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 the latest developments in a project aimed at the design of an image database query engine, where the images are searched at the 3D object-level. This is a novelty since the majority of existing image database query engines search images by comparing the colors, the textures and the 2D shape of regions in the images. This paper specifically discusses a new method to hypothesize volumetric primitives from 2D parts (2D regions corresponding to projections of volumetric primitives). Our new hybrid approach combines two existing approaches to benefit from the advantages of both. It combines a model-fitting approach and a rule-based approach. Using fuzzy logic, this new approach can produce multiple hypotheses to attain the robustness necessary for processing 2D parts originating from real 2D images. A detailed description of the approach is presented along with 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 arcs and segments (lines), which correspond to the projections in the plan of simple volumetric primitives, like cylinders and prisms.

We will show that our hybrid approach 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 approach and shows example results. Finally, Section 6 concludes the paper.

2 Overview of the application

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 arc and segment (line) map and the outline of the object (Object detection). This line 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 (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

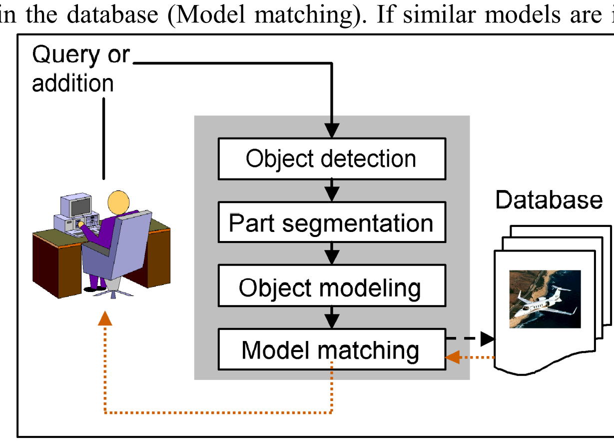

Figure 1: Overview of the application

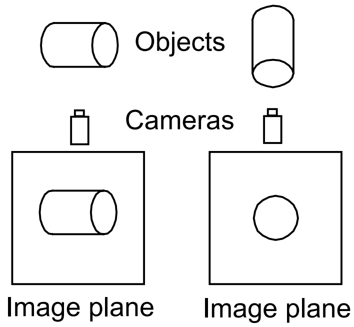

Figure 2: Different projections of the same volumetric primitives.

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.

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 less 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 approach 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 minimize the fitting error between the contour of its projection and the contour of the projection in the image. In general, the optimization is done using a deformable volumetric primitive model like the superquadrics [1]. The second approach studies the spatial arrangements of the lines forming the contour of the 2D parts and optionally the interior lines enclosed by this contour [2-4]. 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 advantage of the model-fitting approach is that it can model 2D parts that have unexpected spatial arrangements of lines. However, as the lines forming the 2D part boundary are not studied, the inference may not be exact. For example, an arc can be fitted onto several segments with a low fitting error. In that case, the 2D part may be labelled as having a boundary made of arcs, instead of segments. With this approach, it is also difficult to have multiple hypotheses.

The rule-based approach has the advantage of being easy to implement to generate multiple volumetric primitive hypotheses. In addition, if the lines 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 lines of a projection do not respect any inference rule, this approach will not be able to label it.

3.2 Our hybrid approach

Figure 3 gives two examples of parts obtained at the part segmentation stage of our system. Details on how these parts are obtained can be found in [5-7]. As it can be observed from these results, the parts have fragmented contours, and their interior lines are often missing. Therefore, methods based on spatial line arrangements involving interior lines cannot be used reliably. Furthermore, because these parts are obtained from a real image, their spatial line 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.

