brief overview of our approach for content-based retrieval and describe two prototypes for retrieving 2D drawings. We also present a preliminary study that combines retrieval of 3D objects and expectation lists to define a new interaction paradigm based on suggestions. 3D objects are described using their face and edge graphs, which are then mapped into multidimensional descriptors through graph spectra. Preliminary results show that the combination of these two descriptors (faces and edges) provide a good novel method to describe and retrieve similar 3D objects. Finally, rather than developing a system to specify and display 3D queries and results, we integrated the retrieval system into a 3D modeling tool, through the use of expectation lists. This way, results from the query are presented as suggestions to the user, in what constitutes a new interaction paradigm, which is more flexible then present approaches.

Categories and Subject Descriptors (according to ACM CCS): H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval H.5.2 [Information Interfaces and Presentation]: User Interfaces

1. Introduction

Present–day CAD applications provide powerful tools to create and edit vector drawings in several domains, such as architecture, mechanics, automobile or mould industries. Even though reusing past drawings is common practice in such domains, there are almost no developed mechanisms to support this activity in an automated manner. Thus, it becomes important to develop new systems to support automatic classification and retrieval of technical drawings based on their contents, rather than relying solely on textual annotations or metadata for such purposes.

Some studies [Do98] refer that project libraries with old case studies are crucial to help designers identify relevant features to include or problems to avoid in new designs. Additionally, in some design firms, designers often work by making or copying diagrams from colleagues in their design team for further development [Do95]. Furthermore, during task analysis performed in the context of ongoing research projects [Con00] in informal conversations with draftspeople, we found out that industrial designers often include elements from libraries of ready-to-use components. Moreover, they also re-use old drawings during the creation phase of a new project, to get at ideas or review insights from previous problems and their solutions.

Even though reusing drawings often saves time, manually searching for them is usually slow and problematic, requiring designers to browse through large and deep file directories or navigate a complex maze of menus and dialogs for component libraries. Moreover, CAD systems while making the creation of new drawings easier, exacerbate the retrieval problems, because they do not provide adequate search mechanisms. Indeed, present-day CAD systems rely on conventional database queries and direct-manipulation to retrieve information. Some solutions to this problem use textual databases to organize the information [Bak90, CW91]. These classify drawings by keywords and additional information, such as designer name, style, date of creation/modification and a textual description. However, solutions based on textual queries are not satisfactory, because they force the designers to know in detail the meta-information used to characterize drawings. Worse, these approaches also require humans to produce such information when cataloging data.

In contrast to the textual organization, our approach uses a visual classification scheme based on shape geometry and spatial relationships, which are better suited to this problem, because they take advantage of designers visual memory and explore their ability to sketch as a query mechanism. These are combined with an indexing method that efficiently supports large sets of data, new schemes that allow us to hierarchically describe figures by level of detail and graph-based techniques to compute descriptors for such drawings in a form suitable for machine processing.

The rest of this paper is organized as follows: Section 2 provides an overview of related work in content-based retrieval. In Section 3 we describe our approach for content-based retrieval of visual information. Sections 4 and 5 present the two prototypes developed for retrieving 2D drawings and the correspondent experimental evaluation. In Section 6 we present our approach towards a new interaction paradigm based on suggestions that combines content-based retrieval techniques and expectation lists. We describe the extension of our retrieval approach to the 3D domain, present the modeling tool and the mechanism used to integrate both systems (modeling and retrieval). We conclude the paper by discussing our conclusions and presenting directions for further research.

2. Related Work

The work that we present here, in its preliminary stage, is innovative and new, in the sense that it combines multimedia information retrieval with modeling techniques to define a new interaction paradigm based on suggestions. In this section we analyze some representative examples of systems developed to classify and retrieve 2D drawings and 3D objects.

Gross's Electronic Cocktail Napkin [GD96] addressed a visual retrieval scheme based on diagrams, to indexing databases of architectural drawings. Users draw sketches of buildings, which are compared with annotations (diagrams), stored in a database and manually produced by users. Even though this system works well for small sets of drawings, the lack of automatic indexing and classification makes it difficult to scale the approach to large collections of drawings.

The S3 system [BK97] supports managing and retrieving industrial CAD parts, described by their geometry (2D contour) and thematic attributes. S3 retrieves parts using bi-dimensional contours drawn using a graphical editor. Contours can also be taken from sample parts stored in a database. S3 relies exclusively on matching contours, ignoring spatial relations and shape information, making this method unsuitable for retrieving complex multi-shape drawings.

Park and Um described an approach to retrieve complex 2D drawings of mechanical parts based on the dominant shape [PU99]. Objects are described by recursively decomposing its shape into a dominant shape, auxiliary components and their spatial relationships (Inclusion and adjacency). Their approach breaks a complex drawing into many adjacent closed loops or blocks, each with an arbitrary polygonal shape. Spatial relationships among blocks are extracted to create a graph. Then, each block is decomposed and recursively divided into many fragments of primitive shapes, yielding another type of graph that describes the geometry of blocks. The combination of these two types of graphs produces a complex graph structure, where each node of the graph describing spatial arrangement, has another graph describing the geometry of the correspondent block. The small set of base geometric primitives and the not-so-efficient matching algorithm, based on the breadth-first tree matching, make it hard to handle large databases of drawings.

