

# Structural Description with Classified Polygon and Surface Features and Their Groupings

C. Rasche  
Division of Biology  
Caltech  
Pasadena, 91125 CA  
JAN 2002

## Abstract

We describe a recognition system that categorizes furniture objects depicted in line drawings. In a feature extraction process, polygons and rectangular surfaces are gradually evolved, classified according to their shape and orientation in 3D space, and grouped according to common grouping regularities. The category representations are expressed by only a few relations of these surface and polygon features and their groupings, and are structurally only loosely formulated in order to be able to cope with the structural variability across different category instances. We use a matching approach to determine the corresponding category representation.

## 1 Introduction

### 1.1 Structural variability - Category essence

When we enter a new room, like a living or a bed room, we readily recognize its furniture within, like chair, bed and closet, and we can say what approximate orientation they have in the room. Recognizing a piece of furniture means assigning its specific structure to its corresponding basic-level category (Rosch et al., 1976; Palmer, 1999). To determine its orientation in space, its geometry needs to be interpreted. In order to build a visual system, that can readily do this we need an efficient representation. We begin searching for such a representation by examining a so far underestimated issue of the basic-level categorization process: the variability aspect.

**Classifying variability.** The basic-level categorization process is able to master a large structural variability between the different instances of a category (figure 1). We find the following types of structural variability:

1) Part-shape variability: the different parts of a chair - leg, seat and backrest - can be of varying geometry. The legs' shape for example can be cylindrical, conic or cuboid, sometimes they are even slightly bent. The seating shape can be round or square like or of any other shape, so can the backrest (see figure 1a and b for simple examples of part-shape variability).

2) Part-alignment variability: the exact alignment of parts can differ: the legs can be askew, as well as the backrest for more relaxed sitting (figure 1c). The legs can be exactly aligned with the corners of the seating area, or they can meet underneath it. Similar, the backrest can align with the seating area exactly or it can align within the seating width (compare figure 1c and d respectively).

3) Part redundancy: there are sometimes parts that are not necessary for categorization: for example, there can be an armrest (figure 1d) or stability support for the legs (figure 1e). Omitting these parts does still lead to proper categorization. [Part redundancy could be helpful in recognizing occluded objects. For simplicity we discuss only the regular, canonical views.]

**Essence.** An efficient visual system, as possibly the human visual system is one, must have a category representation that is able to cope with the described variability: Firstly, the category representation should not depend on part-shape variability. Rather it should be some structural skeleton that ideally is independent of any detailed part shapes. Secondly, due to the alignment variability, the category representation should not be rigid, but rather flexible to be able to deal with varying part alignments. Thirdly, the category representation should not contain part redundancy, but merely its structural essence. In the present work we make a first step to find such a flexible essence for some basic-level categories of furniture.

## 1.2 Related literature

The variability issue discussed in the previous subsection, is only one of two principal aspects of the recognition process. The second principal one is the viewpoint aspect (Palmer, 1999): We are able to categorize objects from different viewpoints, although the object's structure projected onto the retina, is different for varying viewpoints. We start discussing related literature according to these two principal aspects. We then shortly review what features and object models have been used so far for object descriptions.

**View-point aspect.** Many well-known object recognition systems have been developed to be able to recognize specific objects from different viewpoints, especially for industrial

applications (Robert, 1965; Brooks, 1981; Lowe, 1987; ULLMAN, 1990; Grimson, 1990). But because these systems often represent single object instances, and not a category with several, structurally different instances, they rather represent accurate object identification systems. And because they are able to recognize their objects from very unusual view points - sometimes in a cluttered background -, they rather correspond to efficient visual search systems.

**Variability aspect.** There are only a few attempts that also try to account for the variability aspect. These are mainly approaches which use a small set of primitives to describe a category. One class uses volumetric shapes, as originally proposed by Binford (Binford, 1971). For example, Marr represented objects preferentially by cylinders (Marr and Nishihara, 1978). Biederman proposed a larger set of parts, like cylinders, cuboids, wedges and more (Biederman, 1987). Shapiro et al. have proposed an even more refined set of volumetric parts, classified into sticks, plates and blobs (SHAPIRO et al., 1984). However, all these volumetric approaches share similar short-comings: First, it is computationally very expensive to extract the volumetric information of each single object part. Second, it is doubtful, whether there is a set of generic parts that allows to describe all objects. Finally, as mentioned before, it is desirable not to rely on part shapes due to the part-shape variability. A slightly simpler category representation has been suggested by Lee et al (Lee et al., 1992): Instead of reconstructing the 3D information of each part, they merely extracted 2D regions as primitives (which are used to describe categories). Although this remedies the first drawback, this approach still has not addressed yet the part-alignment variability and part redundancy.

**Features.** Many of the previously mentioned systems rely on extraction of vertices as the first step in forming complex features. This silent paradigm roots in the early work on scene and object recognition of polyhedral volumes: Guzman developed a program that was able to recognize a scene made of cuboid-like building blocks (Guzman, 1969). The program started by extracting the vertices, followed by determining regions (or areas), which then were searched for groupings that form volumes. Later, this approach has been formalized (Clowes, 1971; Huffman, 1971) and extended (Waltz, 1975). Yet, vertex extraction is not the only choice there is. An alternative is to segment a scene or object into closed polygons. For example Robert has used a square or a '6-line' polygon delineating a part or the entire outline of a cube, respectively (Robert, 1965). Lin has developed a system to recognize industrial parts by surfaces like rectangles or more complicated polygons (LIN et al., 1984).

Surprisingly there has not been much more work on the usage of these polygons features.

**3D reconstruction, object model.** Because a 3D object is projected onto a 2D retina (or camera image), a valuable source of information is lost: depth. In recognition systems, this depth loss is either ignored or compensated for in different ways, depending on the task to be solved.

In the work on polyhedral scenes (Guzman, 1969; Clowes, 1971; Huffman, 1971; Waltz, 1975), which generally aimed at segmenting a scene into its objects, neither a model nor a 3D reconstruction was necessary: it sufficed to interpret 2D features like vertices. Roberts work aimed at segmenting the scene and at determining the pose of objects (Robert, 1965). He employed 3D models of its scene objects. To find the appropriate object model he extracted 2D features like polygons of the object's outline, which served as indicators for choosing and testing possible 3D object models. Speaking in terms of 3D reconstruction, it was essentially a '2D feature - 3D model' matching process. Lowe developed a similar 2D-3D recognition system (Lowe, 1987): Instead of using 2D polygon features, he carefully analyzed the extracted lines for grouping regularities. Using such 2D groupings, he matched a 3D object model against it. Marr devised a recognition system, in which first a 2.5D sketch is evolved, which is then matched against a 3D model (Marr, 1982). Brooks has elaborated this approach using a prediction-verification cycle and built a system that identifies objects and determines their orientation. The problem of these approaches - as noted before - is that they have focused on the view-point aspect and are thus difficult to extend to the variability aspect. For example, what would be a 3D model of a category, whose instances all show structural differences? Furthermore, 3D reconstruction is computationally expensive, because a 2D coordinate has to be matched to a 3D coordinate (or the other way around). Approaches with 2D models, which basically correspond to a 2D-2D matching process, like (SHAPIRO et al., 1984; Lee et al., 1992), are thus computationally much less intensive but have not shown orientation specificity.

### 1.3 Our recognition system

**View-point: Canonical views.** Our work focuses on objects depicted in canonical views (see figures 13-15). Any other views are less familiar (Palmer et al., 1981) and they therefore likely require longer inspection time until the visual system has recognized the object. Thus, recognizing an object from an unfamiliar view, rather corresponds to a visual search and an intense memory search, in which the most likely category representation has to be found.

Here we are concerned with constructing an efficient category representation in a first place, which later may be applied for recognizing unfamiliar views.

**Variability and features.** As reviewed before, there are two feature classes to reconstruct the shape of volumetric objects: vertices and surfaces. Figure 2a shows a possible reconstruction of a desk by its five vertices (We here consider a vertex made of three intersecting lines). This approach was so strongly advocated, because it was reasoned on building blocks whose surface borders touch each other exactly (Guzman, 1969). In real-world objects however, this is not always the case. As introduced before, there is part-alignment variability. For example the desk plate might overlap the two corpi and hence three of the vertices 'loosen up' (figure 2b, [compare also figure 1c and d]). What remains more stable and is easier to reconstruct, are the rectangular surfaces that are projected as rectangles and trapezoids onto the retina (or parallelograms if it is a line drawing in parallel projection) (figure 2b). We extract such, as we call them now, '3D rectangles' and classify them according to their geometry, which gives us also information about the objects orientation in 3D space (yet we do not assign any 3D coordinates with it). For clarity, we will use the term '2D rectangle' if we mean a rectangle or square in the frontal plane.

In addition, 3D rectangles often appear in specific patterns, as for example the drawers in a desktop are nested within a larger rectangle. We thus group rectangles according to these patterns.

**3D reconstruction and object model.** Naturally, one would like to represent a chair as being made of legs, a seat and a backrest (as shown in figure 1a and b). This is also the way, many part-based approaches represent objects: objects are represented the way we humans can label the object's parts. But is this necessarily also an efficient visual representation? Rather we will also use polygon features which are characteristic to part of the outline of an object, similar to Roberts method of using polygons for model indexing. For cuboid-like structures, as the desk, we represent the volume-like structure by rectangle groupings that reflect folded surfaces. The category representations are generally loose skeletons of characteristic 3-line polygons and 3D rectangles and their groupings that we specify. Such a skeleton is supposed to represent the flexible categorical essence that is able to find its fit to the object's structure extracted in the feature extraction process. Stated simplified, a 2D model of a category representation is fitted against the 2D output of the feature extraction process.