These operating conditions 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. This is the approach we have chosen. Figure 4 illustrates the principle of our approach. 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

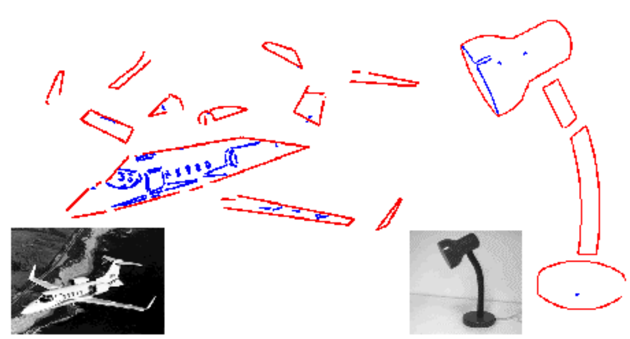

Figure 3: Original parts of an airplane and a desk lamp.

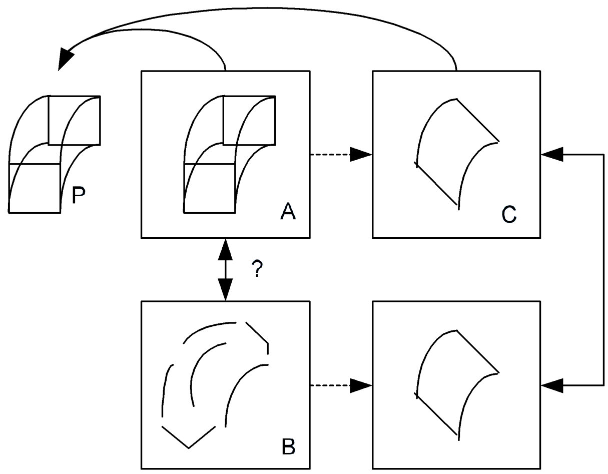

Figure 4: Matching an idealized projection with an actual projection.

B shows the projection of the same volumetric primitive P as it may appear in a real image. 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. Hence, 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 lines, as in image C, the primitive P can still be recovered, although with more ambiguity. Three or four lines 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 lines may represent the generic type of the projection of the section, and the other two lines may correspond to the boundary of the projected area swept by the section. This simplification of the projection contours also has advantages. The problematic projection of image B can easily be approximated by three or four lines, and thus a relationship between image B and primitive P is established, via image C.

Our approach can be summarized as follow. First, the projection (or part) contours are simplified to correspond to one of a finite number of spatial lines 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 approach.

4 Simplification of 2D parts

As mentioned in the previous section, simplified parts are computed from the original 2D parts, to transform any possible spatial arrangements of lines into spatial arrangements of lines 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 lines. The optimization consists in defining a simplified part made of three or four arcs or segments (lines), that covers as accurately as possible the area of the original 2D part and that is as rectangular as possible (the angles between the lines of the simplified part must be as close as possible to 90°). This rectangularity criterion is based on the fact that during part segmentation, lines are grouped using symmetry and parallelism criteria. Therefore, the parts tend to be rectangular. The optimization can be expressed in the following way. Let $Pc_i = (\#, x, y)$ 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 $Pc = \{Pc_i\}$ be the set of all the Pc_i and $Pc_o = (Pc, \prec_\#)$ be the ordered set of the elements of Pc on #. Furthermore, let $\angle Pc_i$ be the angle between the vector defined by Pc_i and the preceding point, and the vector defined by Pc_i and the following point in an ordered set. Let *Area(PL)*, be the area enclosed by the closed cycle defined by linking the points of the ordered set *PL*. Next, let $Pso_i = \{Pc_a, Pc_b, Pc_c, Pc_d\}$ be an ordered set of four points from Pc_o . Finally, let $Ps = \{Pso_i\}$ be the set of all the possible Pso_i . The ordered set of points *Psf* making the boundary of the simplified parts is:

$$Psf = \min_{Pso_i \in Ps}\left(\sum_{Pc_i \in Pso_i} (\angle Pc_i - 90^\circ) + \left| Area(Pco) - Area(Pso_i) \right| \right) \quad \text{(EQ. 1)}$$

The simplified parts with three lines are obtained by removing one of two very close points in Psf . Then, a polygon is constructed from Psf . The contour of the polygon is compared with the original contour of the 2D part to replace segments by arcs whenever needed to obtain an accurate simplified part. Pairs of consecutive points of Psf (defining a segment) 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 an arc is found, an arc replaces the corresponding segment of the polygon.

