Modeling of 2D Parts Applied to Database Query

Guillaume-Alexandre Bilodeau and Robert Bergevin
Laboratoire de vision et systèmes numériques, Pavillon Adrien-Pouliot
Université Laval, Sainte-Foy (QC), Canada, G1K 7P4
bilodeau@gel.ulaval.ca, bergevin@gel.ulaval.ca

Abstract

This paper presents recent developments in a project aimed at the design of an image database query engine, where the images are searched using 3D part-based object models. This is a novelty since most existing image database query engines search images by comparing colors, textures and 2D shape of regions in the images. In our project, 3D part-based models are built from qualitative volumetric primitives. This paper proposes a method to hypothesize such volumetric primitives from projected object parts. It combines concepts from two existing approaches, a model-fitting and a rule-based approach. Using fuzzy logic, this new method can produce multiple hypotheses to attain the robustness necessary for processing 2D parts originating from 2D images of real scenes. A detailed description of the method is presented along with promising preliminary results.

1 Introduction

A problem still unresolved in computer vision is the comparison of objects in different 2D images using efficient and reliable algorithms. This is similar to the problem of identifying an object in an image. In the case of the comparison of objects, a value of similarity is obtained as the result of the computations. In contrast, for the identification of an object, one or more identifiers are produced as the result. The images are processed with common algorithms, but the results obtained are interpreted differently. The resolution of these two problems is of high interest as it permits the development of autonomous robots and efficient image database query engines.

Our present work aims precisely at developing robust algorithms to model and compare 3D objects in 2D images in the context of an image database query engine. In this paper, one specific aspect is presented, that is modeling 2D parts. In the context of this work, 2D parts are defined as regions delimited by groups of constant curvature primitives (CCPs), which correspond to the projections in the plan of simple volumetric primitives, like cylinders and prisms.

We will show that our hybrid method combines the benefits of traditional approaches and is suited for processing 2D parts originating from real 2D images.

The paper is structured as followed. Section 2 gives an overview of the application. Section 3 provides our basic strategy to model 2D parts. Section 4 and 5 gives the specifics of the method and shows example results. Finally, Section 6 concludes the paper.

2 Overview of the application

The application under development aims at querying an image database of manufactured objects, which are arrangements of simple volumetric primitives. Images in the database are from real scene with an object in the foreground. The only other constraint is that the object must be detectable in the image [1].

Figure 1 shows an overview of the image database query engine under development. The shaded region represents the four algorithmic steps required to add an image or query the database. The database is composed of various 2D images of 3D objects, and their associated models. To query or add an image to the database, the user gives as input an example 2D image or a sketch of the 3D object. The image is first processed to obtain its CCPs map and the outline of the object (Object detection). This CCPs map is then processed to obtain parts using the outline (Part segmentation). These parts are then labelled based on the possible volumetric primitives that may project onto them

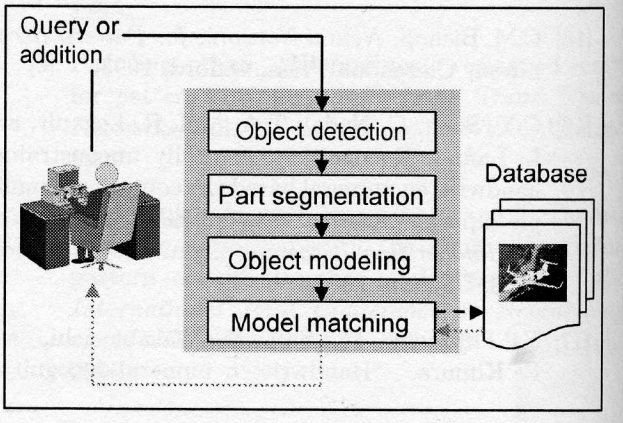

Figure 1: Overview of the application

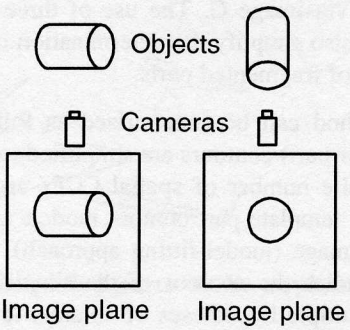

Figure 2: Different projections of the same volumetric primitives.