**Recognition evolvment.** In a feature extraction process, 3D rectangles are gradually evolved starting with L-type features, followed by their coupling to form 3-line polygons, which in turn are coupled to form 3D rectangles. This evolvment occurs according to grouping laws (Lowe, 1987). In a feature grouping process 3D rectangles are grouped according to common patterns. By then, the feature evolvment process has created many relations between different features and because of part redundancy, there are also many 'accidental' features extracted, which are irrelevant for categorization. For that reason we have to describe the category representation ourselves. We use - for reason of simplicity - a matching approach to determine the category: each category representation searches for its category-specific features in the feature evolvment output. Thus, we do not attempt to fully analyze the feature evolvment output, rather we let the matching step look for the characteristic essence of a specific category.

**Choice of objects and category variability.** We will seek the category representation for a few furniture pieces depicted in line drawings (figure 13-15). The categories are chair, desk, bed, table, drawers and closet. We chose the following two main restrictions on the category variability: 1) We use only straight lines. 2) We use only 3D rectangles as surfaces. The drawn objects show part-alignment variability and sometimes part redundancy.

## 2 Feature extraction and grouping

**Images.** Objects were drawn in a simple graphic program. Image sizes vary from 300x300 to 500x700 pixels, the line width is 5 pixels. Objects are drawn mostly in a parallel projection.

**Contour extraction.** The choice of edge-detection algorithm was arbitrary. We use the Canny algorithm (Canny, 1986). We extract contours only from the first scale with a low high threshold. Due to the aliasing problem some contours are broken and represent collinear lines, which are joined in the line extraction step.

**Line extraction.** For (straight) line detection we use Lowe's algorithm (Lowe, 1987), which recursively breaks contours at their maximum deviation from the line connecting the contour's endpoints. Lines that lie parallel and proximal, and are of about the same length, likely represent a drawn object line, and are thus fused. Due to the noisy edge detection process, especially in corners, we are not able to find all such object line groupings and therefore 'double' lines are sometimes left. We link collinear lines, with a short gap,

resulting from lines that are split into two separate line pieces due to an intersecting line or due to aliasing. The double lines will cause redundant features, features that are closely spaced and of similar size and orientation. They are eliminated in a later step of feature extraction.

**L features (L).** Detecting L features bears the following, occasional problem: Due to sometimes insufficient contour extraction and noisy line extraction, corners are not always exactly captured by lines. It therefore requires a tolerant measure to determine whether two lines are proximal enough to form a L feature (figure 3a). If the measure is too tolerant, accidental grouping can occur. A common case is shown in figure 3b. These accidental groupings can be avoided by using an algorithm checking the context of an L feature, a process which we have not modeled here. The measure for L-feature formation is as follows (Figure 3c): Firstly, we determine the intersection point,  $p_i$ , of the rays of the two lines. Secondly, we measure the distances between  $p_i$  and the lines endpoints,  $p_i p_2$  and  $p_i p_3$ . The two lines are considered as a L feature if the sum of the two gaps is smaller than a fraction of the length of the shorter line. This criterion has the property that the more the orientation of a line points directly to the endpoint of another line, the more distant they can be. The intersection point is the corner point of the L feature.

We make no use of T junctions: they are split into two L features (figure 3d). As for L feature formation, T junction formation requires a tolerant measure as well: Two lines are considered as a T, if the distance between the stem's line and intersection point is shorter than a fraction of the shorter line. When we split the T feature, we do not cut the intersected line itself, but add these two L features to the list. Due to the alignment variability, many T junctions create one L feature with one very short leg (see figure 3e, L no. 2). We discard such short-legged L features.

The subsequent feature groupings serve to evolve 3D rectangles (2D rectangles, trapezoids, parallelograms). Their evolvment is also visualized in figure 4.

**Polygon features (pg3).** Pg3s are three sequentially connected lines. They are detected by browsing the list of L-features for common lines (indicated by the two close, parallel lines in figure 4). We will hereafter call the line connecting the two corner points, the middle line, and the two other lines, the legs. Pg3s are classified into U-like and Z-like features, whose legs are on the same side, or on opposing side of the ray running through the middle line, respectively.

**3D rectangle precursors (rect indicators).** Rect indicators, Upills and Usang, are 'precursors' for 3D rectangles. Upills are U-like features with parallel legs. Usang are U-like features whose angles between the middle line and the legs ( $\alpha_1$  and  $\alpha_2$ ) are the same. They are useful for finding incomplete trapezoids.

**3D rectangles (rect).** 3D rectangles are detected by coupling Upills' and Usangs (see figure 4 top). Because 3D rectangles are sometimes incomplete due to part-alignment variability, we use different coupling formations to complement them (cases 3 to 4).

Case 1: Two Upills, that share the same legs, form a 3D rectangle, trapezoid or parallelogram, depending on the angle between the Upills' middle lines and their legs.

Case 2: This is case 1, but one side of the 3D rectangle is formed by two L features whose legs overshoot the 3D rectangle's side length.

While in the first two cases, the corner points of the newly formed rectangle are given by the two coupled Upills', in the next two cases, one corner point is missing due to part-alignment variability.

Case 3: Two Upills', that share one middle line and one leg, form a 3D rectangle or parallelogram, depending on the corresponding angles.

Case 4: One Upills and one Usang, that share one middle line and one leg, form a trapezoid.

For cases 3 and 4, we have to find a tolerant measure for accepting the coupling as a 3D rectangle, similar to L and T formation, otherwise accidental 3D rectangle formation occurs: We use the same measure as for L feature detection except that the measure is dependent on the 3D rectangle's total side length, which allows us to be more tolerant.

As mentioned in the line extraction paragraph, many redundant features are generated due to the double lines, but also due to closely spaced lines like in case of drawers. This results in many similar sized 3D rectangles overlapping each other. We discard smaller 3D rectangles residing within a slightly larger one. It therefore sometimes happens, that not exactly a drawer rectangle is left but one that delineates the boundaries of the drawers and the desktop's chest. This does not affect our categorization process, however, we could not analyze structurally fine details, something we do not aim for here.

**Cuboid/Classification of 3D rectangles.** Some of the furniture categories, in particular desk, closet and drawer have a cuboid-like global structure. We explain now how we would represent a cube from certain common view-points by classifying 3D rectangles. Representing cuboid-like furniture pieces is a deviation of this scheme. (When we talk about rectangles



and parallelograms we imply that these are squares and rhomboids for cubes.)

Our real world is projected in a perspective view onto our retina. Still, we are able to easily recognize parallel (orthogonal) projection of cuboids. We therefore represent volumes and derived categories for both projections. Figure 5 shows a cube in parallel projection, left in a frontal/side view (1) and right in a pure side (side/side) view (2). The same two views are also shown for a perspective projection (3 and 4, respectively). In terms of 3D rectangles, the cubes consist of 2D rectangles, parallelograms and trapezoids. 2D rectangles simply represent frontal views of rectangular surfaces. Parallelograms and trapezoids represent rectangular surfaces, tilted and slanted in 3D space. We classify 3D rectangles according to the slope of their parallel sides (figure 5b): If the parallel sides (for a parallelogram it is either two parallel sides) are vertical, then we define the 3D rectangle's tilt as  $\pi/2$ , (which corresponds to the slope of the parallel sides in radians) and the 3D rectangle is classified as 'wall', because it likely represents a standing surface. If two parallel lines are horizontal, then we define tilt as 0, and the 3D rectangle is classified as 'tile', because it resembles the typical geometry of a tile in a floor. If the parallel sides are neither vertical or horizontal, then we classify the 3D rectangle as a plate (figure 5c), because it is its most likely appearance. A plate's tilt is defined by the angular slope of the parallel sides (in case of a parallelogram we take the longer pair of the parallel sides)

The slant for 3D rectangles is estimated as follows. For walls and tiles we measure the angle between the slanted side and the orthogonal of the corresponding parallel sides (see arrows in figure 5b). For asymmetric trapezoids we take the larger angle between the side and the orthogonal of the parallel sides (see perspective/horizontal case in figure 5b). For plates we take a very crude measure: we measure the angle of the lowest L feature ( $\gamma$ ) and subtract  $\pi/2$ .

Walls are subclassified into 'left' and 'right': Left ones appear on the left side of the cube, right ones on the right side. Walls are also subclassified according to their orientation into 'high' and 'wide' (if they are rectangles (2D or 3D) and not squares). In high walls, the vertical side is the longer one and the 3D rectangle appears standing. In wide walls, the vertical side is the shorter one and thus the slanted side makes the 3D rectangle appear lying.

Summarized, we have classified 3D rectangles according to their shape and orientation, firstly into walls, tiles and plates. Walls are subclassified into left and right, and into high and wide. Using this notation of classified 3D rectangles, the cubes in figure 5a can be represented as follows. Cube 1 and 3: One 2D rectangle (frontal view), one right wall and one tile. Cube 2 and 4: One right wall, one left wall and one plate. If it is a cuboid, then

we can read out the wall’s wide/high classification and we know whether it is a standing or lying cuboid.

This form of representation has three advantages: firstly, it is independent of the type of projection; secondly, it is independent of view-point to some degree; thirdly, the cube’s orientation in 3D space can be easily estimated by reading out its 3D rectangles’ estimated tilt and slant.