Leung and Chen proposed a sketch retrieval method to store and retrieve unstructured free-form hand-drawings stored in the form of multiple strokes [LC02]. They use shape information from each stroke exploiting the geometric relationship between multiple strokes for matching. Their approach then computes a matching score between the query and each sketch in the database. More recently, authors improved their system by also considering spatial relationships between strokes [LC03]. Authors use a graph based description, which is similar to ours [FJ02], but their method only considers inclusion, while we also describe adjacency between drawing elements. Their approach has two drawbacks. First, they use a restricted number of basic shapes (circle, line and polygon) to classify strokes. Second, their approach is difficult to scale to large databases of drawings, since they explicitly compare the query with all the figures in the database.

Elad et al present a technique to searching for similar objects in a database of 3D objects [ETA01], specified using VRML. Authors tried to address the subjective matter of object similarity, by providing an approach based not only on geometric feature similarity but also letting the user influence subsequent searches by marking some of the results as "good" or "bad". This algorithm, which is both iterative and interactive, uses statistical moments computed from 3D objects' surfaces as features to define object signatures. Experiments exhibit promising results, since after very few iterations (usually, 1-4) searching seems to converge to objects that users had in mind. However, it is not clear that this approach scales well for large collections of 3D Objects, since they do not use any indexing structure to avoid a sequential scan of the complete database. Authors suggest that their work can be extended with new features, such as topological traits, color and texture.

Chen et al proposed a system for retrieving 3D models based on the visual similarity among objects [CTSO03], using 2D projections. The main idea is that if two 3D objects are similar then they will also look similar from all viewing angles. To apply this idea, the authors encode multiple orthogonal projections (2D views) of a 3D object using both Zernike moments and Fourier descriptors. To create query models users can use an existent 3D model or draw 2D shapes. Authors demonstrate that their approach provides better performance than other competing approaches. Although authors use a scheme to speed up the retrieval process, they can not avoid the comparison between the query and each of the models in the database. For databases with a large number of models this procedure is impracticable, requiring some kind of indexing.

Funkhouser et al [FMK*03] describe a method for retrieving 3D shapes using textual keywords, 2D sketched contours, 3D sketches or a combination of both. The authors developed a new 3D shape descriptor based on spherical harmonics that is descriptive, concise, efficient to compute and invariant to rotations. While their approach relies on silhouettes and their fitting to projections of 3D images, our method, which is based on structural matching of graphical constituents, uses both shape and spatial relations.

Shokoufandeh et al presented a framework, in its preliminary stage, for shape matching through scale-space decomposition of 3D models [BSRS03]. Their algorithm is based on efficient hierarchical decomposition of metric data using its spectral properties. 3D objects are mapped into binary rooted trees, thus recasting the problem of finding a match between 3D models as the much simpler technique of comparing rooted trees. The authors are planning to use the structural parameters of tree representation as signatures for indexing purposes. This way, they expect to avoid the comparison between a query and every model in the database.

More recently Lou et al presented an approach for searching 3D engineering shapes [LPR04]. Their system incorporates multiple feature vectors, relevance feedback, query by example, browsing and a multidimensional indexing structure to support efficient execution. The system extracts four feature vectors, namely principal moments, moments invariants, geometric parameters and eigenvalues of the skeletal graph, to represent 3D objects. A drawback of using various types of features vectors to describe shapes, is the space needed to store descriptors. Authors also proposed and evaluated a multistep similarity search strategy, that combines moment invariants and geometric parameters, to improve effectiveness. Experimental evaluation revealed that this technique presents a precision value 51% higher than the best individual feature vector (principal moments). Another drawback of this system is that it only supports query by example, requiring the use of shape models, either created using a 3D modeling tool or picked from a set of shapes sampled from the database.

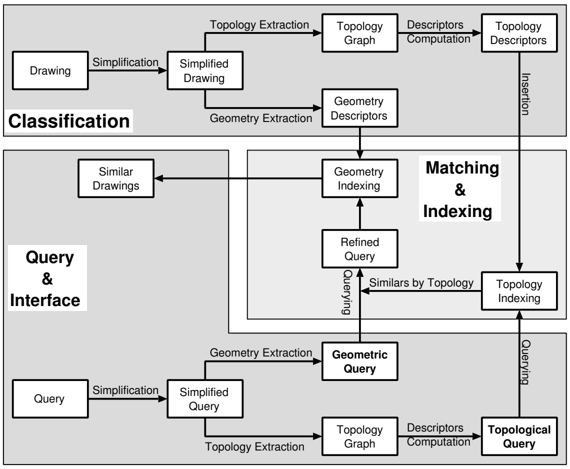

Figure 1: *Detailed architecture for our approach.*

Looking at the majority of existing content-based retrieval systems for drawings, we can observe three things. The first is scalability: most published works use databases with few elements (typically less than 100). The second is complexity: figures stored in the database are simple elements not representing sets of real drawings. Third, complex matching schemes (graph matching) make it difficult to adopt efficient algorithms.