Figure 5 shows simplified parts obtained by applying the optimization criteria on different parts from real objects.

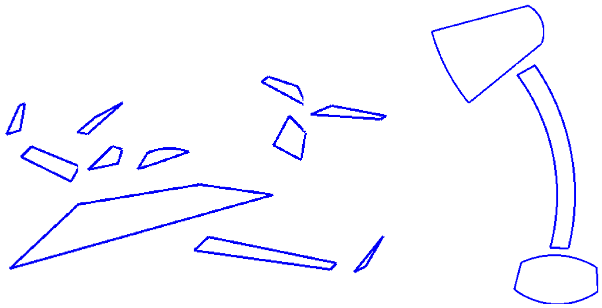

Figure 5: Simplified parts of the airplane and the desk lamp.

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 line arrangements. Since all simplified parts are composed of three or four lines, 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 three or four lines projected contour.

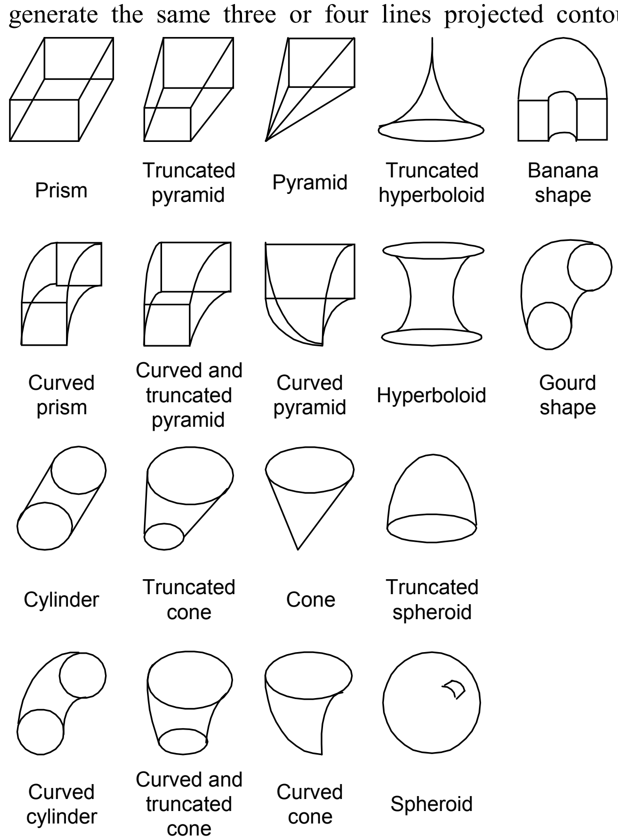

Figure 6: The eighteen volumetric primitives that can be hypothesized.

Each of these hypotheses corresponds a volumetric primitive and its projected axis.

5.1 Volumetric primitives

Eighteen simple volumetric primitives, which can be hypothesized from projection made of three or four lines, 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 [8]. Their projections generate all the possible three or four line 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.

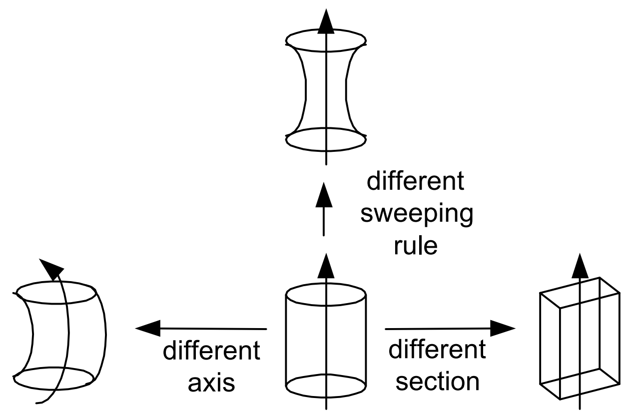

Figure 7: The parameters differentiating the volumetric primitives.

