Hierarchical Decomposition of Objects in Line Drawings

Robert Bergevin Martin D. Levine

Computer Vision and Robotics Laboratory
McGill Research Center for Intelligent Machines
McGill University
Montréal, Québec, Canada

Abstract

We describe an algorithm for segmenting a 3-D object into its constituent volumetric parts from a single 2-D line drawing. This algorithm is used at an intermediate stage of a generic object recognition system, the *Primal Access Recognition of Visual Objects* (PARVO) system. The latter is based on *Recognition by Components* (RBC), a theory of human image understanding from the field of psychology. A short description of the RBC theory is presented first. This is followed by an overview of the main aspects of a specific implementation of our PARVO system. A preliminary stage to the segmentation consists of extracting viewpoint-invariant and non-accidental features from this line drawing. The segmentation process itself is described as a hierarchical decomposition of the line drawing on the basis of pairs of high-concavity points in its silhouette. The pairing of these so-called segmentation points rests on (i) an object visibility model describing the generic structures present in any line drawing satisfying our assumptions, and (ii) the feature instances extracted from the specific line drawing analyzed.

Résumé

Nous décrivons un algorithme de segmentation d'un objet tri-dimensionnel en ses parties volumétriques constituantes à partir d'un seul dessin au trait bi-dimensionnel. Cet algorithme est intégré à un système de reconnaissance générique d'objets appellé PARVO (pour *Primal Access Recognition of Visual Objects* ou *Reconnaissance visuelle d'objets par accès primitif*). Ce système est lui-même basé sur une théorie psychologique de la compréhension d'images appellée *Recognition by Components* (RBC). Nous présentons d'abord une courte description de cette théorie suivi d'un survol des principaux aspects de notre implémentation sous la forme du système PARVO. Une étape préliminaire à la segmentation consiste à extraire du dessin des caractéristiques géométriques non-accidentelles et invariantes par rapport au point de vue. Le processus de segmentation lui-même est présenté comme une décomposition hiérarchique du dessin sur la base des paires de concavités situées sur sa silhouette. Un modèle générique des structures visibles du dessin ainsi que les caractéristiques géométriques extraites pour un dessin particulier sont utilisés à cet effet.

1. Introduction

The *Primal Access Recognition of Visual Objects* (PARVO) system is a generic object recognition system. It is based on some ideas expressed in *Recognition by Components* (RBC), a theory of human image understanding developed by I. Biederman [1], a researcher in the field of perceptual psychology. This theory aims at integrating part of the knowledge obtained from psychological and computational vision research into a consistent model describing how objects are visually recognized by humans.

The recognition process modelled by the RBC theory differs significantly from most existing artificial object recognition systems [2, 3, 4, 5] because it addresses the problem of *primal access*, that is, "the first contact of a perceptual input from an isolated, unanticipated object to a representation in memory" [1] (p.32). In other words, RBC models visual recognition of generic objects in unconstrained environments in the sense that no context is used in the analysis of the visual information. The primal access task which is addressed theoretically by RBC and computationally by the PARVO system is very different from what is known as *model-based* recognition in the field of computer vision. In the latter, objects may only be recognized as a result of a detailed inference over a limited set of accurate models. Primal access, on the other hand, relies only on qualitative and coarse models of objects and parts. A typical example of a primal access task is the recognition, by humans, of objects appearing in rapidly displayed photographic slides.

There are five basic principles behind the RBC theory: (i) a *line drawing* is sufficient for the primal access task, (ii) objects are better represented and analysed by decomposing them into their natural *components* (parts), (iii) a *qualitative description* of the components is necessary and sufficient to permit *fast access* to a *large database* of object models, (iv) *non-accidental* instances of *viewpoint-invariant features* in the 2-D line drawing are sufficient to permit fast access to the qualitative model of a 3-D object, (v) primal access for visual object recognition is obtained by *matching* a description of the spatial structure of the object components to an *indexed* database of models in a similar representation.

In this paper, we shall describe the part segmentation module of the PARVO system. The paper begins with a short summary of the RBC theory as reported in [1]. This is followed by an overview of the computational aspects of the PARVO system. Four main aspects are identified. They are: (i) *extraction of features* from a single view line drawing of an object, (ii) *segmentation* of this line drawing in conformity with the part connection structure of the visible object, (iii) *determination* of the *primitive volumetric shapes* each part is an instance of, and (iv) *identification* of the visible object as being one of the many known objects. The emphasis of this paper is on the second aspect. Details of the other aspects will be provided in forthcoming papers.

2. The RBC theory and the PARVO system

2.1 Processing stages in the RBC theory

According to the RBC theory, a primal access object recognition system must analyze a single view intensity image of an object to produce a qualitative description of its shape. The latter is computed in terms of volumetric primitives in a representation of shape suggested in [1]. This representation is based on the decomposition of an object into a set of simple parts which are described as instances of generic shapes from a set of only thirty-six volumetric primitives (the so-called *geons*). The geons are closely related to the *generalized cylinders* representation [5]. Figure 1 shows the four attributes by which a geon is represented and the few symbolic values each of these attributes can possibly assume. As can be seen, the description of any part of an object will be very coarse and qualitative. A model of a complete object consists of a *spatial structure* of parts that are instances of such geon primitives. This spatial structure is basically defined by the physical connections which exist between the parts of the object. The RBC theory assumes that parts are joined in one of two generic connection *modes*: (i) *end-to-end* connection mode, and (ii) *end-to-side* connection mode.

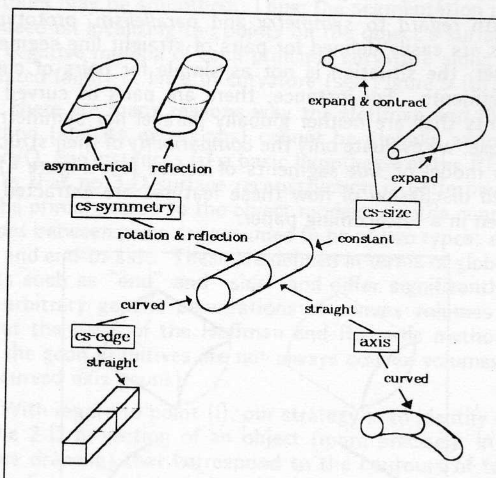

Figure 1 Geon representation for primitive volumetric parts. This figure, adapted from Figure 5 in [1], shows how the shape of the geons varies according to the symbolic values taken by the four attributes. These four attributes appear at the nodes of the graph. They are: (i) the type of edges forming the cross-section (CS-EDGE), (ii) the expansion function of the cross-section (CS-SIZE), (iii) the type of symmetry of the cross-section (CS-SYMM), and (iv) the type of axis (AXIS). The symbolic values they can take are represented by labels on the directed links to the prototypical geon shapes.