We will shortly describe our approach for content-based retrieval that improves on Berchtold [BK97] and Park [PU99] systems, since we aim to retrieve technical CAD drawings and privilege the use of spatial relationships and dominant shapes. Indeed, our method is more ambitious in the sense that we do automatic simplification, classification and indexing of existing drawings, to make the retrieval process both more effective and accurate.

3. Overview of Our Approach

Our approach to solve both scalability, complexity and matching problems uses a mechanism for retrieving vector drawings, in electronic format through hand-sketched queries, which takes advantage of user's natural ability at sketching and drawing. Furthermore, unlike the majority of systems cited in the previous section, our technique was developed to support large sets of drawings. To that end, we devised a multidimensional indexing structure (NB-Tree) that scales well with growing data set size.

Figure 1 presents a detailed diagram of our system architecture, identifying its main components, which we describe in the remainder of this section.

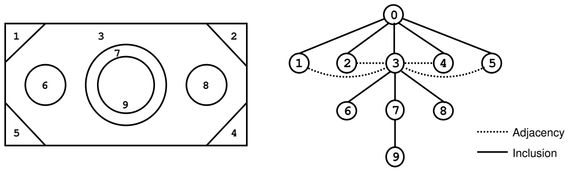

Figure 2: *Drawing (left) and correspondent topology graph (right).*

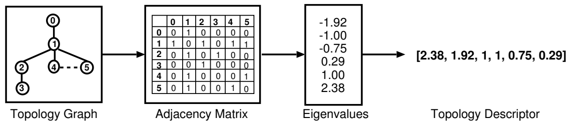

Figure 3: *Block diagram for topology descriptor computation.*

3.1. Classification

Content-based retrieval of pictorial data, such as digital images, drawings or graphics, uses features extracted from the corresponding picture. Typically, two kinds of features are used. Visual features encode information, such as color, texture and shape. Relationship features describe topological and spatial relationships among objects in a picture. While digital images rely mainly on color and texture to describe their content, for vector drawings these features are irrelevant. Thus, we focus on topology, a global feature of drawings, and geometry, a local feature.

Our classification process starts by applying a simplification step, to eliminate most useless elements. We try to remove visual details (i.e. small-scale features) while retaining the perceptually dominant elements and shapes in a drawing, in order to speed up queries.

After simplification we identify visual elements, namely polygons and lines, and extract shape and topological information from drawings, using two relationships, **Inclusion** and **Adjacency**, which are a simplified subset of the relationships defined by Max Egenhofer [EAT92]. The original set of topological relationships includes eight distinct relations between two planar regions, namely `disjoint, contain, inside, meet, equal, cover, covered-by` and `overlap`. We group these into three main categories: `Disjoint`, `Adjacent`, which includes `meet` and `overlap`, and `Inclusion`, which spans the other five classes. The relationships thus extracted are then compiled in a Topology Graph, where "parent" edges mean `Inclusion` and "sibling" connections mean `Adjacency`, as illustrated in Figure 2. Even though our relationships are less discriminating than Egenhofer's original set, they hold regardless of rotation and translation. Evidently, the limitations of this scheme lie in that only two very simple spatial relations are considered.

However, topology graphs are not directly used for searching similar drawings, since graph matching is a NP-complete problem. We use the corresponding graph spectra instead. For each topology graph to be indexed in a database we compute descriptors based on its spectrum [CRS97]. In this way, we reduce the problem of isomorphism between topology graphs to computing distances between descriptors. To support partial drawing matches, we also compute descriptors for sub-graphs of the main graph. Moreover, we use a new multilevel description scheme to describe drawings, by dividing them in different levels of detail and then computing descriptors at each level. This combination of subgraph descriptors and levels of detail, provides a powerful way to describe and search both for drawings or subparts of drawings, which is a novel feature of our work.

To compute the graph spectrum we start by determining the eigenvalues of its adjacency matrix. The resulting descriptors are multidimensional vectors, whose size depends on graph (and corresponding drawing) complexity. Very complex drawings will yield descriptors with higher dimensions, while simple drawings will result in descriptors with lower size. Figure 3 illustrates how descriptors are computed from topology graphs. Descriptors are stored in a database indexing all graphs. It can be shown that descriptors computed in this manner are stable with respect to small changes in graph topology [CRS97]. Furthermore, from experiences performed with 100,000 randomly generated graphs versus a set of 10 candidate similar graphs, we have observed that collisions with descriptors of very different graphs still allow us to retrieve the most likely graphs reliably [Fon04].