(Object modeling). Parts are modeled by volumetric primitives because the aspect of a projected 3D object may change significantly for different viewpoints (Figure 2). At the object modeling step, the spatial relationships between parts are also computed. Finally, the constructed model is compared with the models in the database (Model matching). If similar models are in the database, the corresponding 2D images are shown to the user. If not, the newly built model and its corresponding image may be added to the database. The model matching step has not yet been implemented.

Therefore, the general goal of the image database query engine is to show to the user the images of the database, which resemble the most the query image. The outputted images will be classified from the most to the least similar image based on the score obtained during the matching step. As mentioned in the introduction, the topic of this paper is step 3, object modeling, and specifically part modeling. The following section describes our method to resolve this problem.

3 Basic Strategy

3.1 Two existing approaches

Two main approaches exist to infer volumetric primitives from 2D parts. The model-fitting approach and the rule-based approach. The first approach attempts to infer the volumetric primitives by fitting their projections onto the 2D parts in the image. The best volumetric primitive hypothesis is the one that minimizes the fitting error between the contour of its projection and the contour of the projection in the image. In general, optimization is done using a deformable volumetric primitive model like superquadrics [2]. The second approach studies the spatial arrangements of the CCPs forming the contour of the 2D parts and optionally the interior CCPs enclosed by this contour [3-5]. A set of inference rules based on these arrangements are associated with each volumetric primitive the system can infer. The rules are then applied onto the 2D parts in the image to obtain the volumetric primitive hypotheses.

The main advantage of a model-fitting approach is its ability to model 2D parts having unexpected spatial arrangements of CCPs. However, as the CCPs forming the 2D part boundary are not studied, the inference may not be exact. For example, a circular arc (CA) can be fitted onto several straight line segments (SLSs) with a low fitting error. In that case, the 2D part may be labelled as having a boundary made of CAs, instead of SLSs.

The rule-based approach has the advantage of being easy to implement to generate multiple volumetric primitive hypotheses. In addition, if the CCPs of all the 2D parts in an image respect at least one inference rule, the results are very good. However, this is seldom the case for parts extracted from real images. If the CCPs of a 2D part do not respect any inference rule, this approach will not be able to label it.

3.2 Our hybrid method

Figure 3 gives two examples of parts obtained at the part segmentation stage of our system. They are obtained by first extracting the CCPs using MAGNO [6] which is based on a standard Canny edge detector and a contour extraction and segmentation technique. Then, the CCPs are grouped into 2D parts using cocurvilignity, symmetry, proximity, similarity and the outline of the object. Details on how these parts are obtained can be found in [1, 7, 8]. These results show that the parts have fragmented contours and often missing interior CCPs. Therefore, methods based on spatial CCPs arrangements involving interior CCPs cannot be used reliably. Furthermore, because these parts are obtained from a real image, their spatial CCPs arrangements are highly diversified, and hence it is difficult to choose a set of inference rules that covers all the possibilities. Finally, multiple hypotheses are required for our system to be capable of matching volumetric primitives projected in different ways on the image plane.

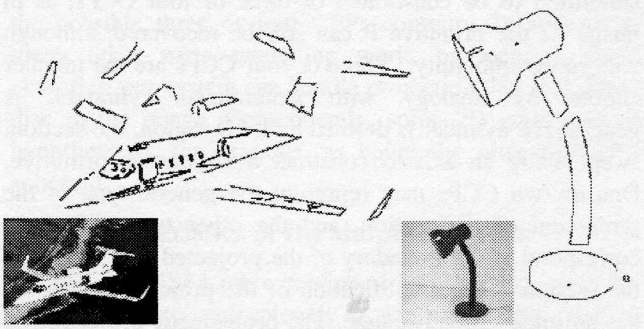

Figure 3: Original parts of an airplane and a desk lamp.

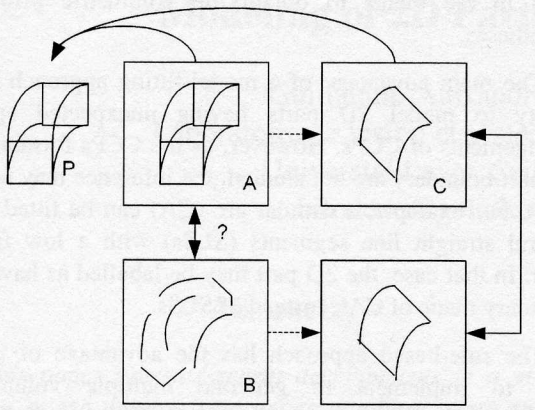

Figure 4: Matching an idealized projection with an actual projection.