The processing modules of a hypothetical system satisfying the principles of the RBC theory are (see Figure 2): (i) an *edge extraction* module which computes a *line drawing* from an intensity image, (ii) a *nonaccidental properties detection* module which extracts line segment *features* from the line drawing [1], (iii) a *parsing* (part segmentation) module which decomposes the line drawing primarily at concave regions, (iv) a *component determination* module which infers the *geon* class of each visible part of the object, and (v) a *matching* module which compares the description obtained in the geon-based representation to symbolic descriptions of models of known objects. As a result, the system is able to *identify* the object appearing in the scene as being either an instance of one of the known generic object models or a new (previously unknown) object.

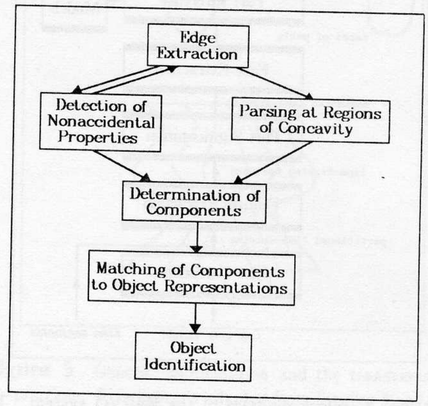

Figure 2 The processing stages in the RBC theory. This figure, adapted from Figure 3 in [1] presents the presumed processing stages in the primal access task of visual object recognition according to the RBC theory. The modules represent only the main aspects of the information processing system. The global structure shows the mainly bottom-up approach of the theory (where the "bottom" and "top" are upside-down in the figure).

2.2 Processing stages in the PARVO system

Figure 3 is a block diagram summarizing the processing modules of the PARVO system [7]. We note many differences between its structure and the original RBC block diagram shown in Figure 2. These changes result from our computational refinement of the RBC theory. Also, the first processing stage, which should produce a *partitioned line drawing* from an *intensity image* of a single-object scene, is added only for completeness. The existing implementation of the PARVO system uses a partitioned line drawing made up of *simple* line segments as input. More details on this aspect are given below.

The remainder of the paper provides more details concerning the algorithm used in the part segmentation module of PARVO. A detailed description of the other modules and experimental results will appear elsewhere. An early account of some other aspects of PARVO is also available in [7].

3. Extraction of the parts in the line drawing

3.1 The single view line drawing

The line drawings analyzed by PARVO are determined only by the shape of the objects in view. We refer to them as *geometrical* line drawings. Other types of line drawings

[1] The five classes of features suggested by Biederman are instances of the viewpoint-invariant and non-accidental features introduced by Lowe [6].

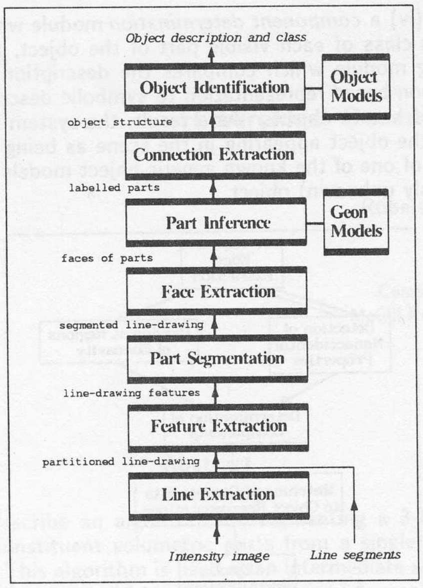

Figure 3 Block diagram of the PARVO system. The processing is mainly bottom-up in the abstraction hierarchy of representations. The present implementation of the *PARVO* system requires the user to create a partitioned line drawing interactively using a graphics interface.

are *optical, perceptual*, and *artistic* whose illumination, textural, illusory, and aesthetic effects we have considered to be extraneous to the primal access task. To be acceptable to PARVO, line drawings must satisfy a certain number of assumptions. Basically, *objects* are assumed to be the result of *connecting* idealized simple volumes, the *parts*. *Line drawings* are made up of contours of depth and surface normal (orientation) discontinuities of these objects, as seen from a particular *viewpoint*. There are four classes of assumptions and these will be discussed in detail in a later paper. The first three are *geometrical* in scope. These are: (i) assumptions concerning the *shape* of the parts making up the object, (ii) assumptions concerning the structure of the *connections* between these parts, and (iii) assumptions concerning the spatial characteristics of the *viewpoint* from which a given object gives rise to an acceptable line drawing. The fourth class of assumptions concerns the relevant *physical* properties of the scene.

These four classes of assumptions lead to idealized line drawings depicting the surface shape (geometrical) discontinuities of an object in view. Line drawings of objects satisfying these assumptions can be considered to be within the scope of (or acceptable to) PARVO. Hence, the analysis performed by the system will be based on the structures that can arise in such acceptable line drawings. We have defined these structures in terms of a qualitative *object visibility model*. There are two aspects to this model: (i) the *part visibility model* describes the structure of the line drawing of a part isolated as a result of the segmentation process (see Figure 4), and (ii) the *connection visibility model* describes the structure of the line drawing in the neighbourhood of the *join* where two parts are connected. This structure is defined in terms of certain features extracted from the line drawing (see below).

3.2 Extraction of viewpoint-invariant features

Part segmentation of a line drawing is preceded by a stage in which viewpoint-invariant and non-accidental features are extracted (Figure 3). The first step of this feature-extraction process is a decomposition (partition) of the line drawing into a set of *straight* and *curved line segments*. This decomposition is based on the *inflection points* and the *tangent* and *curvature* discontinuities in the line drawing. As mentioned above, this partition is presently provided interactively. Furthermore, the *simple* line segments resulting from this process are approximated by straight line segments and arcs of circles, and constitute the first two classes of features.