To acquire geometric information about drawings we use a general shape recognition library which is able to identify a set of geometric shapes and gestural commands called CALI [FJ01, FJ00]. This enables us to use either drawing data or sketches as input, which is a desirable feature of our system, as we shall see later on. Furthermore it allows us to use geometric features instead of polygon classifications. In this manner we can index and store potentially unlimited families of shapes. After submitting each polygon of the drawing to this recognition library, we end up with a set of multidimensional feature vectors that describe their geometry.

The geometry and topology descriptors thus computed are inserted in two different indexing structures, one for topological information and another for geometric information, respectively.

3.2. Query

Our system includes a Calligraphic Interface [Jor94] to support the specification of hand-sketched queries, to supplement and overcoming limitations of conventional textual methods. The query component performs the same steps as the classification process, namely simplification, topological and geometric feature extraction, topology graph creation and descriptor computation. This symmetrical approach is unique to our method. In an elegant fashion two types of different information (vector drawings + sketches) are processed by the same pipeline.

We developed a new multidimensional indexing structure, the NB-Tree [FJ03, Fon04], which provides an efficient indexing mechanism for high–dimensional data points of variable dimension. The NB-Tree is a simple, yet efficient indexing structure for high–dimensional data points of variable dimension, using dimension reduction. It maps multidimensional points to a 1D line by computing their Euclidean Norm. In a second step we sort these points using a B^+-Tree on which we perform all subsequent operations. Moreover, we exploit B^+-Tree efficient sequential search to develop simple, yet performant methods to implement point, range and nearest-neighbor queries.

Computing the similarity between a hand-sketched query and all drawings in a database can entail prohibitive costs especially when we consider large sets of drawings. To speed up searching, we divide our matching scheme in a two-step procedure. The first step searches for topologically similar drawings. This is accomplished by performing a KNN query to the topology indexing structure, using the descriptor computed from the sketched query. Results returned by the indexing structure represent a set of descriptors similar (near in the space) to the query descriptor. Each returned descriptor corresponds to a specific graph or subgraph stored in the topology database, which will be used in geometry matching. Filtering based on topology drastically reduces the number of drawings to compare, by selecting only drawings with a high likelihood of being similar to the sketched query. In the second step we use geometric information to further refine the set of candidates. At the end, we have a measure of similarity between the sketched query and drawings retrieved from the database.

4. Retrieving 2D Technical Drawings

Using the approach described before, we developed a system to retrieve technical drawings from the mould industry, through a hand-sketched query [FFJ04a, FFJ04b]. Our system retrieves sets of technical drawings similar to queries, from large databases. Figure 4 depicts a screen-shot of our application, where we can see on the left the sketch of a part from the injection side and on the right the results returned by the implied query. Results are ordered from top to bottom, with the most similar on top.

We also developed the simplification and classification component, to extract topological and geometric information from drawings. Information thus collected was used to compute geometric descriptors and to create the topology graph, and correspondent topological descriptors. Resulting descriptors were then inserted into two indexing structures (NB-Trees), one for topology and another for geometry.

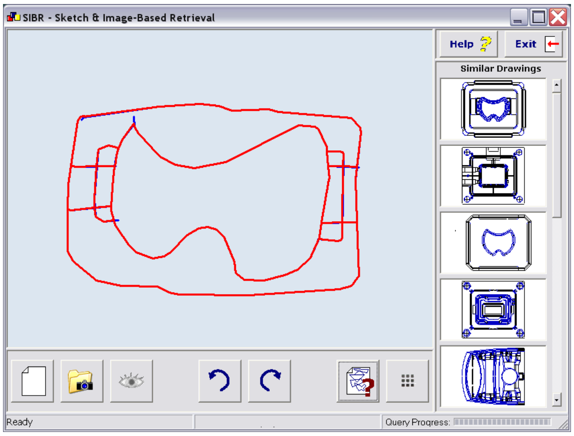

Figure 4: *Sketch-Based Retrieval prototype.*

4.1. Experimental Evaluation

To evaluate the current status of this prototype and the performance of its algorithms, we made a preliminary usability test with three users at CENTIMFE (a technological training center for the Portuguese Mould Industry). The evaluation included three parts. First, we explained the experiment and introduced the prototype to users. During the second part, users executed two sets of queries using the prototype, first by sketching basic drawings and then querying for simple technical drawings. Finally, users answered a questionnaire, where we tried to figure out their profile, opinions about the prototype and their evaluation of the user interface. We also asked for informal suggestions and ideas to improve the current version of the prototype. This experiment involved a database with 78 drawings and 12 queries.

Notwithstanding the low number of users involved, preliminary results are very encouraging. In the majority of the queries, sought drawings were among the first five results. Furthermore, the correct answer was almost always within the first ten returned. One of the things that we observed was that users did not care about where in the order of retrieval the intended drawing appeared, the important fact being that it was present.