The imperfection of the data, and the need to infer volumetric primitives dependant of the type of the boundary CCPs, make it difficult to use either one of the two common approaches for our image database query engine. However, a combination of the two approaches benefits from the advantages of both. Our method is based on this combination of approaches. Figure 4 illustrates the principle of our method. A rule-based classifier can classify only a finite number of projection contours, which are the projection contours that satisfy its rules. The rules of the classifier are selected to identify the projections of chosen volumetric primitives. Hence, if volumetric primitive P in figure 4 is ideally projected in image A, it is possible to design a rule to recover P from image A. Image B shows the projection of the same volumetric primitive P as it may appear in a image of a real scene. As it can be observed, image A and image B differ greatly because of the noise and of the performance of the previous algorithms applied to the image. Consequently, recovering the primitive P from image B is very difficult using only few rules. To recover P, a large number of rules would have to be designed to take into account all the possible imperfections. Another way to deal with these imperfections is based on the following observation. If the ideal projection of P is simplified to be constituted of three or four CCPs, as in image C, the primitive P can still be recovered, although with more ambiguity. Three or four CCPs are the number chosen by analogy with generalized cylinders. A generalized cylinder is defined by a 2D region, the section, swept along an axis to construct a volumetric primitive. One or two CCPs may represent the generic type of the projection of the section, and the other two CPPs may correspond to the boundary of the projected area swept by the section. This simplification of the projection contours has an important advantage. The problematic projection of image B can easily be approximated by three or four CCPs, and thus a relationship between image B and primitive P is established, via image C. The use of three or four CCPs specifically also simplify the determination of the axes and the merging of fragmented parts.

Our method can be summarized as follow. First, the projection (or part) contours are simplified to correspond to one of a finite number of spatial CCPs arrangements. In other words, template part contour models are fitted to the part in the image (model-fitting approach). Next, a fuzzy classifier studies the contour of the simplified parts, and generates multiple hypotheses of volumetric primitives for each part (rule-based approach). The following sections detail the method.

4 Simplification of 2D parts

Simplified parts are computed from the original 2D parts, to transform any possible spatial arrangements of CCPs into spatial arrangements of CCPs the fuzzy classifier can process. To reach this goal, two criteria are optimized under the restriction that the simplified part obtained must be composed of three or four CCPs. The optimization consists in defining a simplified part made of three or four CAs or SLSs, that covers as accurately as possible the area of the original 2D part and that is as rectangular as possible (the angles between the CCPs of the simplified part must be as close as possible to 90°). This rectangularity criterion is based on the fact that during part segmentation, CCPs are grouped using symmetry and parallelism criteria. Therefore, the parts tend to be rectangular. The optimization can be expressed in the following way. Let $C_i = (\#, x, y)$ be a point sampled on the contour of the 2D part, where $\#$ is the sample number and x and y are the coordinates of the point. Let $C = \{C_i\}$ be the set of all the C_i and $CO = (C, \prec_{\#})$ be the ordered set of the elements of C on $\#$. Furthermore, let $\angle C_i$ be the angle between the vector defined by C_i and the preceding point, and the vector defined by C_i and the following point in an ordered set. Let *Area(LP)*, be the area enclosed by the closed cycle defined by linking the points of the ordered set *LP*. Next, let $SO_i = \{C_a, C_b, C_c, C_d\}$ be an ordered set of four points from CO. Finally, let $SO = \{SO_i\}$ be the set of all the possible SO_i. The ordered set of points *SF* making the boundary of the simplified parts is:

$$SF = \min_{SO_i \in SO} \left(\sum_{C_i \in SO_i} (\angle C_i - 90°) + \left| Area(CO) - Area(SO_i) \right| \right) \quad \text{(EQ. 1)}$$

The first term of the sum expresses the rectangularity criterion and the second term of the sum expresses the area coverage criterion. The simplified parts with three CCPs are obtained by removing one of two very close points in SF. Then, a polygon is constructed from SF. The contour of the polygon is compared with the original

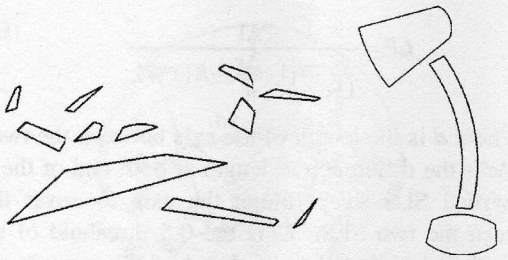

Figure 5: Simplified parts of the airplane and the desk lamp.

contour of the 2D part to replace SLSs by CAs whenever needed to obtain an accurate simplified part. Pairs of consecutive points of *SF* (defining a SLS) are associated with pairs of points of the original part contour. The path between the pair of points on the original part contour is scanned, and if a CA is found, a CA replaces the corresponding SLS of the polygon. Two simplified parts that have similar cross-section, sweeping rule and axis type (CA or SLS) are merged.