The remaining three classes of features suggested by Biederman are *parallelism, symmetry*, and *cotermination*. The last one does not present any particular difficulty since it is simply a local property related to the distance between endpoints of different line segments. More precisely, PARVO extracts pairs of line segments which have close endpoints (*corners*) and groups them into higher order junctions such as *arrow, fork, peak*, and *tree*. In addition to these *cotermination* junctions, *occlusion* junctions are also extracted. That is, *T-junctions* are first obtained from the partitioned line drawing and are then grouped to form the higher order junctions, *matched-T (mt)* and *pi*-junctions.

With regard to *symmetry* and *parallelism*, prototypical models are easily defined for pairs of straight line segments. However, the situation is not as simple for pairs of curved line segments. For instance, there are pairs of curved line segments that are neither globally parallel nor symmetrical. For these, we evaluate only the *compatibility* of their structure with a model of *side* segments of a *face* (see Figure 4). A detailed discussion of how these features are extracted will be given in a forthcoming paper.

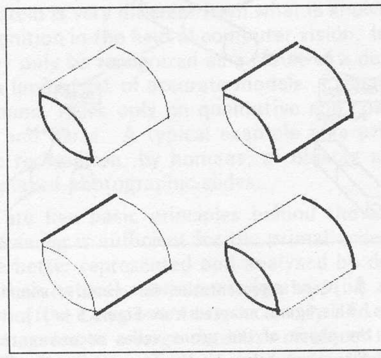

Figure 4 Part visibility model. The system's *object visibility model* is comprised of a *part visibility model* and a *connection visibility model*. Only the former is presented here. Thus, any visible part is made up of *cross-section* and *body* faces (top left and right, respectively). Also, the line segments bounding these faces are either *side* or *end* segments (bottom left and right, respectively).

3.3 Segmentation of an object into its constituent parts

The RBC theory approach to the part decomposition problem is based on a method suggested by Hoffman and Richards [8]. According to these authors, a general-purpose visual object recognition system must (i) separately consider the two problems of segmenting an object into its constituent parts and describing these parts in a given representation, and (ii) rely on constraints provided by regularities of nature for segmentation.

From point (i), it is concluded that segmentation must be founded on a so-called *boundary-based* approach rather than on the more common *primitive-based* approaches. In the latter, specific assumptions are made regarding the shape of parts (see for instance [4]) and they determine the particular part decomposition obtained. With regard to point (ii) above, Hoffman and Richards have suggested that the *transversality* principle of differential topology [9] provides a powerful natural constraint for segmentation. This principle states that, with probability one, two arbitrarily intersecting surfaces create a spatial contour made up of points of concave discontinuities of their tangent planes. More specifically, when a convex volume penetrates another convex volume a contour made up of points at which there is a concave discontinuity of the tangent planes of their surfaces is created. Any inter-penetration of two convex volumes for which this contour is closed in space is said to be *generic*. Thus, their part segmentation method consists of decomposing an object into convex volumetric parts along the closed discontinuity contours on its surface [2] (see Figure 5 (a)). For some objects, the discontinuous surface in the neighbourhood of the contours may be smoothed. Thus, the segmentation method is based on localizing the points on the object's surface that are negative minima of each principal curvature along its associated family of lines of curvature (see Figure 5 (b)).

There are many reasons why the Hoffman and Richards method (and its extensions) cannot be globally adopted by PARVO. For instance: (i) a basic hypothesis of the RBC theory is that no 3-D surface reconstruction is performed prior to the primal access to the object models, (ii) the connection *modes* between parts are assumed to be of two types: end-to-end and end-to-side. These are defined in terms of global concepts such as "end" and "side" and differ significantly from the arbitrary generic penetrations of convex volumes which are at the basis of the Hoffman and Richards method, and (iii) the geon primitives are not always convex volumes (e.g., the curved axis geons).

With regard to point (i), our strategy is to identify events in the 2-D projection of an object (more precisely, in its 2-D line drawing) that correspond to the contours of tangent plane discontinuities on the object's surface. Clearly, these events must account for all possible part connections in line drawings of objects satisfying our assumptions. This criterion leads to the *connection visibility model* which is defined in terms of the extracted features under the assumption that all connections are of simple generic types (point (ii) above). This model makes explicit the different possible structures in the line drawing that result from any part connection, including those involving the curved axis geons mentioned in point (iii). The details of these aspects of PARVO's part segmentation method are provided next.

3.4 Outline-based part segmentation in the PARVO system

As shown in the block diagram of PARVO (Figure 3), the part segmentation process is followed by the face extraction and part inference (labelling) stages. Thus, part segmentation must produce different subsets of the line drawing, each to be interpreted in the same uniform way. Intuitively, the part segmentation of an object should be as shown in Figure 6. A formal definition of such a part segmented line drawing could be as follows. A segmented line drawing consists of the different subsets of simple line segments from the partitioned line drawing that belong to the different parts of the object. That is, these subsets of line segments make explicit the part decomposition of the object. Let $L = \{l_i \mid 1 \leq i \leq N\}$ be the set of line segments making up the line drawing, where N represents the number of these line segments. The part segmentation process produces M sets of line segments P_j with $L = \bigcup_{j=1}^{M} P_j$; that is, each of these sets of line segments satisfies $P_j \cap L = P_j$ (where $\mid P_j \mid = n_j$, with $n_j \in [1, N]$) and corresponds to a part of the object. Some line segments in the line drawing may be shared by two adjacent and connected parts. Thus for any two sets of line segments P_i and P_j corresponding to two different parts of the object, we have $P_i \cap P_j \subseteq L$. Of course, the segmented line drawing contains only the visible line segments for each of the parts.