In summary, users liked the interaction paradigm very much (sketches as queries), were satisfied with the returned results and pleased with the short time they had to spend to get what they wanted in contrast to more traditional approaches.

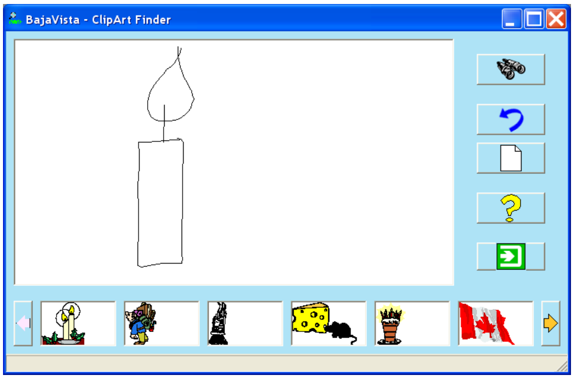

Figure 5: *Clip-art finder prototype.*

5. Retrieving Clip-Art Drawings

Another prototype developed using our retrieval approach was Bajavista [FBRJ04a], an application to index and retrieve clip-art drawings by content [FBRJ04b]. This allows the search and retrieval of drawings through sketches or refined by using one of the results as query (query-by-example). Experimental tests revealed that this prototype can already handle databases with thousands of drawings using commodity hardware.

Figure 5 depicts a screen-shot of our application. On the top-left we can see the sketch of a cloud and on the bottom results returned by the implied query. These results are ordered from left to right, with the most similar on the left. It is also possible to perform Query-by-Example by allowing the user to select a result and use it as the next query, since our classification scheme handles graphics and sketches in the same manner.

This system also includes heuristics to simplify drawings. These eliminate redundant information and useless elements from vector graphics, increasing the processing speed and reducing the complexity of extracted information. The classification component of this system extracts topological and geometric information from drawings, converts it in descriptors and inserts these into the correspondent topological and geometric indexing structures.

5.1. Experimental Evaluation

To evaluate Bajavista we conducted preliminary usability tests with twelve users, 8 male and 4 female, from 23 to 31 years old, who performed two tasks and answered a questionnaire at the end. The experiment involved a database of 968 clip-art drawings and seven queries. These drawings were classified using our multilevel scheme to derive descriptors for each level of detail and for each subpart. This way, we not only describe the overall drawing, but we also describe subparts and different levels of representation of it. Resulting descriptors were then inserted into NB-Trees.

Our experiment was made of three parts. First, we gave users a brief description of our prototype, and allowed them to practice using the digitizing tablet. Second, users performed two tasks, one where they searched drawings from a verbal description and other where we provided sample drawings. Finally, users answered to a questionnaire.

Experimental results show that the BajaVista system performs better for drawings which contain collections of easy-to-draw shapes, with a strong topological component. In these cases, the topological filtering is effective and reduces the number of drawings to compare in the geometric matching.

In summary, we can say that our generic approach for content-based retrieval works better for drawings with a very rich topological structure (technical drawings). In other application domains where geometry predominates over topology (clip-art drawings), the first filtering based on topology is not fruitful. For example, in the BajaVista system we notice that the best results were achieved when we provide reach topological information in the query.

6. Towards Retrieving 3D Objects

We have shown in the previous section that our generic approach for content-based retrieval presents good results for indexing and retrieving 2D drawings. Currently, we are extending this approach to the domain of 3D objects. We started some initial work to integrate this retrieval mechanism into a 3D modeling tool towards the definition of a new interaction paradigm based on suggestions. In this section, we shortly describe the modeling tool, by presenting its interaction paradigms, the expectation lists and how it allows creating 3D objects. Finally, we propose a seamless way of integrating the retrieval component, queries and returned results, into the modeling tool through the expectation lists.

6.1. Overview of the Modeling Tool

GIDeS is a system for creating geometric models through Calligraphic Interaction [PJBF03]. Its main goal is to improve on the usability of CAD systems at the early stages of product design. To this end it combines different paradigms: First, a *calligraphic* sketching metaphor provides for a paper-like interaction. Second, dynamic menus - *expectation lists* - try to expose the state of the application without interfering with the task. Third, an incremental drawing paradigm allows precise drawings to be progressively constructed from sketches through simple constraint satisfaction. Finally, reducing instruction set and command usage allow for a simple and learnable approach in contrast with the complexity of present-day interactive systems.