Figure 5 shows simplified parts obtained after the application of the optimization criteria and merging.

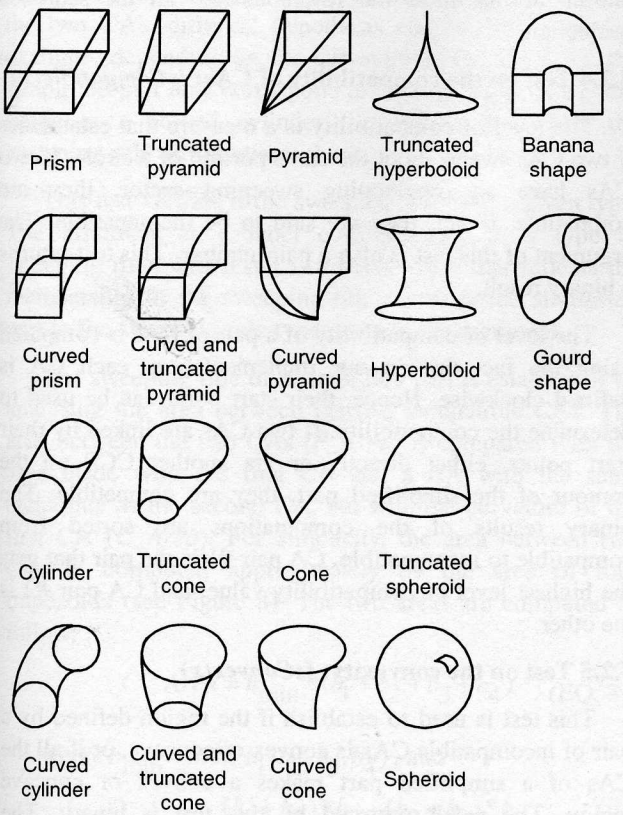

Figure 6: The eighteen volumetric primitives that can be hypothesized.

5 Fuzzy classification of simplified parts

The goal of this step is to determine which volumetric primitives may have given rise to the observed part projection. The fuzzy classifier labels the simplified parts by studying their spatial CCPs arrangements. Since all simplified parts are composed of three or four CCPs, the fuzzy classifier has to process only a finite number of well-defined contour types. Thus, the fuzzy classifier will always generate results if there are rules for each such type of contour. The fuzzy classifier generates multiple hypotheses since different volumetric primitives can generate the same projected contour of three or four CPPs. Each of these hypotheses corresponds to a volumetric primitive and its projected axis.

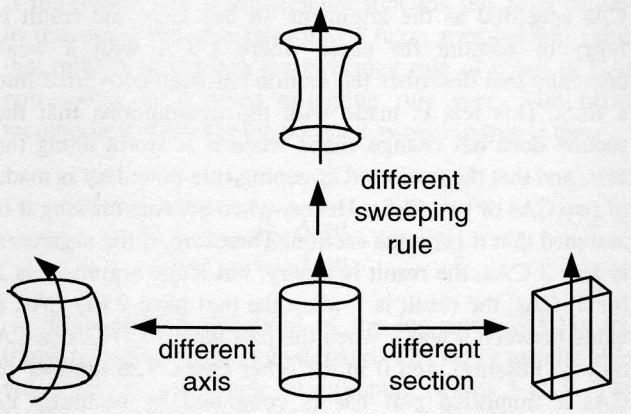

Figure 7: The parameters differentiating the volumetric primitives.

5.1 Volumetric primitives

Eighteen simple volumetric primitives, which can be hypothesized from projection made of three or four CPPs, have been chosen for the 3D inference of 2D parts. These primitives are illustrated in Figure 6. They are differentiated by axis type, section type and sweeping rule type (figure 7). This is similar to the geon primitives defined by Biederman [9]. Their projections generate all the possible three or four CPPs contours. In contrast to many other parts modelling work, by using a fuzzy classifier, our system can handle projection that look like a disc in the image. Consequently, some 2D parts may be hypothesized, for example, as being the projection of a spheroid.