The previous definition of a segmented line drawing pertains to our understanding of the goal of the segmentation process. It remains to relate this top-level description of a segmented line drawing to an algorithmic definition of the segmentation process. Let us again consider the line drawing in Figure 6. This is a simple and illustrative example of a line drawing analysed by PARVO. Figure 7 presents this line

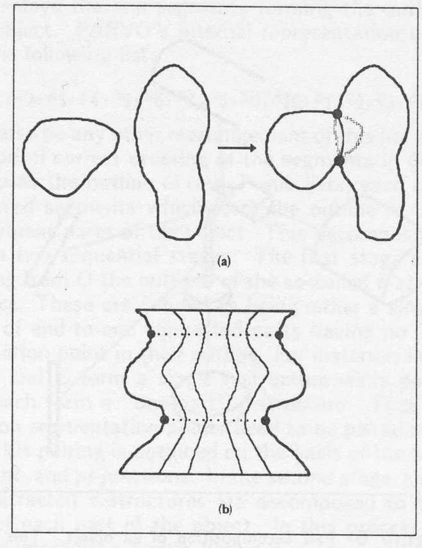

Figure 5 Generic segmentation and the transversality principle. This figure is adapted from Figures 3 and 8 in [8]. The transversality principle (a) says that when two volumes interpenetrate at random (in a region where they are convex), they generically meet at a contour of concave discontinuity of their surfaces' tangent planes. In (b) is shown a side view of a solid of revolution. This object may be described as being made up of three *smoothly* connected parts. The families of solid lines shown on its surface are the lines of curvature whose minima give rise to the dotted segmentation contours. The latter are closed 3-dimensional contours (circles for solids of revolution). In the 2-D projection of an object, the contours can be detected from the concave discontinuity points highlighted in (a) or from the points of local maxima of negative curvature shown in (b).

[2] Beusmans *et al.* [10] have extended the Hoffman and Richards method by also permitting nongeneric penetrations and subtractions of convex parts. Thus, their segmentation method is able to deal with "elbow" and "dent" structures. However, their method is more complex since it relies on many segmentation "rules" that must be applied in a strict sequence in order to provide a unique segmentation of a given object.

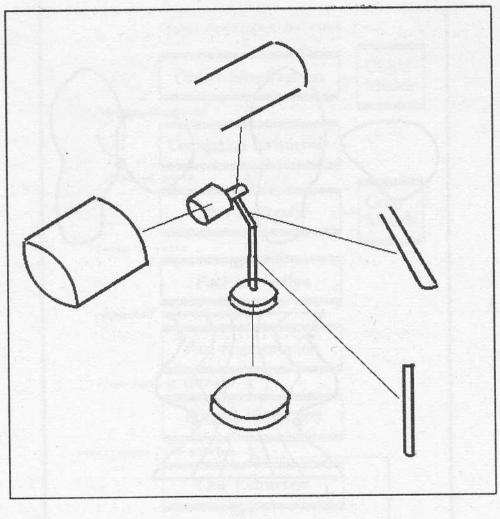

Figure 6 Part decomposition of an object. This exploded view of the object's parts shows what is meant by a part decomposition of an object. Each resulting part must be labelled as one of the geon shapes.

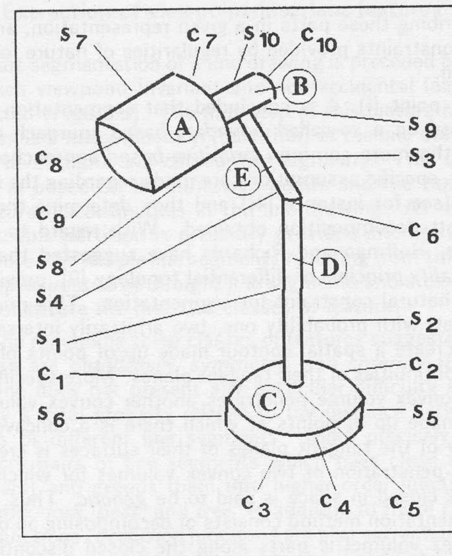

Figure 7 A simple line drawing analyzed by the PARVO system. There are 10 straight line segments ($s_i, i = [1, 10]$) and 10 curved line segments ($c_j, j = [1, 10]$) in this line drawing. The object has five parts, all of which are visible in this view.

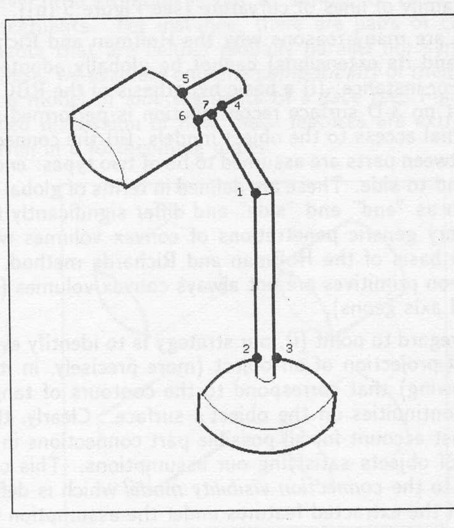

Figure 8 Silhouette and segmentation points. The silhouette (or outline) which is shown here by heavy lines is represented in the *PARVO* system as an unordered list of line segments. The filled-in circles in the figure indicate the segmentation points.

drawing with the straight and curved line segments labelled in an arbitrary order. Basically, part segmentation should occur at concave tangent discontinuities in the *outline* (or silhouette) [3] of this object (see Figure 8) [1]. These discontinuities in the line drawing outline are the 2-D events which correspond to the closed contours of tangent plane discontinuities on the surface of an object. The points at which they occur are called the *segmentation* points. The latter are situated at high-concavity *corners* and *T-junction* features (see points 1 and 7, respectively, in Figure 8) in the outline of an object and result from the connection of geon-like parts in any one of the two generic *connection modes* introduced earlier.

We have already observed that a part connection produces a contour of tangent discontinuities on the surface of an object [4]. In the 2-D line drawing corresponding to a given viewpoint either a pair of corner segmentation points (e.g., see point 1 in Figure 8) or a pair of T-junction segmentation points (e.g., points 4 and 7 in Figure 8) will appear for each discontinuity contour on the object's surface. Thus, segmentation of the line drawing into parts is only possible after the segmentation points have been identified and paired.

The role of the connection visibility model is precisely to determine the criteria for pairing the segmentation points. These criteria can be summarized as follows: two segmentation points may be paired if and only if they both belong to a common *join* in the line drawing, at the connection between two parts. The evidence for these joins consists of higher order junctions in the line drawing. The *mt-* and *pi-*junctions are strong evidence of such joins for many part connections. In effect, most part connections give rise to occlusions in a given view and hence to higher order *mt-* and *pi-*junctions (see Section 3.2). Thus, two T-junction segmentation points are paired only when their segments form such a higher order junction. For instance, points 4 and 7 in Figure 8 belong to the T-junctions made up of segments (s_9, s_4) and (s_9, s_3), respectively. In turn, these two T-junctions are grouped into a *pi-*junction. Hence, points 4 and 7 may be paired.