To deal with ambiguous input the GIDeS system uses expectation lists, a kind of non-intrusive context-based dynamic menus, that free users from memorizing modeling

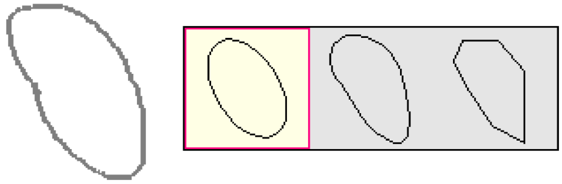

Figure 6: *Expectation lists.*

gestures and constructs. Whenever users' strokes are ambiguously recognized, the application displays a menu with icons that correspond to two or more possible different interpretations of the input. Figure 6 illustrates how expectation lists deal with ambiguity and explore it to user's benefit. In this case the designer sketched a stroke that resembles an ellipse. The resulting expectation list suggests an accurate ellipse as its default option as well as other possible interpretations. Users can then select one of the suggestions or continue drawing, in which case the default option is selected.

6.2. 3D Object Description

As we have seen, GIDeS provides for constructing 3D objects from 2D interaction. In order to integrate our retrieval mechanism into the modeling tool, we must have a logical database with the description of all existing 3D objects that can be searched and included during a modeling task. Thus, for each 3D object we must compute a signature that describes it. While for 2D drawings we used topological and geometric information to compute descriptors, for 3D objects we are using the spatial relationship between faces and edges in a boundary representation.

Thus, for each 3D object we create a graph describing face organization and another graph describing the connections between edges. Then, from these two graphs we compute descriptors using graph eigenvalues [CRS97], as we did for topology graphs of 2D drawings. However, since face and edge graphs (see Figure 7) do not have the same structure of the topology graphs described before (see Figure 2), we are not able to apply our multilevel method, which allows describing subparts of objects. Resulting descriptors are then inserted into two indexing structures, one for each type of graph, using our NB-Tree. Thus, as we did for 2D drawings, 3D objects are described using multidimensional feature vectors and the similarity between them is converted into the computation of a distance between descriptors.

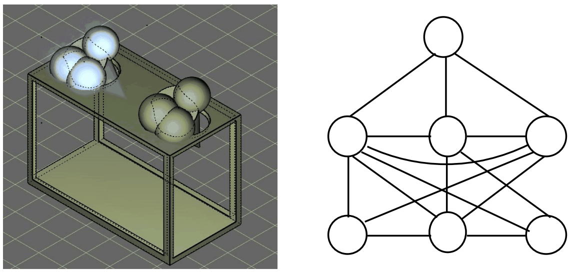

Figure 7: *3D object (left) and correspondent faces graph.*

Finally, to match the query with existing objects in the database, we compute descriptors for the query and search both indexing structures separately (one with information about face graphs and the other with information about edge graphs). To compute the final similarity between the query and objects in the database, we combine both similarity measures, one from the face graph and another from the edge graph.

6.3. Query Formulation and Execution

Given that the retrieval method is very efficient, contrary to other systems where users explicitly have to specify a query and execute it, we propose an approach where the editing tool creates and executes queries automatically without user intervention. In this way, the modeling tool is always "looking" at users' actions, and whenever it detects changes to an object, it searches the database and suggests similar drawings using expectation lists.

Figure 8 shows a modeling sequence, where the user is sketching an object and the system is suggesting results through the expectation lists. This figure illustrates the different types of suggestions provided by the GIDeS system. On top we have the suggestion of simple geometric volumes, in the middle we have modeling operations and finally, on the bottom we have the new type of suggestions provided by the content-based retrieval module. Through this interaction paradigm users can select one of the suggested objects or continue to draw. The main advantage of this scheme is that it is less intrusive than other approaches. Moreover, this exploits the familiar interaction paradigm to provide query as extension of drawing in a natural manner.

6.4. Experimental Results

To validate the extrapolation of our indexing and retrieval approach to the domain of 3D objects we performed experimental tests. To that end we created a data set with 17 objects, like those from Figure 9. Basically, we have two instances of five different "ice cream" objects, (see Figure 9 left), yielding a total of ten ice creams, three objects like the one in Figure 9 right and five ice cream supports (bottom part of Figure 9 left).

Results show that our method is able to correctly isolate the different types of object. Moreover, it considers the two instances of each ice cream object as similar objects, isolates the three objects from the rest of the database and finally, returns ice cream supports as similar objects between each other.