**Rectangle Grouping.** In order to find cuboids we need to search the list of 3D rectangles for couples that share a side, which is parallel and close. While searching such 3D rectangle groupings, we can easily search for other typical 3D rectangle groupings. We thus start by searching 3D rectangles which are ‘adjacent’: they have one or more sides that are parallel and close. A 3D rectangle can have several adjacent 3D rectangles, which are grouped into a list. In such an adjacent group, the largest 3D rectangle is stored first and called ‘head’. The smaller adjacent 3D rectangle(s) is called ‘dependent(s)’ and are listed by area size. Adjacent 3D rectangle groupings are then further classified into ‘nested’ and ‘folded’.

**Nested:** If two 3D rectangles share two or more sides, then the smaller one is nested within the larger one (figure 5d). We call the larger 3D rectangle the ‘principal’ rectangle. If two sides are adjacent, then two nesting cases are possible: In the ‘corner’ case, the smaller 3D rectangle touches the two sides of one rectangle’s corner, in the ‘sliced’ case, the smaller rectangle touches two opposite sides. If three sides are adjacent, then we call it ‘embedded’. Nested groupings are useful for representing drawers for example. It is also possible that in the corner case, the smaller rectangle overshoots the larger in one direction. This sometimes accidentally occurs during feature extraction, yet we are generally not interested in that case.

**Folded:** Folded groupings are precursors to extract volumes. We search the adjacent groups for folded surfaces, whereby we exclude any dependents that are nested. Two adjacent 3D rectangles are considered as a folded surface when they differ in their geometrical classifications: left wall, right wall, plate and tile. Any two 3D rectangles of the four cubes in figure 5a differ in those classifications. It might be noted again, there are more possible options to extract foldable surfaces, but these are the principal ones found for the regular views of cubes (cuboids).

Another type of 3D rectangle groupings is ‘parallel’ 3D rectangles. Two 3D rectangles are parallel if all sides of one rectangle are parallel to the corresponding sides of the other, distant rectangle (they are not adjacent). We distinguish two cases: aligned and shifted (figure 5e). In the aligned case, two opposing parallel sides of each 3D rectangle are collinear and they likely share the same plane. In the shifted case, no such collinearity exists, and the

3D rectangles are likely to be in two different planes in 3D space.

### 3 Category representation

We now describe the essential structure for the categories, that we created intuitively. The structures are figurally shown in Figure 6, yet they neither express the looseness of some structures, nor do they reflect detailed geometry of features.

**Chair.** Applying the idea of using rectangles for representations, one could think of representing the seat and backrest as a folded surface (two adjacent rectangles with folding angle). Yet, in a canonical view the seat often appears as a thin parallelogram/trapezoid, which is not always evolved as 3D rectangle due to the noisy feature extraction process. There are more obvious structures that are very characteristic to chairs: certain pg3s. For example the frontal legs and the connecting seating contour form U-like feature whose legs point downwards, which we call a 'bridge'. As the chairs legs are sometimes askew, we define the direction of the bridge's legs loosely with a tolerant angle (see  $\alpha$  in figure 6a, chair, 'bridge'). Because the bridge feature per se is not specific enough for chairs, many bridges are found in other categories as well. We therefore combine two bridges to a 'double bridge', sharing a close or even common leg. The middle angle of both bridges should differ by a minimum angle (see  $\alpha$  in figure 6a, chair, 'double-bridge'). Generally, such double bridges are characteristic to objects having legs as part of their structure and are thus also found in tables and beds. A second characteristic pg3 feature is the 'seat': A chair's leg the seating contour and the backrest contour form a Z-like feature (figure 6a). We have not used it in our representation but merely point out its characteristic structure. The backrest is represented by a 3D rectangle. Because the backrest can be askew as well, the 3D rectangle has a maximum slant. Both, the backrest and the double bridge should be in proximity, for example not farther away than a fraction of the backrest's or double bridge's size. This structure is graphically vaguely expressed in figure 6b.

The chair's orientation can be estimated by analyzing the geometry of its double bridge: The two middle line slopes of the double bridge are used to determine the slant.

**Desktop.** Desktops are generally made of a plate and one or two corpi containing drawers. The conditions for the plate are a 3D rectangle of type tile or plate, with a maximum tilt and a minimum slant. The conditions for a chest with drawers are a principal 3D rectangle of type wall or rectangle, of square shape or high orientation, containing a nested 3D rectangle

of type sliced or embedded. The principal 3D rectangle of the chest and the plate should form a foldable surface.

The desktops orientation is easily given by merely looking up the geometry of the chest’s 3D rectangle. If it is a 2D rectangle, we know it is a frontal/side view, if it is a parallelogram or trapezoid we can look up its left or right orientation and its slant.

We point out another characteristic feature grouping that is unique to some desks: parallel aligned 3D rectangles. They occur if the desk has two corpi.

**Bed.** The most characteristic feature is the parallel, shifted grouping of the two 3D rectangles, which we call head and foot wall. The lying surface is described, with the same conditions as for the plate in the category desk. It should form a foldable surface with the head and foot wall.

The bed’s orientation can be read for example from the slope of the line connecting the two 3D rectangles’ middle points of the parallel grouping (figure 5e). Or it can be looked up by reading the slant of the head or foot wall.

**Table.** We represent a simple table, with four legs and a plate, by a double bridge as for chairs, and a 3D rectangle of type tile or plate, with conditions that are the same as for the desktop’s plate. The double bridge and the plate should be in proximity, for example the plate should be close to the middle lines of the double bridge.

The tables orientation can be read by either analyzing the double bridge or the plate.

**Closet and drawers.** Closets appear generally as large standing cuboids, having doors (or maybe movable walls) that are nested within the cuboid. We therefore represent them by a principal 3D rectangle of type wall or 2D rectangle, containing nested 3D rectangles of type sliced or embedded and high orientation. Drawers are similar to closets. We represent them by a principal 3D rectangle of type wall or rectangle containing 3D rectangles of type sliced or embedded and wide orientation.

The orientation of both categories can be read out by reading the geometry of the principal 3D rectangle.

The category representations described so far do not fully capture the typical geometrical proportions of a category. The category representations therefore occasionally overlap with others either because they are similar or due to accidental feature grouping, which is often caused by category redundant structure. In the following result section we discuss these

overlaps and describe little amendments on the proportions of the category structures, that further distinguishes themselves from the others, especially from accidental groupings.

## 4 Results

**Recognition speed.** The recognition system has been programmed in Matlab. For most objects, the edge detection step (Canny algorithm) and the step from pg3 to 3D rectangles is the slowest part of the feature extraction and grouping process. The speed of the matching process depends to a large extent on the number of features extracted in the feature extraction and grouping process. In general, desks and drawers with their many 3D rectangles, require longer matching time for any of the category representations. The overall time for the matching step is about one third of the feature extraction and grouping process.

**Feature notation.** Figures 7 to 10 show the typical results of the feature extraction process for one object of each category. Black solid lines represent the lines left after the line detection process. L features are denoted with little chevrons. Rectangles are indicated by four stippled lines connecting the rectangle's middle point and its corner points. The classification of rectangles is listed on the right with the following notation: the first letter indicates the classification wall/tile/plate, 'w'/'t'/'p' respectively (there is no letter for 2D rectangles). If the 3D rectangle is of type wall, then the second letter indicates the left/right subclassification, 'l'/'r' respectively. If the 3D rectangle is not a square and therefore a 'true' rectangle in 3D space, its high/wide orientation is given, 'h'/'w' respectively. Then follows the tilt and slant, rounded to the first decimal after the comma. Noted last is the estimated 3D area. Rectangle groupings are listed on the left in two blocks: The upper block lists all adjacent groupings (including the nested ones), the lower block lists the folded groupings. The notation for the adjacent block is as follows: Each line lists an adjacent group. The first rectangle is the head rectangle, the following rectangles are dependents. If the head rectangle is a principal one, it is marked with '[p]'. If a dependent is nested it is marked correspondingly with '[s]' or '[e]', standing for sliced and embedded respectively. The second block lists the folded rectangles, with a similar notation pattern as the first block: The head rectangle is listed first, followed by its dependents. A third block - only occurring for some desks and beds - lists the parallel groupings pairwise.

**Double bridge.** As mentioned before, the double bridge is characteristic for all objects having legs as part of their structure. Double bridges are thus expectedly found in all chairs,

tables, beds (numbers 1, 3 and 6) and two desks (number 7 and 8) and one drawer (number 5). In many chairs and tables, numerous accidental bridges are formed with the fourth leg in the background. Accidental bridges are also found in another three desks: these are mostly double bridges with unequal leg lengths resulting for example from splitting T features into two L features.

**Chair.** The chair structure, as described in the previous section is detected in all chairs, in three beds (numbers 1, 3 and 6) and one desk (number 7). The accidental chair in desk number 7 is found in the double bridge of the chest and the drawer rectangle just above. To further refine the chair structure, we introduce a geometrical constraint on the bridge feature: it is only accepted if the sum of the two leg lengths is larger than the length of the middle line. This expresses that the legs of a chair are generally longer in proportion to the seat length than for a bed. With that condition, all the chairs are uniquely distinguished from all other objects, including the accidental chair detection in the desk.

**Desktop.** The desktop structure, as described previously, is distinguished by all objects except one table (number 5, shown also in figure 9). This table contains a nested 3D rectangle in its redundant structure that is an almost vertical embedding. We therefore introduce a condition in which, the nested 3D rectangle has to touch both vertical sides of its principal 3D rectangle. This condition distinguishes itself from the accidental grouping in the table.