A more difficult case is the end-to-end connection between two parts having cross-sections of the same size and shape at their join. These should give rise to two corner segmentation points in the line drawing (e.g., see parts D and E in Figure 7) [5]. There is no junction type defined such

[3] A *directed* line segment is a segment with a specified endpoint. We define a directed line segment on the outline as one for which the object is on the left when going from the specified endpoint to the other endpoint. In the present implementation, the directed line segments on the outline are interactively identified. This amounts to an implicit figure/ground decomposition of the scene.

[4] Note that we have made the assumption that no smoothing occurs at part connections for the objects analyzed by PARVO.

[5] Of course, two connected cylinders having identical cross-sections

that pairing of these two corner segmentation points can be obtained from it. The only solution is to verify that the two segments meeting at the first corner segmentation point form either a *parallel*, *symmetrical*, or *compatible* feature pair with the two segments at the second corner segmentation point. This approach is justified by the fact that these features are extracted from a pair of line segments satisfying a model of the *side* segments of a part (see Figure 4). Clearly, this verification is equivalent to an implicit extraction of a structure made up of a pair of higher order junctions (e.g. two *fork* junctions).

One last remark is necessary in order to understand our part segmentation algorithm. Namely, as as result of our assumption concerning the connection modes of acceptable objects, it would seem that any line drawing analyzed by PARVO could be described as an object made up of a *hierarchical structure* of parts. Actually many objects, such as the one in Figure 7, have such a hierarchical structure and may be recursively decomposed into parts. However, our assumptions do not reject objects having some of their parts connected in a loop (that is, the last part in the loop is connected back to the first one). Such objects will possess holes and a silhouette made up of more than one connected chain of line segments. A simple example of such an object is a cup which has an exterior and an interior outline (formed by the handle). Clearly, such objects cannot be decomposed as easily as the one in Figure 7. The segmentation algorithm we describe is a general one that can deal with both types of objects. We use the object in Figure 7 as a simple and illustrative example. The part decomposition of a more complex object is presented in Figure 9.

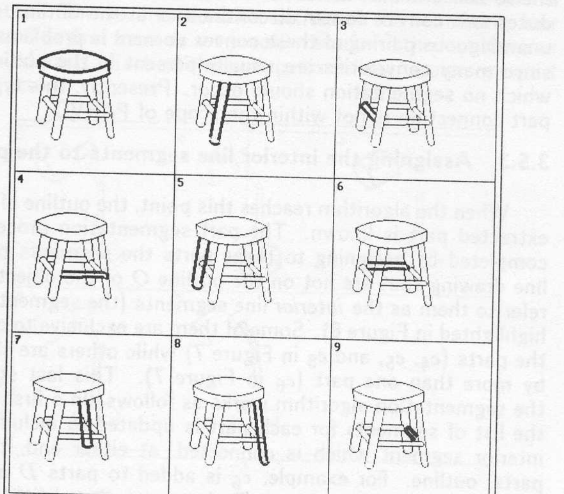

Figure 9 Part decomposition of a complex object. This object contains loops in its part connections. It is more complex than the hierarchical objects and cannot be segmented by a simple recursive process. The actual decomposition sequence obtained by the *PARVO* system is from left to right and top to bottom.

3.5 Part decomposition of a simple hierarchical object

As mentioned above, the first step of the segmentation algorithm consists of decomposing the outline into parts. Figure 8 displays the line segments forming the outline of our simple object. PARVO's internal representation of the outline is the following list:

$O = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, c_1, c_2, c_3, c_7, c_9, c_{10}\}$.

It could also be any other rearrangement of this list since there is no a priori correct ordering of the segments in O. PARVO decomposes the outline O into P sub-lists, each comprising the *ordered* segments which form the outline of one of the P constituent parts of the object. This decomposition is obtained in two sequential stages. The first stage consists of extracting from O the outlines of the so-called ***τ-structures*** of the object. These are defined as being either a single part or a series of end-to-end connected parts having no T-junction segmentation point in their outline. For instance, in Figure 7, parts D and E form a single τ-structure while parts A, B, and C each form a "one-part" τ-structure. Thus, only the T-junction segmentation points need to be paired in this first stage. This pairing is obtained on the basis of the already extracted *mt*- and *pi*-junctions. In the second stage, the outlines of the extracted τ-structures are decomposed to obtain the outline of each part of the object. In this process, only the corner segmentation points within a given τ-structure need to be paired. This second pairing is obtained on the basis of an (implicit) structure of higher order junctions (see Section 3.4). In general, a τ-structure made up of P_s parts will contain $2P_s - 2$ such corner segmentation points since each part connection contributes a unique pair according to our assumptions. The following sections describe these two stages.

3.5.1 Sequential extraction of τ-structures

At each *mt*- or *pi*-junction, two parts are assumed to be connected. One part is *occluding* while the other is *occluded*. Moreover, each of these two parts belongs to a different τ-structure. We refer to the *mt*- and *pi*-junctions as the *seeds* from which the different τ-structures are extracted from the line drawing. Initially, it might be possible to extract from these seeds the outline of either the occluding τ-structure, the occluded one, both structures, or none of them. Intuitively, a τ-structure is extractable if and only if its outline does not contain any gap. More formally, let us define a *connected path* to be an (ordered) list of line segments $O_p = \{l_1, l_2, \ldots, l_n\}$ from O ($l_i \in O$ for $i \in [1, n]$) in which each pair of consecutive segments ((l_j, l_{j+1}) for $j \in [1, n-1]$) makes a *corner* feature in the line drawing. Moreover, a *closed* connected path is defined to be a connected path for which the last segment makes a corner with the first one. Thus, an occluding τ-structure is extractable if and only if its outline forms a closed connected path. Similarly, an occluded τ-structure is extractable if and only if its outline forms a (simple) connected path from the first T-junction branch to the second T-junction branch of the seed [6]. In both cases, we say that the outline of the τ-structure forms an acceptable *c-path*.

If we examine the line drawing in Figure 7, we see that it has three c-paths in its initial outline. Two of them are at the *pi*-junction seed situated at the join between part A and part B. Namely, the outline of the occluding one-part τ-structure, $O_A = \{c_7, s_7, c_9, s_8\}$, forms a (closed) c-path while the outline of the occluded one-part τ-structure, $O_B = \{s_9, c_{10}, s_{10}\}$, forms a (simple) c-path. The third c-path belongs to the occluded one-part τ-structure of the *mt*-junction seed situated at the join between part C and part D. Its outline, $O_C = \{c_1, s_6, c_3, s_5, c_2\}$, forms a (simple) c-path.