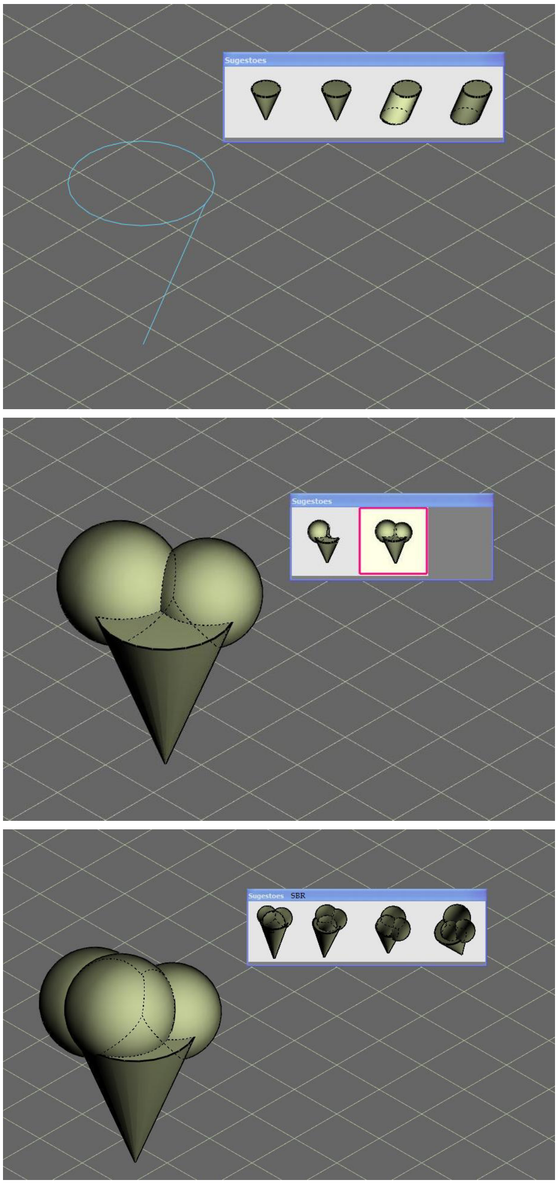

Figure 8: *Modeling sequence using the suggestions mechanism.*

Although our set of 3D objects is very small (only 17 objects) these results are very promising, showing that face and edge graphs can be used with success to describe and retrieve 3D objects.

7. Conclusions and Future Work

We have described a generic approach for content-based retrieval that has already been used with success for retrieving complex 2D drawings. This paper introduces also a novel interaction paradigm for modeling based on suggestions and implicit retrieval of similar objects that integrates a retrieval mechanism with expectation lists. To that end we started by extending the content-based retrieval method to the domain of 3D objects, through the use of face and edge graphs.

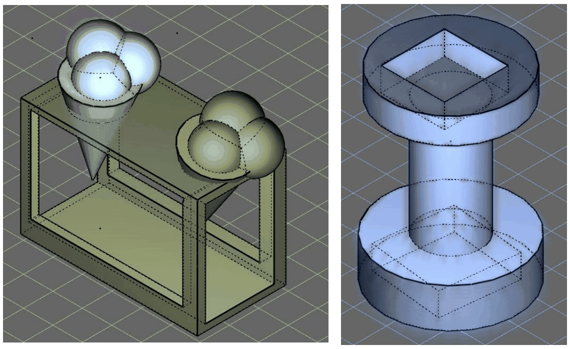

Figure 9: *Type of objects in the database.*

Preliminary results show promising results and suggest that this approach can be successfully extended to three dimensional objects. Although our technique supports large sets of objects, the preliminary studies reported herein were performed using a small data set containing only 17 samples. In the near future we intend to evaluate our interaction scheme using larger data sets, which we are already developing.

Acknowledgements

This work was funded in part by the Portuguese Foundation for Science and Technology, project 34672/99 and the European Commission, project SmartSketches IST-2000-28169.

References

[Bak90] BAKERGEM D. V.: Image Collections in The Design Studio. In *The Electronic Design Studio: Architectural Knowledge and Media in the Computer Age* (1990), MIT Press, pp. 261–272.

[BK97] BERCHTOLD S., KRIEGEL H.-P.: S3: Similarity in CAD Database Systems. In *Proceedings of the International Conference on Management of Data (SIGMOD'97)* (Tucson, Arizona, USA, May 1997), ACM Press, pp. 564–567.

[BSRS03] BESPALOV D., SHOKOUFANDEH A., REGI W. C., SUN W.: Scale-space representation of 3d models and topological matching. In *ACM Symposium on Solid Modeling and Applications* (June 2003), ACM, pp. 208–215.

[Con00] CONSORTIUM S.: SmartSketches Project (IST-2000-28169). `http://sketch.inesc.pt/`, 2000.

[CRS97] CVETKOVIĆ D., ROWLINSON P., SIMIĆ S.: *Eigenspaces of Graphs*. Cambridge University Press, United Kingdom, 1997.

[CTSO03] CHEN D.-Y., TIAN X.-P., SHEN Y.-T., OUHYOUNG M.: On Visual Similarity Based 3D Model Retrieval. *Computer Graphics Forum 22*, 3 (September 2003), 223–232.

[CW91] CLAYTON M., WIESENTHAL H.: Enhancing the Sketchbook. In *Proceedings of the Association for Computer Aided Design in Architecture (ACADIA'91)* (Los Angeles, CA, 1991), pp. 113–125.

[Do95] DO E. Y.: What's in a Diagram that a Computer Should Understand? In *Proceedings of The Sixth International Conference on Computer Aided Architectural Design Futures (CAADF'95)* (1995), The Global Design Studio, pp. 103–114.