5.2 Characteristics of the simplified parts

The simplified parts processed by the fuzzy classifier can be characterized uniquely by six parameters. That is, the number of CCPs that compose it, the number of CAs among these, the level of parallelism of the SLSs, the level of compatibility of the CAs, the convexity of the CAs and

the sweeping rule. The input values to the fuzzy classifier are provided by six tests that verify the conformity of the simplified part to each of these parameters. The arguments of the tests are labels that specify to which parameter value the simplified part must conform. Below, the six tests are described along with the computations they require.

5.2.1 Test on the number of CPPs: *IsNumberCCPs(x)*

This test is simple, it verifies if the simplified part has the number of CCPs specified as the argument. The result is binary.

The number of CPPs a simplified part has is computed by scanning its contour.

5.2.2 Test on the number of CAs: *IsNumberCAs(x)*

This test verifies if the simplified part has the number of CAs specified as the argument. In this case, the result is fuzzy to account for cases where a CA with a weak curvature that describes the section has been converted into a SLS. This test is made with the assumptions that the section does not change shape when it is swept along the axis, and that the projected sweeping rule boundary is made of two CAs or two SLSs. Hence, when a CA is missing it is assumed that it is on the section. Therefore, if the argument is 1 or 3 CAs, the result is binary, but if the argument is 2 (or 4) CAs, the result is 1 when the part have 2 (4) CAs, a value between 0 and 1 when the part have 1 (3) CA (a CA may be missing), and 0 in the other cases. The number of CAs a simplified part has is computed by scanning its boundary.

5.2.3 Test on the parallelism of SLSs: *IsParallel(x)*

This test return a fuzzy value reflecting the level at which two SLSs have the same orientation. If two SLSs have exactly the same orientation, the value returned is 1, and if the SLSs are perpendicular the value returned is 0. The level of parallelism is fuzzy between these two extremes.

The argument of this test is a SLS pair number. SLS pair #1 is the pair that has the highest computed level of parallelism, and SLS pair #2 is the other one. This will be justified shortly.

The level of parallelism for pairs of SLSs is computed by using concepts inspired of perceptual grouping theory. The level of parallelism is not linear with respect to the angle between the SLSs, and the length and the distance between the SLSs are taken into account. A log-sigmoid function is used. This function gives an incertitude region (where the value are between 0 and 1) that is quite narrow and easily customable. The level of parallelism is expressed as follow:

$$LP = \frac{1}{1+e^{-((1-\frac{\Delta s}{d})-K1)*K2}} \qquad \text{(EQ. 2)}$$

where d is the length of the axis between the two SLSs, and Δs is the difference in length at both end of the axis of the virtual SLS swept along the axis to cover the area between the two SLSs. $K1$ is the 0.5 threshold of the log-sigmoid. It is a function of d and of the average distance between the SLSs. $K2$ is the transition speed of the log-sigmoid. It is a function of d.

$K1$ and $K2$ ensure that the difference of orientations between the two SLSs makes the level of parallelism vary more quickly as the length of the SLSs increase. These two values also ensure that the distance between the SLSs has the same effect.

The level of parallelism computations are applied to each pair of SLSs. The pair that gets the highest value is assigned pair #1, and the other is assigned pair #2. This has been introduced to deal with cases where the parts have two SLSs pairs. The results of the computations are simply sorted, so that if a pair is desired as parallel and the other as not parallel for identifying a given projection of a volumetric primitive, the right combination of pairs is chosen to maximize the resemblance with the searched projection.

5.2.4 Test on the compatibility of CAs: *IsCompatible(x)*

The level of compatibility is a measure that establishes if two CAs sweep about the same portion of a circle. If two CAs have an overlapping sweeping sector, they are compatible. If not, they are said to be incompatible. The argument of this test is also a pair number. This test returns a binary result.

The level of compatibility of a pair of CAs is computed using the fact that in our implementation, each CA is defined clockwise. Hence, their start point can be used to determine the compatibility. If two CAs are linked by their start points, either directly or via another CCP on the contour of the simplified part, they are compatible. The binary results of the computations are sorted from compatible to incompatible. CA pair #1 is the pair that gets the highest level of compatibility value, and CA pair #2 is the other.

5.2.5 Test on the convexity: *IsConvex(x)*

This test is used to establish if the region defined by a pair of incompatible CAs is convex or concave, or if all the CAs of a simplified part makes a convex or concave region. The result returned by this test is binary. The argument is a keyword that indicates if all the CAs must be convex, or just a particular pair.