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 lines that compose it, the number of arcs among these, the level of parallelism of the segments, the level of compatibility of the arcs, the convexity of the arcs 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 lines: *IsNumberLines(x)*

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

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

5.2.2 Test on the number of arcs: *IsNumberArcs(x)*

This test verifies if the simplified part has the number of arcs specified as the argument. In this case, the result is fuzzy to account for cases where an arc with a weak curvature that describes the section has been converted into a segment. 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 arcs or two segments. Hence, when an arc is missing it is assumed that it is on the section. Therefore, if the argument is 1 or 3 arcs, the result is binary, but if the argument is 2 (or 4) arcs, the result is 1 when the part have 2 (4) arcs, a value between 0 and 1 when the part have 1 (3) arc (an arc may be missing), and 0 in the other cases.

The number of arcs a simplified part has is computed by scanning its boundary.

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

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

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

The level of parallelism for pairs of segments is computed by using concepts inspired of perceptual grouping theory. The level of parallelism is not linear with respect to the angle between the lines, and the length and the distance between the lines 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 lines, and Δs is the difference in length at both end of the axis of the virtual line swept along the axis to cover the area between the two lines. $K1$ is the 0.5 threshold of the log-sigmoid. It is a function of d and of the average distance between the lines. $K2$ 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 lines makes the level of parallelism vary more quickly as the length of the lines increase. These two values also ensure that the distance between the lines has the same effect.

The level of parallelism computations are applied to each pair of segments. 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 segments 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 arcs: *IsCompatible(x)*

The level of compatibility is a measure that establishes if two arcs sweep about the same portion of a circle. If two arcs 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 arcs is computed using the fact that in our implementation, each arc is defined clockwise. Hence, their start point can be used to determine the compatibility. If two arcs are linked by their start points, either directly or via another line on the contour of the simplified part, they are compatible. The binary results of the computations are sorted from compatible to incompatible. Arc pair #1 is the pair that gets the highest level of compatibility value, and arc 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 compatible arcs is convex or concave, or if all the arcs 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 arcs must be convex, or just a particular pair.

As for the compatibility, the convexity of a pair of arcs is computed using the fact that each arc 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 arcs have their start and end points in the same order as the sampled points. Results are computed for all possible pair of arcs, and for the region made of all the arcs of the simplified part.

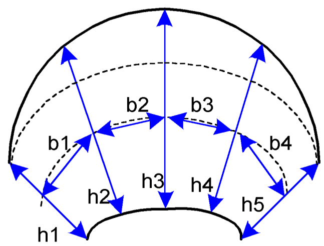

Figure 8: Values used for the determination of the sweeping rule.

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

This test is applied on compatible arc pairs. For some volumetric primitive hypotheses, the axis is defined by two compatible arcs. However, depending on the curvature of the two arcs, different hypothesis can be generated. For example, depending on the curvature of the two arcs, 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 arcs. The area between the two arcs (*P_Area*) is compared with the area made with the first arc and an arc with the same endpoints as the second arc, but with the curvature of the first arc (*C_Area*). For simplicity, the area between two arcs is computed approximately by the area of four trapezoids (see Figure 8). Hence, the two areas are computed as follow:

$$C\_Area = h_{\min} * (b_1 + b_2 + b_3 + b_4)$$

where $h_{\min} = \min(h_1, h_2, h_3, h_4)$, and (EQ. 3)

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

Three types of sweeping rules are possible for a pair of compatible arcs. That is, constant sweeping rule, growing sweeping rule and growing-shrinking sweeping rule. For a given pair of compatible arcs, 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 arcs 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 arcs 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 lines, 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 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 advantages some hypotheses without discarding the ones less likely. Advantaging some hypotheses allows the model-matching step to match the models that are more likely similar first.