---

and aligned axes do not generate any corner discontinuity in their outline. Thus, they are considered as a single cylindrical part by the system.

6 The outlines are always traversed in such a way that the object is kept to the left of the direction of motion.

The search for extractable τ-structures is as follows. Initially, the c-path list is comprised of only the first segment in O. Next, a list of all seeds containing this segment is created and each seed is used as a basis of a search for an acceptable c-path. If the first segment is a branch of the seed (it belongs to the occluded part), the (simple) c-path must reach the other branch of the seed. For instance, s_9 in Figure 7 is a branch of the *pi*-junction seed at the join between part A and part B. Thus, a c-path from s_9 to s_{10}, the second branch of the seed, is acceptable. The outline O_B (see above) is such a c-path. On the other hand, if the first segment is a top of the seed (it belongs to the occluding part) a closed c-path is needed. The outline O_A starting with c_7 is such a c-path.

If the list of seeds is empty, or no acceptable c-path is found after all seeds have been tested, the c-path list is re-initialized with the next segment in O. Then, a new list of seeds is created for this segment and the search is resumed. When an acceptable c-path is found, the line-segments forming this c-path are removed from O to a τ-structure list. This is followed by an updating (restoration) of O in order (i) to close the paths at places in the object where a *pi*-junction has been eliminated by removing this τ-structure, and (ii) to be able to extract the remaining structures whose outlines did not previously form c-paths. This is achieved by first applying two types of general restoration. The first one creates a *virtual segment* connecting the two branches of any *pi*-junction whose top segment belongs to the extracted structure. For instance, the extraction of the structure which contains part A would be followed by the creation of a virtual segment connecting s_9 and s_{10} since they both belong to a *pi*-junction whose top segment (c_7) belongs to the outline of part A. After, these virtual segments can participate in a c-path in the same way as the actual existing segments. However, they are not members of the final list of segments describing the extracted τ-structure.

The second type of general restoration is the completion of any part occluded by the extracted τ-structure. This occlusion may result either from a connection or simply from a particular viewpoint. In the first case, the two branches of an *mt*-junction are merged (e.g. if part D were extracted before part C). In the second case, two merged lines are formed from the collinear or cocircular branch segments of matched *pi*-junctions. There is no such occlusion in the simple object in Figure 7. For the more complex object in Figure 9, we see that the two sides of the sixth part need to be merged after the second part is extracted. Note that the branch segments of the *pi*-junctions on both sides of the second part are collinear. This is considered a sufficient condition to merge them and form the sides of what is to become the sixth part. In the case of an *mt*-junction seed, specific restorations are applied in addition to the above two general ones. First of all, the collinear or cocircular segments of the occluded part are merged to close its outline (segments c_1 and c_2 are merged as shown in Figure 10). Next, a *virtual segment* connecting the branches (segments s_1 and s_2) of the *mt*-junction is created.

After a τ-structure has been extracted and the outline restored, the algorithm is restarted with the updated O. However, the number of seeds and the extractability of the occluding and occluded structures at each of them might be different from what they were before. It is even possible that, after a certain number of structures have been extracted, either no seeds remain or no seed's τ-structure is extractable. In this eventuality, PARVO searches directly for (closed) c-paths in O. We refer to the τ-structures extracted by such a process as *isolated* structures since they have no attached seed.

In Figure 7, the outline of each one-part τ-structure forms a c-path. Thus, a search starting from any one of segments c_7, s_9, s_{10}, c_1, and c_2 would lead to an acceptable c-path. Given the order of the segments in O (see above), it can be predicted that the first acceptable c-path will arise from s_9. That is, B will be the first part extracted.

3.5.2 Decomposition of the τ-structures into parts

After all the outlines of the τ-structures have been extracted, these must be further decomposed into the outlines of their constituent parts (one or more). This second stage of the outline decomposition algorithm is similar to the first one. The only difference is that, this time, pairs of *corner* segmentation points within each τ-structure are used as seeds rather than the *mt*- and *pi*-junctions, as before. However, this stage will work only when the number of corner segmentation points in a given τ-structure is even and each is correctly paired. In practice, some of the end-to-end connections involving parts having equal cross-section sizes at their join may produce either a single concavity or even no concavity at all. In the former situation, there is an isolated concave corner segmentation point that cannot be paired. Fortunately, these *concave-convex* end-to-end connections can be dealt with using a straightforward generalization of the method. Namely, a concave corner segmentation point that cannot be paired with any other such point is paired with a convex corner in the outline. This is exactly what happens in our example of Figure 7 since the connection point between segments s_2 and s_3 is paired with the unique (concave) corner segmentation point at the common end of segments s_1 and s_4. Thus, the only τ-structure having more than one part is decomposed into the outlines of parts D and E. A more difficult case would be the so-called *double-convex* end-to-end connection which produces two convex corner discontinuities in the outline. The unambiguous pairing of these convex corners is problematical since many convexities are usually present in the outline at which no segmentation should occur. Presently, this type of part connection is not within the scope of PARVO.

3.5.3 Assigning the interior line segments to the parts

When the algorithm reaches this point, the outline of each extracted part is known. The part segmentation process is completed by assigning to these parts the segments of the line drawing that are not on the outline O of the object. We refer to them as the *interior* line segments (the segments not highlighted in Figure 8). Some of them are exclusive to one of the parts (c_4, c_5, and c_8 in Figure 7) while others are shared by more than one part (c_6 in Figure 7). This last step of the segmentation algorithm works as follows. In a first pass, the list of segments for each part is updated to include any interior segment which is connected, at either end, to the parts' outline. For example, c_6 is added to parts D and E but c_5 is only added to part D. In a second and final pass, each part adds to its updated list of segments any remaining interior segment which is connected to it at both ends.

3.6 Part decomposition of more complex objects

In the previous sections, we have described the different steps that PARVO must take in order to decompose an object into its parts. First of all, τ-structures are extracted from *mt*- and *pi*-junction seeds in the outline of the object. Then, isolated τ-structures are searched for in the remaining outline and extracted from it. Finally, each extracted τ-structure is decomposed into parts, using the concave corners in its outline, and the proper interior segments are assigned to each resulting part.