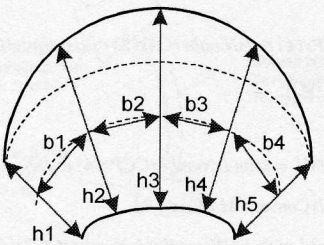

Figure 8: Values used for the determination of the sweeping rule.

As for the compatibility, the convexity of a pair of CAs is computed using the fact that each CA is defined clockwise. In addition, the points sampled on the boundary of the original parts are also obtained by sampling the boundary clockwise. Therefore, convex CAs have their start and end points in the same order as the sampled points. Results are computed for all possible pair of CAs, and for the region made of all the CAs of the simplified part.

5.2.6 Test on the sweeping rule: *IsSweepRule(x)*

This test is applied on compatible CA pairs. For some volumetric primitive hypotheses, the axis is defined by two compatible CAs. However, depending on the curvature of the two CAs, different hypothesis can be generated. For example, depending on the curvature of the two CAs, the simplified part may correspond to the projection of a gourd shape or a curved cylinder (See figure 6). Only the sweeping rule differentiates them.

This test verifies if the sweeping rule is of a given type. The argument is a label corresponding to a type of sweeping rule. The result is a fuzzy value that reflects the membership of the sweeping rule of the tested simplified part to the sweeping rule type given as the argument.

The sweeping rule of a simplified part is established by analysing the area between the two compatible CAs. The area between the two CAs (*P_Area*) is compared with the area made with the first CA and a CA with the same endpoints as the second CA, but with the curvature of the first CA (*C_Area*). For simplicity, the area between two CAs is computed approximately by the area of four trapezoids (see Figure 8). The two areas are computed as follow:

$$C\_Area = h_{\min} * (b_1 + b_2 + b_3 + b_4) \quad \text{(EQ. 3)}$$

where $h_{\min} = \min(h_1, h_2, h_3, h_4)$, and

$$P\_Area = \frac{(h_1 + h_2) * b_1}{2} + \frac{(h_2 + h_3) * b_2}{2} + \frac{(h_3 + h_4) * b_3}{2} + \frac{(h_4 + h_5) * b_4}{2} \quad \text{(EQ. 4)}$$

Three types of sweeping rules are possible for a pair of compatible CAs. That is, constant sweeping rule, growing sweeping rule and growing-shrinking sweeping rule. For a given pair of compatible CAs, only two types of sweeping rules are possible at once. It is always the constant sweeping rule and another type. The ratio between the two computed areas gives a measure of how much the sweeping rule of the pair of compatible CAs differs from a constant sweeping rule. The sweeping rule is also computed by using concepts inspired of perceptual grouping theory. The value of similarity of the simplified part sweeping rule with respect to a constant sweeping rule does not change linearly with the ratio of the areas. A function that allows an uncertainty region (A region where two sweeping rule are possible at once) is necessary, but this uncertainty region must be quite narrow for a good categorization. This is the reason why a log-sigmoid function has been chosen to transform the area ratio into a fuzzy membership value that reflects how much the sweeping rule of a pair of CAs correspond to a given sweeping rule type. The fuzzy membership value for the constant sweeping rule is then:

$$CSR = 1 - \frac{1}{1 + e^{-(\frac{P\_Area}{C\_Area} - K_1) * K_2}} \quad \text{(EQ. 5)}$$

$K1$ is the 0.5 threshold of the log-sigmoid. $K2$ is the transition speed of the log-sigmoid. The fuzzy membership value for the other type of sweeping rule is 1-*CSR*. The length progression of h_1, h_2, h_3 and h_4 determine the identity of the other type of sweeping rule.

5.3 Fuzzy classifier

The fuzzy classifier uses the previously defined tests to generate volumetric primitive hypotheses for each simplified part. A combination of tests that generates a particular set of hypotheses is called a fuzzy classifier rule. To handle any simplified part made of three or four CCPs, the fuzzy classifier has thirty-five rules.