All the rules uses the following formulation:

$$
\begin{aligned}
&if\,(TestA(x) \cap TestB(x) \cap \ldots) \\
&then \\
&X \text{ with } FMV = \min(TestA(x) \cap TestB(x) \cap \ldots) * beliefX \\
&\text{Y with } FMV = \min(TestA(x) \cap TestB(x) \cap \ldots) * beliefY
\end{aligned}
$$

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 lines (4), its number of arcs (2), the convexity of the arcs (convex) and the level of parallelism of its segments (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:

<u>Rule 8:</u>

$if\,(IsNumberLines(4) \cap IsNumberArcs(2) \cap IsParallel(1) \cap$

$IsConvex(ALLARCS))$

$then$

$Cylinder$ with $FMV = \min(IsNumberLines(4) \cap IsNumberArcs(2) \cap$

$IsParallel(1) \cap IsConvex(ALLARCS)) * 1$

$TruncatedSpheroid$ with $FMV = \min(IsNumberLines(4) \cap IsNumberArcs(2) \cap$

$IsParallel(1) \cap IsConvex(ALLARCS)) * 0.2$

<u>Rule 9:</u>

$if\,(IsNumberLines(4) \cap IsNumberArcs(2) \cap not(IsParallel(1)) \cap$

$IsConvex(ALLARCS))$

$then$

$Cone$ with $FMV = \min(IsNumberLines(4) \cap IsNumberArcs(2) \cap$

$not(IsParallel(1)) \cap IsConvex(ALLARCS)) * 1$

$TruncatedSpheroid$ with $FMV = \min(IsNumberLines(4) \cap IsNumberArcs(2) \cap$

$not(IsParallel(1)) \cap IsConvex(ALLARCS)) * 0.2$

where *not()* is the fuzzy negation operator. The numerical values of belief are given for illustrative purposes. These values will be determined more precisely later on during our work.

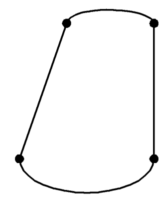

Figure 9: Simplified part for the example.

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. Hence, 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

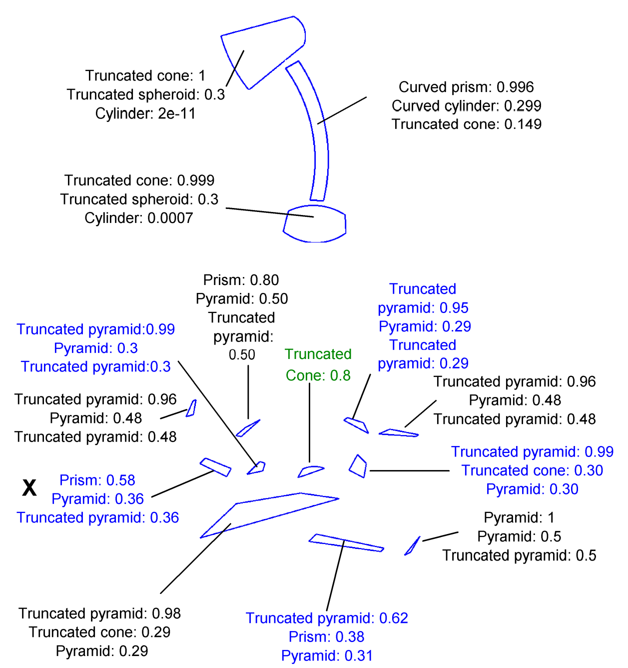

Figure 10: Volumetric primitive hypotheses for the simplified parts of the desk lamp and the airplane.

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.

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 have a significant impact on 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 approach for hypothesizing volumetric primitives from 2D projections. This new approach is hybrid, as it combines a model-fitting approach and a rule-based approach. 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 approach behaves in general as intended. The performance of the approach will be measured more accurately after the model matching algorithms are implemented.

Hence, 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] 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.

[2] 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.

[3] 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.

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

[5] 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.

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

[7] 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.

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