For simple (hierarchical) objects, such as the one in Figure 7, an application of these steps will produce a complete and correct segmentation. However, the sequence by which the parts are extracted is dependent on how the seeds are ordered and searched for extractable τ-structures. We have seen that this ordering is based on the segments in O. Thus, a rearrangement of these segments could change the extraction sequence. However, PARVO must also process complex objects such as the one in Figure 9 whose part connections form many loops. This might require more than one application of the previous steps in order to guarantee that the part decomposition is complete and correct. Therefore, the following algorithm is used for all objects. First of all, PARVO extracts, in turn, all possible parts from *pi*-junction seeds. Next, it obtains all possible parts from *mt*-junction seeds. Finally, the isolated parts are extracted. After, the previous sequence is iterated until either O is empty or no part is extracted at a given iteration. Thus, it is guaranteed that the algorithm will not stop until either all parts are extracted or O contains no c-path. The latter implies that the outline is incomplete. Using this algorithm, the parts of the object in Figure 7 were extracted in the sequence shown in Figure 10. As expected, only one iteration was necessary. The first two τ-structures are extracted from a *pi*- and an mt-junction seed, respectively. The third τ-structure is extracted without any seed since it is isolated in the updated outline. All three of these τ-structures consist of one part. Clearly, the last τ-structure extracted is always isolated. The one in our example is decomposed into parts D and E as described above. This algorithm also produces a complete and correct part decomposition for more complex objects. The one in Figure 9 necessitates two iterations of the algorithm. In the first iteration, five parts are extracted. The first two are from a *pi*-junction seed, the third and fourth are from *mt*-junction seeds, and the last one is isolated. The second iteration extracted the remaining four parts. The first from an *mt*-junction seed and the last three as isolated parts. Each of the nine parts forms its own one-part τ-structure.

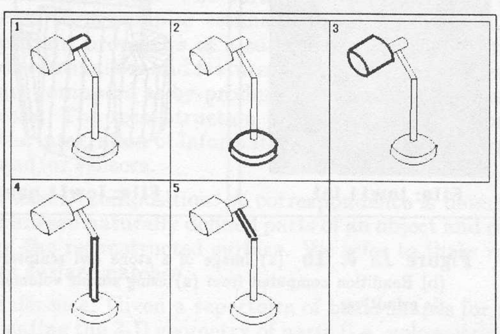

Figure 10 Part decomposition sequence for a simple object. This is obtained by applying the *PARVO* segmentation algorithm. Note the restoration of the second part.

Interestingly, this complex line drawing violates the assumption that any τ-structure connection produces either a completely invisible join in the line drawing (giving rise to a *pi*-junction) or a completely visible join (giving rise to an *mt*-junction). For instance, the left end of the sixth part in Figure 9 is occluded both by the fifth part, to which it is connected, and by the third part. Thus, the two side segments of the sixth part are actually occluded by two different line segments. These four segments do not form a *pi*-junction and, in accordance with the previous assumption, should not provide evidence for the segmentation to proceed. The solution, given such *double occlusions*, is to create a virtual segment connecting the two sides of the occluded part (the sixth part in Figure 9). No seed is created. Thus, the extraction of the sixth part is only possible using the *mt*-junction seed at its other end. However, the added virtual segment is necessary to complete the c-path of its outline. The ninth part in Figure 9 is similarly occluded by both the eighth and the fourth part. Its extraction is possible only after all the other parts are removed.

After part segmentation is completed, it is possible for PARVO to obtain a description of each part of the object in parallel. The description of the parts resulting from the segmentation process will be addressed in a forthcoming paper.

4. Summary

In this paper, we have presented an algorithm for segmenting a line drawing satisfying an object visibility model. The algorithm consists of hierarchically decomposing the line drawing of an object into its constituent parts by pairing segmentation points on the silhouette of the object. Results have been presented that demonstrate its capability to deal with single-view line drawings of complex objects. The context in which this algorithm is used is the *Primal Access Recognition of Visual Objects* (PARVO) system, a computer object recognition system based on the *Recognition by Components* (RBC) theory. Most aspects of the existing implementation of PARVO have only been outlined. A detailed description of each module and our experimental results will be published later. An early account of some aspects of PARVO is also available in [7].

Acknowledgements

The authors would like to thank I. Biederman for helpful discussions on his human image understanding theory. This research was partially supported by an NSERC Postgraduate Scholarship to R. Bergevin and the NSERC Operation Grant A1733 to M.D. Levine. M.D. Levine would like to thank the Canadian Institute for Advanced Research for its support.

References

[1] I. Biederman, "Human image understanding: recent research and a theory," *Comput. Vision, Graphics and Image Processing*, vol. 32, pp. 29-73, 1985.

[2] P. J. Besl, and R. C. Jain, "Three-Dimensional Object Recognition," *Computing Surveys*, vol. 17, no. 1, pp. 75-145, March 1985.

[3] D. G. Lowe, "Three Dimensional Object Recognition from Single Two Dimensional Images," *Artif. Intell.*, vol. 31, pp. 355-395, 1987.

[4] A. Pentland, " Recognition by Parts," Technical Note No. 406, Stanford, CA: Artificial Intelligence Center, SRI International, December 16, 1986.

[5] R. Nevatia, and T. O. Binford, "Description and recognition of curved objects," *Artif. Intell.*, vol. 8, pp. 77-98, 1977.

[6] D. G. Lowe, " Perceptual organization and visual recognition," STAN-CS-84-1020, Stanford, CA: Department of Computer Science, Stanford University, September 1984.

[7] R. Bergevin, and M. D. Levine, " Recognition of 3-D Objects in 2-D Line Drawings: An Approach Based on Geons," TR-CIM-88-24, Montréal, Québec, Canada: McGill Research Center for Intelligent Machines, November 1988.

[8] D. Hoffman, and W. Richards, "Parts of recognition," *Cognition*, vol. 18, pp. 65-96, 1984.

[9] V. Guillemin, and A. Pollack, *Differential Topology*. Englewood Cliffs, N.J.: Prentice-Hall, 1974.