5.3.1 Fuzzy classifier rules

The fuzzy classifier rules use the results of the tests to generate a number of volumetric primitive hypotheses. A fuzzy membership value (FMV) is associated to each hypothesis. This membership value reflects how much the simplified part belongs to the set of projection contours associated with a particular volumetric primitive. To generate the membership values, the rules combine the results of the tests by taking the minimum value returned by the set of tests used. Furthermore, a value of belief multiplies this minimum value for each hypothesis. This value of belief is introduced because, for a given simplified part and a given rule, some volumetric primitive

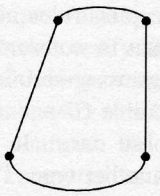

Figure 9: Simplified part for the example.

hypotheses are more likely than others are. This is explained by the fact that the projection corresponding to the simplified part may be observed more often from some volumetric primitives than others, for which it may be an accidental or rare view. The value of belief supports some hypotheses without discarding the ones less likely. Supporting some hypotheses allows the model-matching step to match the models that are more likely similar first. The values of belief are not computed from observation probabilities calculated from a view sphere. The values of belief are used to order the hypotheses. They do not need to represent exactly the observation probability of each hypothesis. Any values that produce an ordering that reflects observation probabilities can be used. The values chosen will influence the distribution of the scores obtained during the model matching step. The impact of the magnitude of the values will be studied when the matching step is implemented.

All the rules uses the following formulation:

$if\ (TestA(x) \cap TestB(x) \cap ...)$

then

X with $FMV = \min(TestA(x) \cap TestB(x) \cap ...) * beliefX$

Y with $FMV = \min(TestA(x) \cap TestB(x) \cap ...) * beliefY$

where X and Y are volumetric primitives, and *beliefX* and *beliefY* are values of belief. Note that since some tests return fuzzy values, many rules can be activated simultaneously.

5.3.2 A simple example

Consider the simplified part of figure 9. This simplified part is differentiated from others, by its number of CCPs (4), its number of CAs (2), the convexity of the CAs (convex) and the level of parallelism of its SLSs (more or less parallel). Projections that resemble this part are the projection of a cylinder (the level of parallelism tends toward 1), the projection of a cone (the level of parallelism tends toward 0) and the projection of a truncated spheroid (the level of parallelism is not important, but this volumetric primitive can have this projection only from few viewpoints). Therefore, rules hypothesizing these volumetric primitives will be activated with different strength. In this case, the two rules activated for this simplified part are:

Rule 8:

$if\ (IsNumberCCPs(4) \cap IsNumberCAs(2) \cap IsParallel(1) \cap IsConvex(ALLCAs))$

then

Cylinder with $FMV = \min(IsNumberCCPs(4) \cap IsNumberCAs(2) \cap IsParallel(1) \cap IsConvex(ALLCAs)) * 1$

TruncatedSpheroid with $FMV = \min(IsNumberCCPs(4) \cap IsNumberCAs(2) \cap IsParallel(1) \cap IsConvex(ALLCAs)) * 0.2$

Rule 9:

$if\ (IsNumberCCPs(4) \cap IsNumberCAs(2) \cap not(IsParallel(1)) \cap IsConvex(ALLCAs))$

then

Cone with $FMV = \min(IsNumberCCPs(4) \cap IsNumberCAs(2) \cap not(IsParallel(1)) \cap IsConvex(ALLCAs)) * 1$

TruncatedSpheroid with $FMV = \min(IsNumberCCPs(4) \cap IsNumberCAs(2) \cap not(IsParallel(1)) \cap IsConvex(ALLCAs)) * 0.2$

where *not()* is the fuzzy negation operator. The numerical values of belief are sample values that give an ordering that takes into account the observation probabilities.

Some volumetric primitives may be hypothesized more than once, since many rules can be activated simultaneously. Because, the fuzzy classifier considers all possible viewpoints, some of these hypotheses are not redundant. For example, a square simplified part may be the projection of a prism seen directly from one of its faces, or a view where more than one faces are visible. For these two projections, the axis is not same. Thus, the hypotheses are not redundant. Therefore, if the same volumetric primitive is hypothesized more than once for the same viewpoint, the maximum value is taken as the final FMV, and if a volumetric primitive is hypothesized more than once, but for different viewpoints, all the hypotheses are kept.

Figure 10 gives the FMV of the three most likely volumetric primitive hypotheses for the simplified parts of the desk lamp and the airplane. When a volumetric primitive is hypothesized twice for a simplified part, the hypothesis corresponding to a view directly perpendicular to a face receives the lowest score, because this type of view is less likely.

The results obtained are in general in conformity with what was expected. A few results may seem odd, but they are justified. Consider the result marked with an X. At first view, the pyramid result may seem wrong. However, if all viewpoints are considered, the simplified part can correspond to a pyramid seen from its top or from its bottom. Hence, this possibility cannot be rejected.