[Do98] DO E. Y.: *The right tool at the right time*. PhD Thesis, Georgia Institute of Technology, Sept. 1998.

[EAT92] EGENHOFER M. J., AL-TAHA K. K.: Reasoning about Gradual Changes of Topological Relationships. In *Theory and Methods of Spatio-Temporal Reasoning in Geographic Space*, Frank A., Campari I.,, Formentini U., (Eds.), vol. 639 of *Lecture Notes in Computer Science*. Springer-Verlag, Pisa, Italy, September 1992, pp. 196–219.

[ETA01] ELAD M., TAL A., AR S.: Content Based Retrieval of VRML Objects - An Iterative and Interactive Approach. In *Proceedings of the sixth Eurographics workshop on Multimedia 2001* (Manchester, UK, September 2001), pp. 97–108.

[FBRJ04a] FONSECA M. J., BARROSO B., RIBEIRO P., JORGE J. A.: Bajavista. `http://immi.inesc-id.pt/projects/bajavista/`, 2004.

[FBRJ04b] FONSECA M. J., BARROSO B., RIBEIRO P., JORGE J. A.: Retrieving Vector Graphics Using Sketches. In *Proceedings of the 4th International Symposium on Smart Graphics (SG'04)*, Butz A., Krüger A.,, Olivier P., (Eds.), vol. 3031 of *Lecture Notes in Computer Science*. Springer-Verlag, May 2004, pp. 66–76.

[FFJ04a] FERREIRA A., FONSECA M. J., JORGE J. A.: Sketch-Based Retrieval Prototype for Mould Drawings. `http://immi.inesc-id.pt/projects/sbr/`, 2004.

[FFJ04b] FONSECA M. J., FERREIRA A., JORGE J. A.: Content-Based Retrieval of Technical Drawings. *Special Issue of International Journal of Computer Applications in Technology (IJCAT) "Models and methods for representing and processing shape semantics" (To appear)* (2004).

[FJ00] FONSECA M. J., JORGE J. A.: CALI : A Software Library for Calligraphic Interfaces. INESC-ID, available at `http://immi.inesc-id.pt/cali/`, 2000.

[FJ01] FONSECA M. J., JORGE J. A.: Experimental Evaluation of an on-line Scribble Recognizer. *Pattern Recognition Letters 22*, 12 (2001), 1311–1319.

[FJ02] FONSECA M. J., JORGE J. A.: Towards Content-Based Retrieval of Technical Drawings through High-Dimensional Indexing. In *Proceedings of the 1st Ibero-American Symposium in Computer Graphics (SIACG'02)* (Guimarães, Portugal, July 2002), pp. 263–270.

[FJ03] FONSECA M. J., JORGE J. A.: Indexing High-Dimensional Data for Content-Based Retrieval in Large Databases. In *Proceedings of the 8th International Conference on Database Systems for Advanced Applications (DASFAA'03)* (Kyoto, Japan, Mar. 2003), IEEE Computer Society Press, pp. 267–274.

[FMK*03] 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 (January 2003).

[Fon04] FONSECA M. J.: *Sketch-Based Retrieval in Large Sets of Drawings*. PhD thesis, Instituto Superior Técnico, Technical University of Lisbon, Lisboa, Portugal, July 2004.

[GD96] GROSS M., DO E.: Demonstrating the Electronic Cocktail Napkin: a paper-like interface for early design. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI'96)* (1996), pp. 5–6.

[Jor94] JORGE J. A.: *Parsing Adjacency Grammars for Calligraphic Interfaces*. PhD thesis, Rensselaer Polytechnic Institute, Troy, New York - USA, Dec. 1994.

[LC02] LEUNG W. H., CHEN T.: User-Independent Retrieval of Free-Form Hand-Drawn Sketches. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP'02)* (Orlando, Florida, USA, May 2002), vol. 2, IEEE Press, pp. 2029–2032.

[LC03] LEUNG W. H., CHEN T.: Hierarchical Matching for Retrieval of Hand-Drawn Sketches. In *Proceedings of the IEEE International Conference on Multimedia and Exposition. (ICME'03)* (Baltimore, USA, July 2003), vol. 2, IEEE Press, pp. 29–32.

[LPR04] LOU K., PRABHAKAR S., RAMANI K.: Content-based Three-Dimensional Engineering Shape Search. In *Proceedings of the 20th International Conference on Data Engineering (ICDE'04)* (Boston, USA, April 2004), pp. 754–765.

[PJBF03] PEREIRA J. P., JORGE J. A., BRANCO V. A., FERREIRA F. N.: Calligraphic interfaces: Mixed metaphors for design. In *10th International Workshop on the Design, Specification and Verification of Interactive Systems (DSV-IS'03)* (Funchal, Madeira, Portugal, June 2003).

[PU99] PARK J., UM B.: A New Approach to Similarity Retrieval of 2D Graphic Objects Based on Dominant Shapes. *Pattern Recognition Letters 20* (1999), 591–616.