[10] J. M. Beusmans, D. D. Hoffman, and B. M. Bennett, "Description of solid shape and its inference from occluding contours," *J. of the Optical Soc. of America (A)*, vol. 4, pp. 1155-1167, July 1987.

and correct segmentation. However, the sequence by which the parts are extracted is dependent on how the seeds are ordered and searched for extractable τ-structures. We have seen that this ordering is based on the segments in O. Thus, a rearrangement of these segments could change the extraction sequence. However, PARVO must also process complex objects such as the one in Figure 9 whose part connections form many loops. This might require more than one application of the previous steps in order to guarantee that the part decomposition is complete and correct. Therefore, the following algorithm is used for all objects. First of all, PARVO extracts, in turn, all possible parts from *pi*-junction seeds. Next, it obtains all possible parts from *mt*-junction seeds. Finally, the isolated parts are extracted. After, the previous sequence is iterated until either O is empty or no part is extracted at a given iteration. Thus, it is guaranteed that the algorithm will not stop until either all parts are extracted or O contains no c-path. The latter implies that the outline is incomplete. Using this algorithm, the parts of the object in Figure 7 were extracted in the sequence shown in Figure 10. As expected, only one iteration was necessary. The first two τ-structures are extracted from a *pi*- and an mt-junction seed, respectively. The third τ-structure is extracted without any seed since it is isolated in the updated outline. All three of these τ-structures consist of one part. Clearly, the last τ-structure extracted is always isolated. The one in our example is decomposed into parts D and E as described above. This algorithm also produces a complete and correct part decomposition for more complex objects. The one in Figure 9 necessitates two iterations of the algorithm. In the first iteration, five parts are extracted. The first two are from a *pi*-junction seed, the third and fourth are from *mt*-junction seeds, and the last one is isolated. The second iteration extracted the remaining four parts. The first from an *mt*-junction seed and the last three as isolated parts. Each of the nine parts forms its own one-part τ-structure.

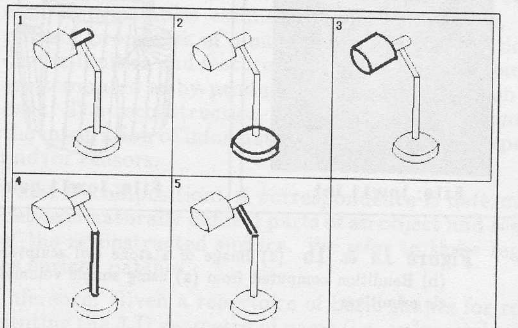

Figure 10 Part decomposition sequence for a simple object. This is obtained by applying the *PARVO* segmentation algorithm. Note the restoration of the second part.

Interestingly, this complex line drawing violates the assumption that any τ-structure connection produces either a completely invisible join in the line drawing (giving rise to a *pi*-junction) or a completely visible join (giving rise to an *mt*-junction). For instance, the left end of the sixth part in Figure 9 is occluded both by the fifth part, to which it is connected, and by the third part. Thus, the two side segments of the sixth part are actually occluded by two different line segments. These four segments do not form a *pi*-junction and, in accordance with the previous assumption, should not provide evidence for the segmentation to proceed. The solution, given such *double occlusions*, is to create a virtual segment connecting the two sides of the occluded part (the sixth part in Figure 9). No seed is created. Thus, the extraction of the sixth part is only possible using the *mt*-junction seed at its other end. However, the added virtual segment is necessary to complete the c-path of its outline. The ninth part in Figure 9 is similarly occluded by both the eighth and the fourth part. Its extraction is possible only after all the other parts are removed.

After part segmentation is completed, it is possible for PARVO to obtain a description of each part of the object in parallel. The description of the parts resulting from the segmentation process will be addressed in a forthcoming paper.

4. Summary

In this paper, we have presented an algorithm for segmenting a line drawing satisfying an object visibility model. The algorithm consists of hierarchically decomposing the line drawing of an object into its constituent parts by pairing segmentation points on the silhouette of the object. Results have been presented that demonstrate its capability to deal with single-view line drawings of complex objects. The context in which this algorithm is used is the *Primal Access Recognition of Visual Objects* (PARVO) system, a computer object recognition system based on the *Recognition by Components* (RBC) theory. Most aspects of the existing implementation of PARVO have only been outlined. A detailed description of each module and our experimental results will be published later. An early account of some aspects of PARVO is also available in [7].

Acknowledgements

The authors would like to thank I. Biederman for helpful discussions on his human image understanding theory. This research was partially supported by an NSERC Postgraduate Scholarship to R. Bergevin and the NSERC Operation Grant A1733 to M.D. Levine. M.D. Levine would like to thank the Canadian Institute for Advanced Research for its support.

References

[1] I. Biederman, "Human image understanding: recent research and a theory," *Comput. Vision, Graphics and Image Processing*, vol. 32, pp. 29-73, 1985.

[2] P. J. Besl, and R. C. Jain, "Three-Dimensional Object Recognition," *Computing Surveys*, vol. 17, no. 1, pp. 75-145, March 1985.

[3] D. G. Lowe, "Three Dimensional Object Recognition from Single Two Dimensional Images," *Artif. Intell.*, vol. 31, pp. 355-395, 1987.

[4] A. Pentland, " Recognition by Parts," Technical Note No. 406, Stanford, CA: Artificial Intelligence Center, SRI International, December 16, 1986.

[5] R. Nevatia, and T. O. Binford, "Description and recognition of curved objects," *Artif. Intell.*, vol. 8, pp. 77-98, 1977.

[6] D. G. Lowe, " Perceptual organization and visual recognition," STAN-CS-84-1020, Stanford, CA: Department of Computer Science, Stanford University, September 1984.

[7] R. Bergevin, and M. D. Levine, " Recognition of 3-D Objects in 2-D Line Drawings: An Approach Based on Geons," TR-CIM-88-24, Montréal, Québec, Canada: McGill Research Center for Intelligent Machines, November 1988.

[8] D. Hoffman, and W. Richards, "Parts of recognition," *Cognition*, vol. 18, pp. 65-96, 1984.

[9] V. Guillemin, and A. Pollack, *Differential Topology*. Englewood Cliffs, N.J.: Prentice-Hall, 1974.

[10] J. M. Beusmans, D. D. Hoffman, and B. M. Bennett, "Description of solid shape and its inference from occluding contours," *J. of the Optical Soc. of America (A)*, vol. 4, pp. 1155-1167, July 1987.