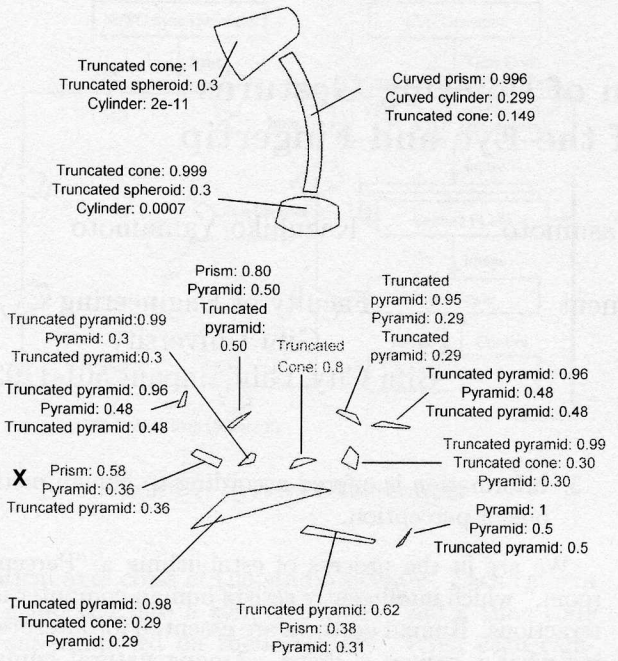

Figure 10: Volumetric primitive hypotheses for the simplified parts of the desk lamp and the airplane.

Furthermore, the results shown were obtained with unadjusted parameters. The shapes of the log-sigmoids used for the level of parallelism and the sweeping rule have to be determined precisely. The values of belief change the magnitude of the FMV also. These parameters will be adjusted during the implantation and the testing of the matching step.

6 Conclusion and future work

This paper has presented a new method for hypothesizing volumetric primitives from 2D projections. This new method is hybrid, as it combines concepts from model-fitting approaches and rule-based approaches. The simplification of 2D parts corresponds to a model-fitting approach and the fuzzy classifier corresponds to a rule-based approach. The combination of the two approaches permit hypothesizing volumetric primitives from 2D parts which do not obey any implemented inference rule. It also allows accounting for the noise and errors inherent to the processing of real 2D images, although at the price of increased ambiguity for identification of some projections. Note however, that the fuzzy classifier has been designed to account for these ambiguities by generating multiple hypotheses.

The results obtained thus far show that the implementation of the hybrid method behaves in general as intended. The performance of the method will be measured more accurately after the model matching algorithms are implemented.

Future work will consist in implementing the model matching algorithms and integrate all the components of the image database query engine. Then the performance of the system will be characterized by using the query engine on a variety of images.

7 References

[1] Bilodeau, G.A. and R. Bergevin. *Generic Modeling of 3D Objects from Single 2D Images.* in *International Conference on Pattern Recognition.* 2000. Barcelona, Spain. pp. 770-773.

[2] Pilu, M. and R.B. Fisher, *Model-Driven Grouping and Recognition of Generic Object Parts from Single Images.* Robotics and Autonomous Systems, 1997(21): pp. 107-122.

[3] Dickinson, S.J., A.P. Pentland, and A. Rosenfled, *3-D Shape Recovery Using Distributed Aspect Matching.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992. **14**(2): pp. 174-198.

[4] Bergevin, R. and M.D. Levine, *Generic Object Recognition: Building and Matching Coarse Descriptions from Line Drawings.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1993. **15**(1): pp. 19-36.

[5] Zerroug, M. and R. Nevatia, *Volumetric Descriptions from a Single Intensity Image.* International Journal of Computer Vision, 1996. **20(1/2)**: pp. 11-42.

[6] Moktari, M. and R. Bergevin. *Multiscale Compression of Planar Curves using Constant Curvature Segments.* in *14th International Conference on Pattern Recognition.* 1998. Brisbane, Australia. pp. 744-746.

[7] Bilodeau, G.A. and R. Bergevin. *Outline-based Part Segmentation using Intermediate-level Symetries.* in *Vision Interface'99.* 1999. Trois-Rivières, Canada. pp. 66-73.

[8] Bilodeau, G.A., *Segmentation en parties d'objets 3D provenant d'images 2D réelles et complexes,* Master thesis. Université Laval. 1999.

[9] Biederman, I., *Human Image Understanding: Recent Research and a Theory.* Computer Vision Graphics Image Processing, 1985. **32**: pp. 29-73.