**Bed.** The bed structure is detected in all beds and two chairs (numbers 5 and 6). The two chairs possess redundant structure that accidentally contains the beds structure: In both chairs, the seat’s 3D rectangle corresponds to the plate of the bed; in chair number 5, the head and foot walls are detected in the 3D rectangles extracted within the arm support; in chair number 6, the head and foot walls are detected in the backrest’s 3D rectangle and the 3D rectangle detected within the leg support. To further refine the bed structure we introduce the following constraint: the bed’s plate 3D area should be larger than any other 3D area of a rectangle in that structure by a certain degree. Because the seat area of chairs is generally smaller compared to its entire structure, we are able to distinguish chairs and beds completely with that condition.

**Table.** The table structure is detected in all tables, in all chairs, two beds (numbers 1 and 3) and three desks (numbers 1, 4 and 8). We introduce two conditions to further characterize the table structure: 1) We look for bridges with a) similar leg lengths, which excludes the

accidental choices of desks 1 and 4, b) summed leg length that is larger than the middle line length by a certain degree, which excludes the two beds. 2) We chose only plates with a certain rectangular dimension, which excludes all chairs, which generally have a square like seating area. Desk number 8 could not be distinguished from the category table, because it actually contains the structure of a table, but containing drawers.

**Drawers and closets.** The drawers structure is recognized in all drawers and desks. In order to further distinguish between these two, we introduce the following constraint to drawers. The principal 3D rectangle, containing the nested 3D rectangles, has to be the largest 3D rectangle of the object. The closet structure is detected in all closets, two drawers (numbers 1 and 3) and one desk (number 2). The two drawers contain high 3D rectangles caused by the center divider, the desk contains high rectangles, which are a combination of for example two drawers, extracted as one 3D rectangle due to the splitting of T features into L features. If we introduce the same condition as for drawers (with the principal 3D rectangle being the largest 3D rectangle), we can differentiate the accidental desk detection. To further differentiate from the two accidental drawers, we introduce the condition, that the nested 3D rectangle can not contain any nested 3D rectangles itself.

## 5 Discussion

**Geometrical refinements and context.** We could distinguish all categories from each other by only a few relations and occasionally refined proportions of the structure except for one desk that is recognized as a table. The reason is that a desk has the same structure as a table but contains additional structure delineating for example the drawers. This emphasizes again, as noted already for accidental L feature detection, that a context mechanism could be useful to solve this dilemma. Such contextual information could also help distinguish other categories for which we had to refine the structural proportions, for example the differentiation of the bed structure found in two chairs, or the closet structure found in two drawers. Thus, structural relations per se are possibly only part of an efficient category representation.

**Complexity of features and category representations.** We use 3-line polygon and 3D rectangle features and their groupings to represent categories. Although these features are not more complex than vertices, we gain a lot of information from their detailed geometry, like shape and orientation. Using such feature information and the feature groupings,

enables us to formulate category representations of only a few relations. Other approaches in contrast, that used simple parts, no groupings and no interpretation of geometry had to specify category representations, made of more relations (SHAPIRO et al., 1984; Lee et al., 1992). Additionally, as noted before, the formulation of category representations with 3D rectangles bears a large degree of view-point independence: the loose skeleton of (grouped) features simply swallows different poses as long as they are in a canonical view.

**Grouping regularities.** Although, the evolvement of 3D rectangles and their groupings is chosen intuitively and seems natural to us, it is noteworthy that it might occur according to some grouping regularities that others have already conceived long before. Wertheimer, a Gestaltist, devised a set of grouping phenomena (Wertheimer, 1958), which recently have been extended (Rock and Palmer, 1990). We feel that a comparison of these grouping laws against our choice of groupings, is worth another study.

**Matching and evolvement.** A matching approach, as we use it here, is possibly not the best solution for a large database of categories. Despite this matching step, there is a continued evolvement, which is equivalent to indexing approaches. The feature extraction and grouping process evolves features, which are characteristic for a category. For example, the many 3D rectangles extracted and grouped in desks are thoroughly checked by similar category representations like drawers, but other category representations that consist of different features like chair, are quickly discarded. Thus, the feature extraction process generates a specificity, which will rapidly falsify unsuitable category representations during the matching process.

We have represented certain category representations with complex features, for example a double bridge, which are not evolved in the feature extraction and grouping process per se, but are evolved when we looked for individual category matches. Because the double bridge occurs in all objects that are standing on legs, one could move it into the feature extraction and grouping process as well. Indeed, there is no particular reason for this separation, we have simply focused on merely extracting general 3-line polygons and 3D rectangles in the feature extraction process and on grouping of 3D rectangles in the feature extraction process.

**Essentiality of category representations.** To get an object recognized and distinguished from other categories, we need only to check a fraction of its structure, unlike other recognition approaches, that analyzed the object’s full structure (SHAPIRO et al., 1984; Lee et al., 1992). In fact, we have taken an almost opposite route by searching for some

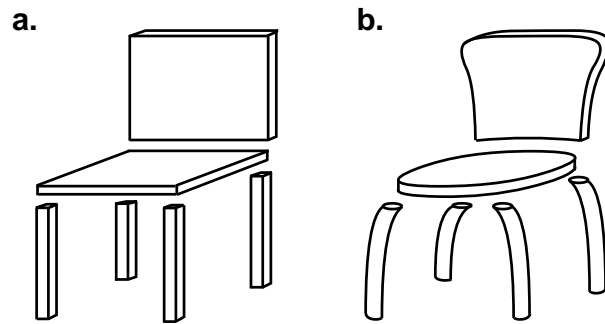


sort of minimalistic representation: we have developed an essential structure that allowed one category to distinguish itself from the others. Because the essential structure is a loose skeleton, one could devise objects that either have rare or unusual structure, or are even physically impossible (like impossible objects in illusions), with which the described recognition system could be deceived. For example, in most objects surfaces align exactly (apart from alignment variability): yet we have hardly checked whether two surfaces actually align exactly, or whether they overlap only partially, for example by one half only; or we hardly check the exact nesting of rectangles: they could be of type sliced and overlap, which actually sometimes happens due to the redundant feature extraction. The recognition system could be therefore deceived, with objects having a lot of structural inadequateness (which in turn does not occur very often in our environment). Thus, our system as it is, serves good for a quick categorization and pose determination, which is what we aimed for. A subsequent process could analyze the object's full structure in detail and determine whether it is possible. This requires a thorough analysis of for example the surface foldings that are physically possible.

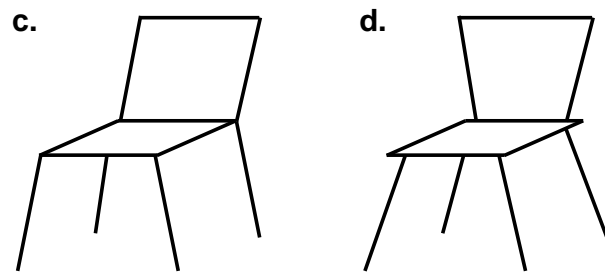
## 6 Summary

- 1) We have concentrated on line-drawing objects in canonical views with reduced part-shape variability (straight lines and 3D rectangles only), yet with part-alignment variability and part-redundancy.
- 2) Surface (3D rectangles) and polygons were evolved. 3D rectangles were classified into wall, tile and plate, and the corresponding tilt and slant was determined. Walls are subclassified into left and right, and high and wide.
- 3) 3D rectangles are grouped into nested and parallel. Nested groupings are of type sliced or embedded. Parallel groupings are of type aligned or shifted.
- 4) Given these 3D rectangle classifications and their groupings we are able to formulate concise category representations that suffice to distinguish the categories roughly. We have occasionally adjusted the structural proportions to refine the distinction between categories. Category representations are loosely formulated in order to be able to cope with the structural variability across category instances.
- 5) The orientation of the object can be determined by reading out the slant of one of the significant features.

**part-shape variability**



**part-alignment variability**



**part redundancy**

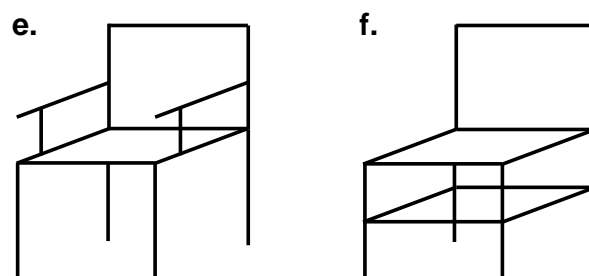


Figure 1:

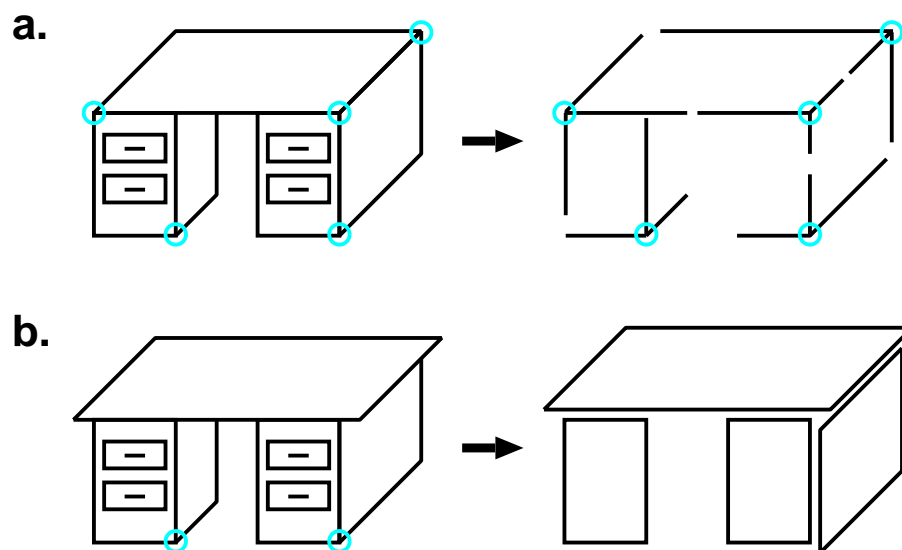


Figure 2:

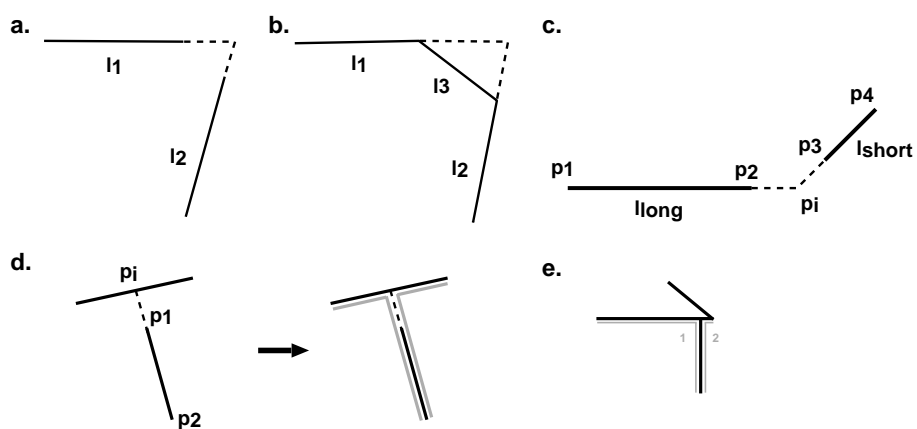


Figure 3:

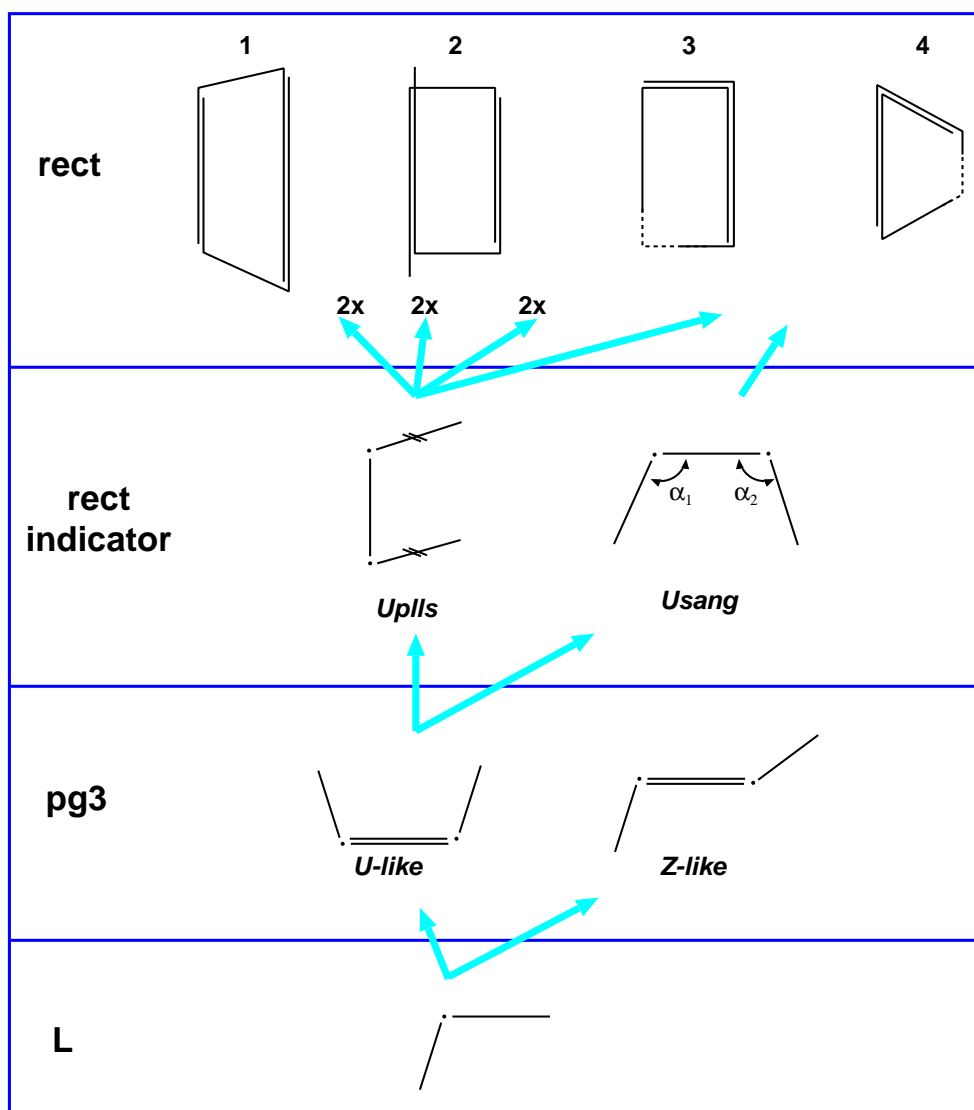


Figure 4:

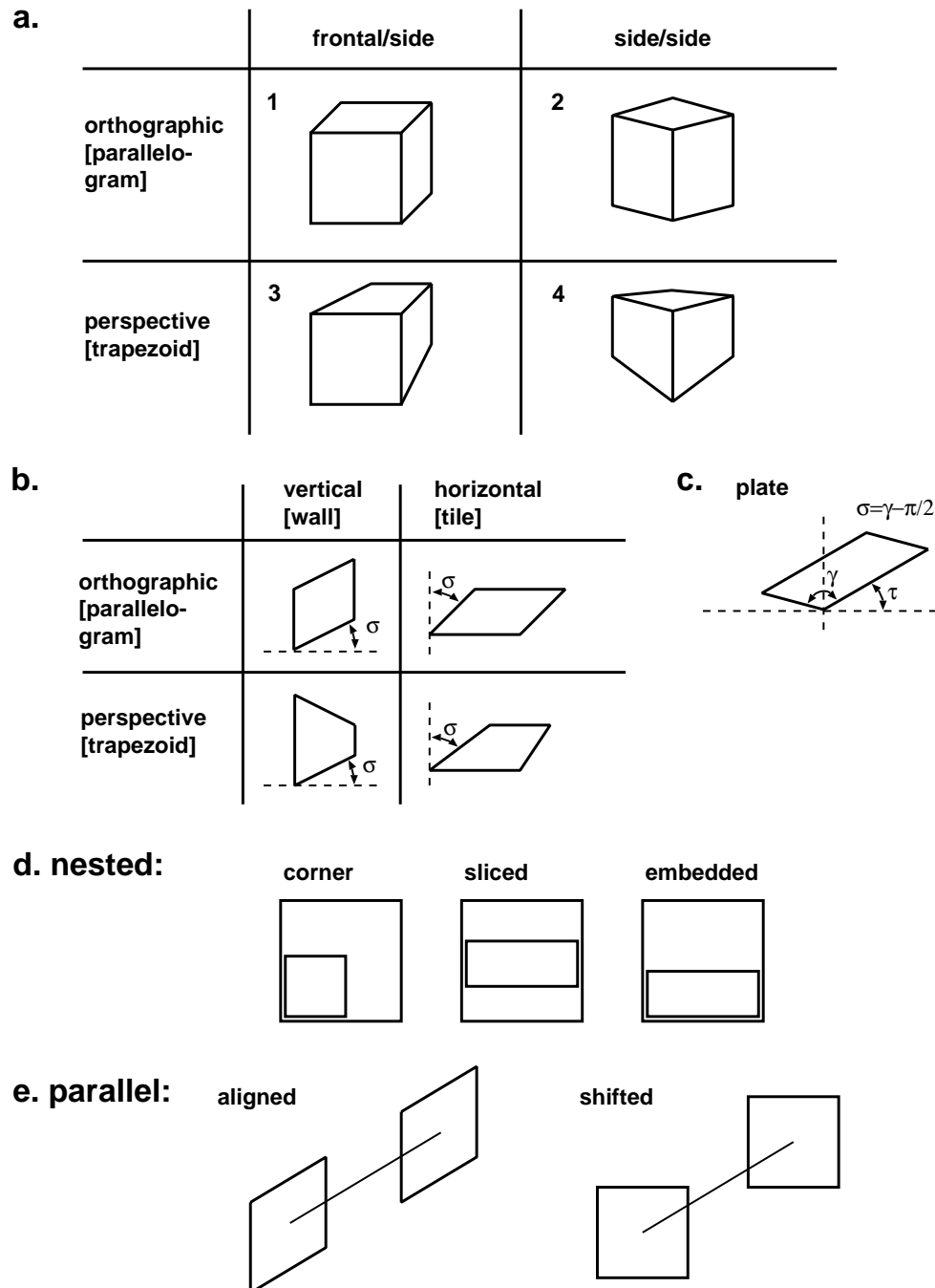


Figure 5:

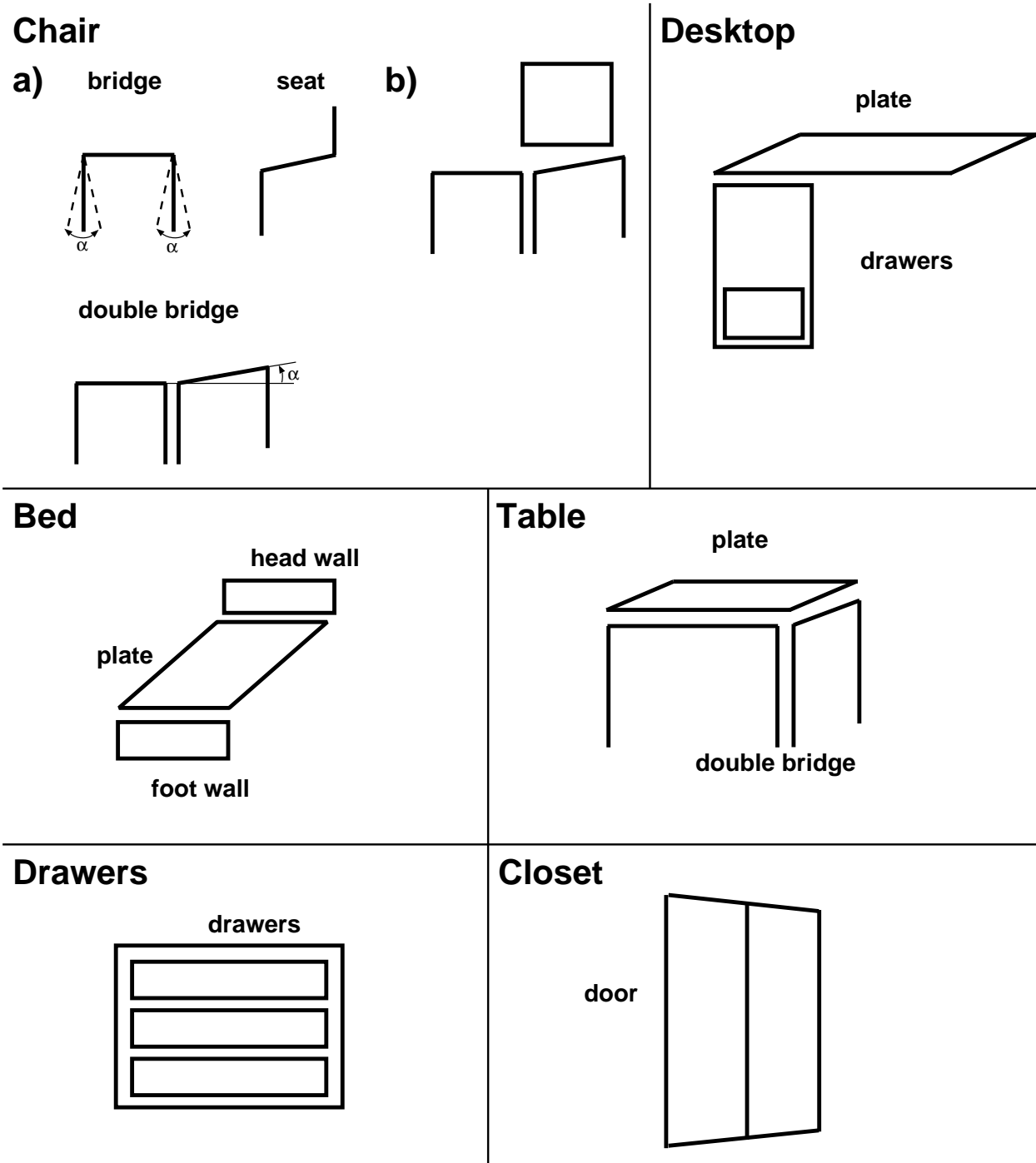


Figure 6:

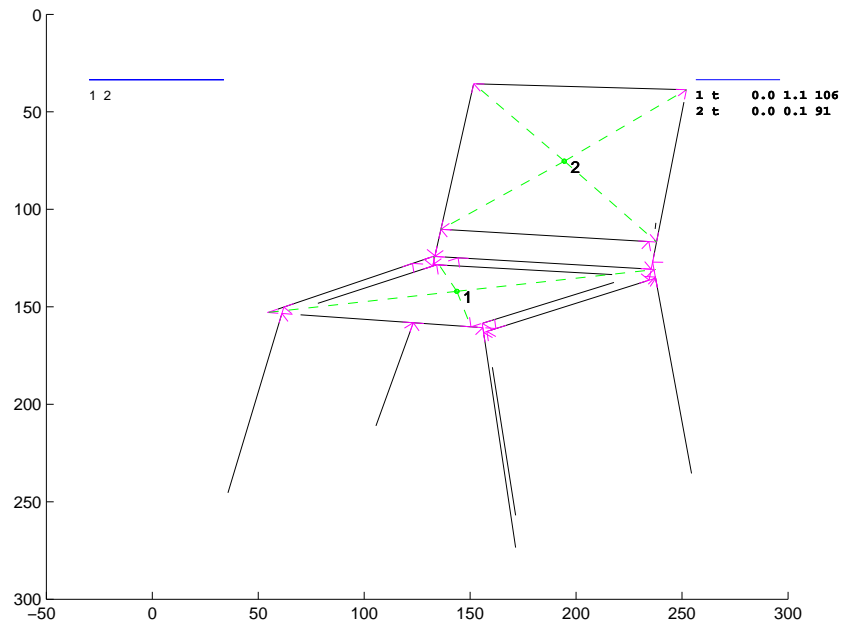


Figure 7:

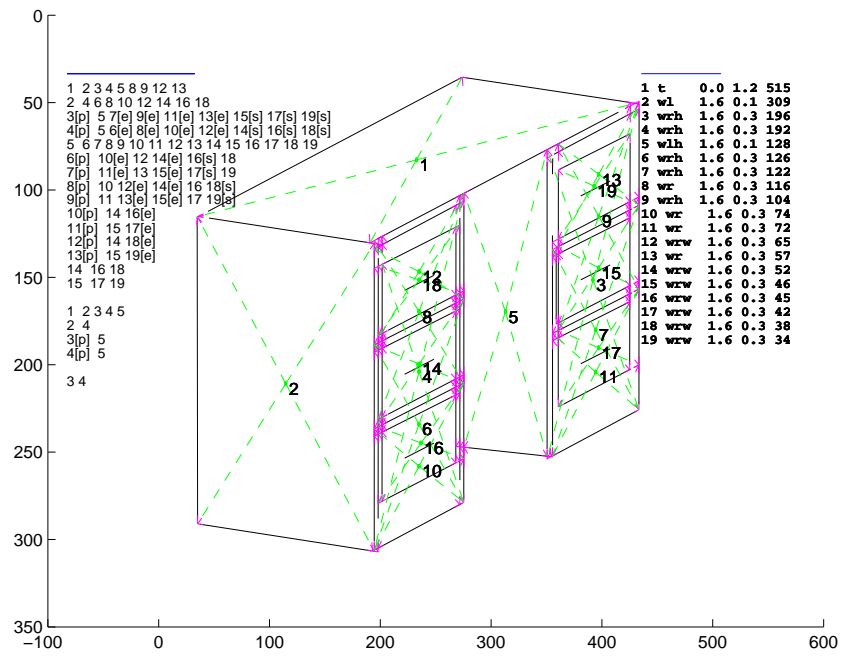


Figure 8:

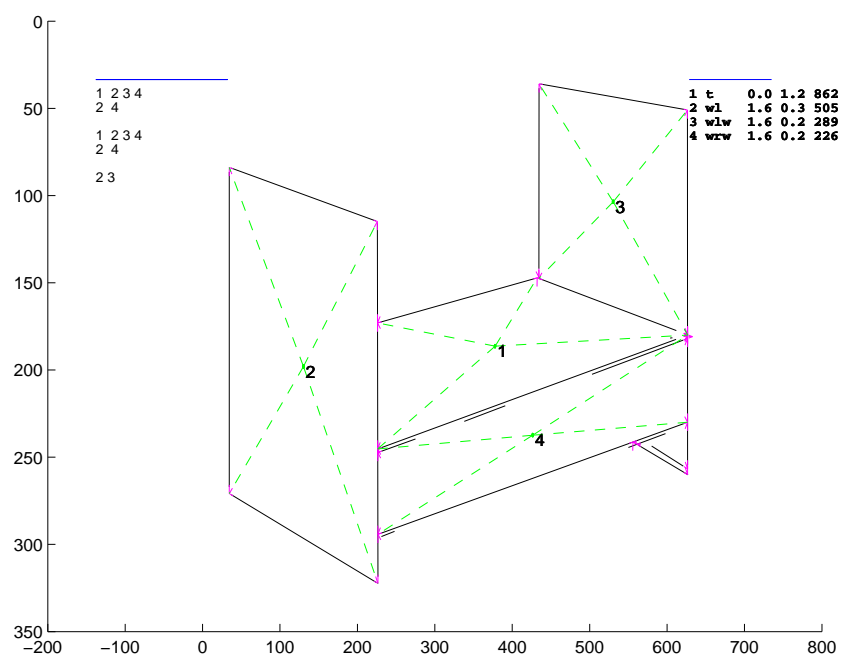


Figure 9:

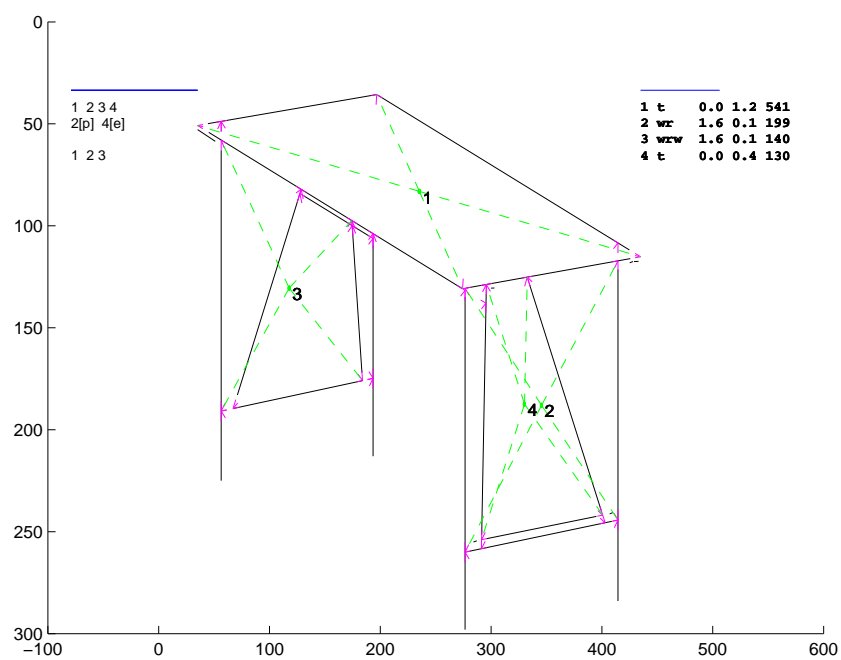


Figure 10:



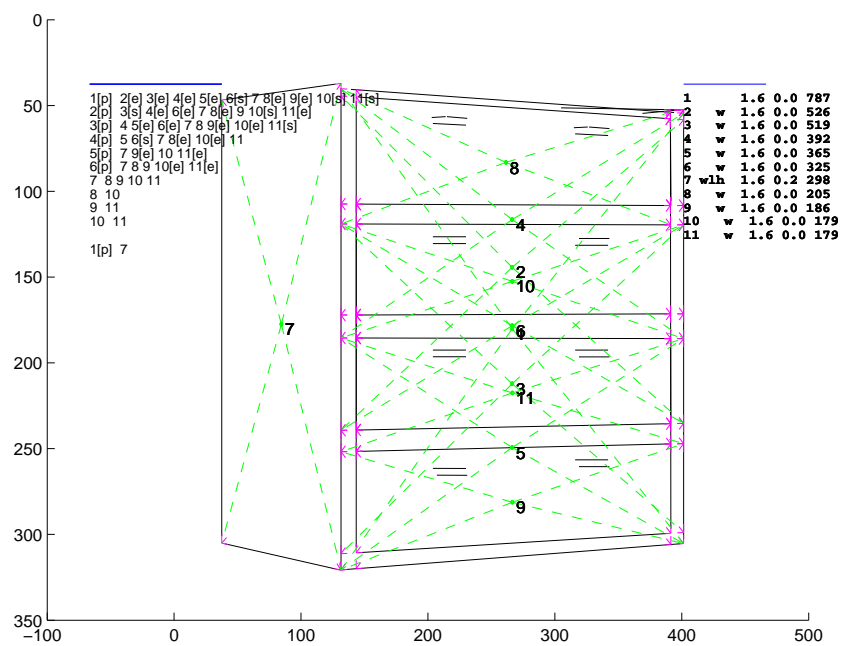


Figure 11:

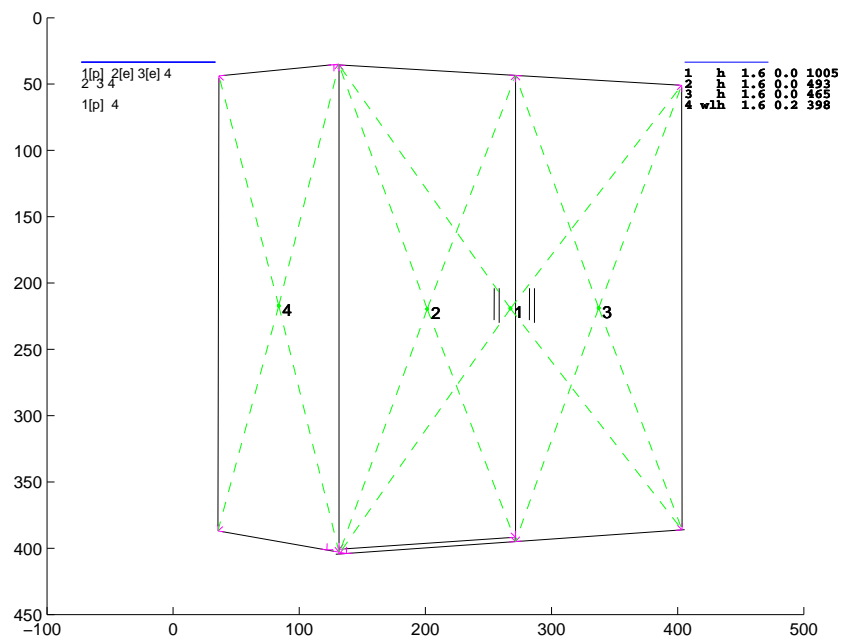


Figure 12:

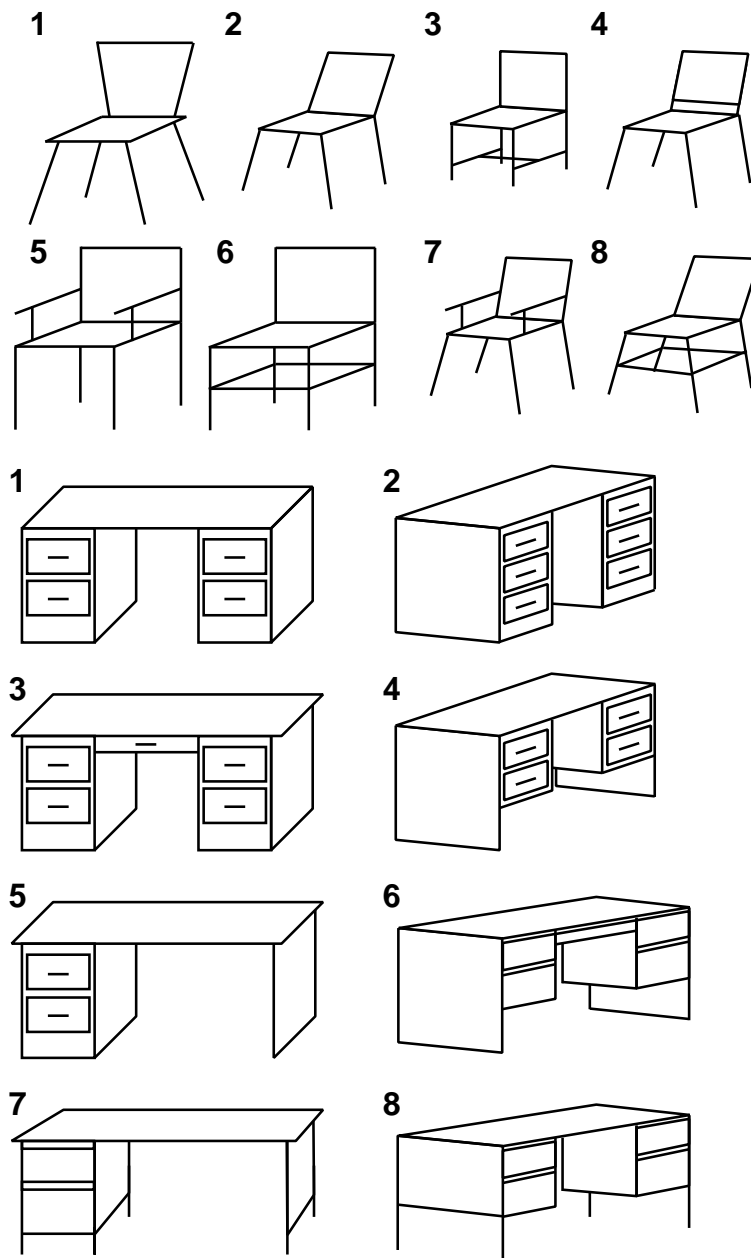


Figure 13:

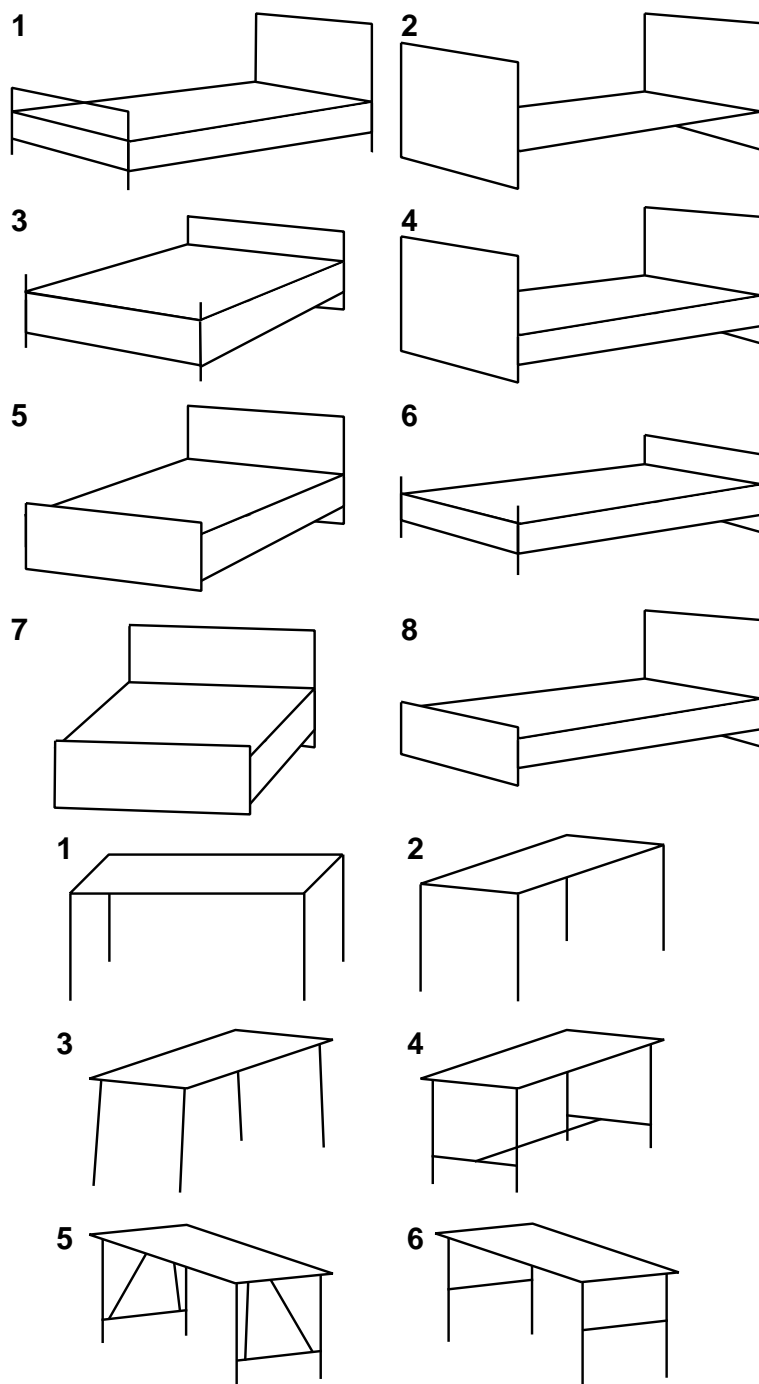


Figure 14:

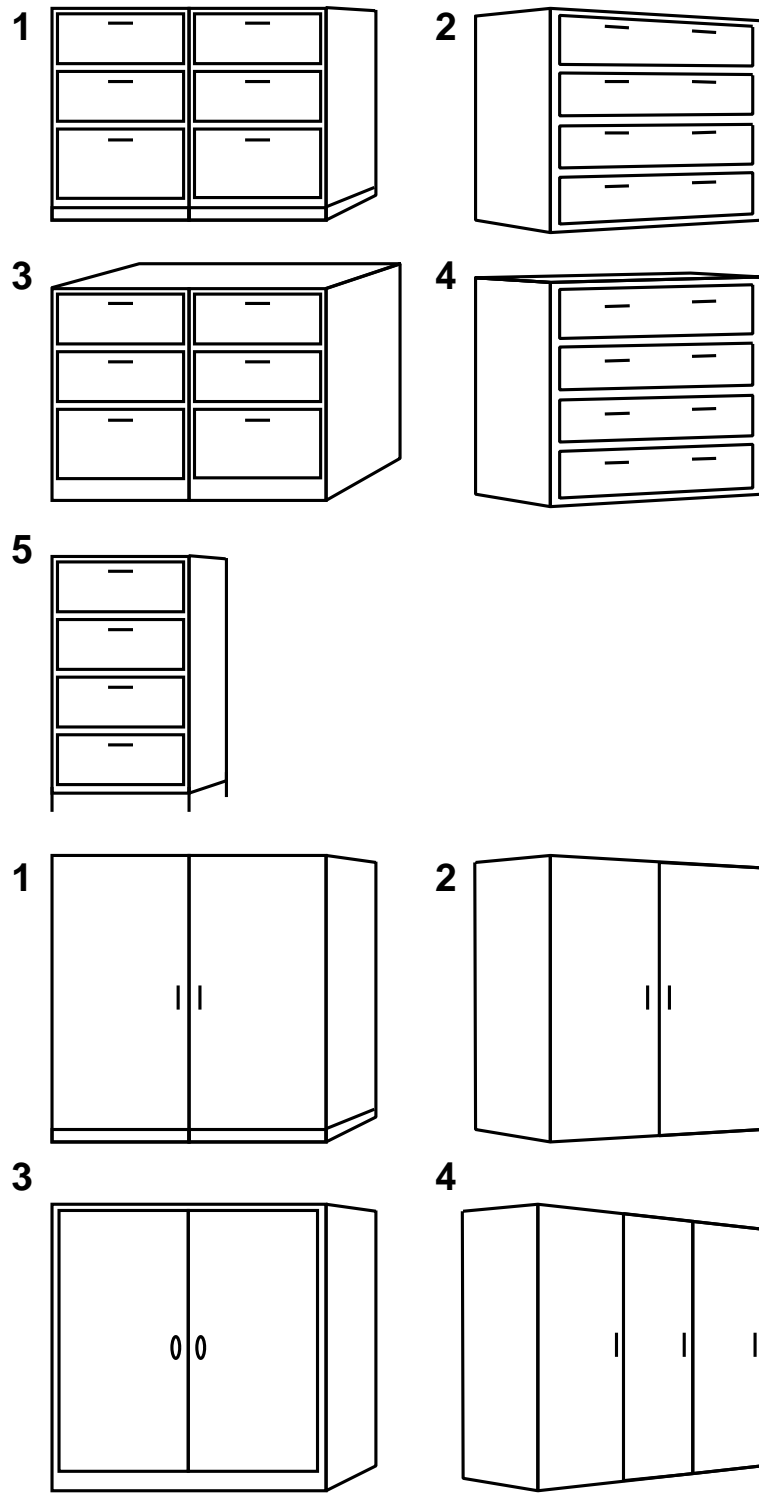


Figure 15:

## FIGURE CAPTIONS

Fig. 1. Structural variability in chairs. a and b show part-shape variability, c and d alignment variability, e and f part redundancy. What would be an efficient category representation?

Fig. 2. Vertex versus rectangle. a. The object contains five 3-line vertices (left, marked with grey circle) which form the basis for reconstruction of the object (right). b. The object has a larger desk plate due to part-alignment variability: 3 vertices disappear (left). Detection of parallelograms - trapezoids in perspective projection - and rectangles (hereafter called 3D rectangles) is more robust to this part-alignment variability (right).

Fig. 3. Detection of L features. a. Common noisy L detection. black: line, dashed black: extrapolation to corner point. b. Occasional accidental L detection between l1 and l2. c. L detection measure (see text for details) d. T junction detection (see text for details) and separation into two L features. e. Alignment variability creates L's with small legs (2), which are dropped.

Fig. 4. Formation of 3D rectangles starting with L features.

Fig. 5. Cubes (a), rectangle classification (b, c) and rectangle grouping (d, e). a. Representing canonical views of cuboids with rectangles as surfaces. b. Classification of rectangles into walls and tiles.  $\sigma$  indicates the measure for slant. c. Plates. For details on tilt ( $\tau$ ) and slant measure see text d. Grouping rectangles: Nesting types (shown only for frontal view). e. Grouping rectangles: Parallel types.

Fig. 6. Category essential features and their groupings. Only a few relations between features and or groupings suffice to achieve a high structural characteristic for a category.

Fig. 7. Object chair. 3D Rectangles are listed on the right with corresponding classification: type (wall/tile/plate), tilt, slant, 3D area. Angles are given in radians, the area size is in pixels times  $10^{-2}$ . 3D rectangle groupings are listed on the left. In this case, only the adjacent 3D rectangle pair is listed.

Fig. 8. Object desk. On the left side three blocks are listed: Adjacent groupings (1st block, 15 lines): Each line lists the 3D rectangles that are adjacent to the first one in the line. Folded groupings (2nd block, 4 lines): Each line lists the 3D rectangles that form a

foldable surface with the first one in the line. Parallel groupings (3rd block, 1 line): Lists the parallel and shifted groupings. (Caution: x and y axis are not drawn in proportion, which needs to be considered when interpreting angles).

Fig. 9. Object bed.

Fig. 10. Object table.

Fig. 11. Object drawers.

Fig. 12. Object closet.

Fig. 13. Objects of categories chair and desk.

Fig. 14. Objects of categories bed and table.

Fig. 15. Objects of categories drawers and closet.

## FOOTNOTES AND ACKNOWLEDGMENTS

Affiliation of Author: Caltech, Division of Biology, 139-74, Pasadena, 91125 CA

Acknowledgements: This work was supported primarily by the Engineering Research Centers Program of the National Science Foundation under Award Number EEC-9402726, and in part by the Swiss National Science Foundation. The author wishes to thank C. Koch for sustained support.

## References

- Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychol Rev*, 94(2):115–47.
- Binford, T. (1971). Visual perception by computer. In *Proceedings of the IEEE Conf. on Systems and Control*, Miami.
- Brooks, R. (1981). Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, 17:285–348.
- Canny, J. (1986). A computational approach to edge-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.
- Clowes, M. B. (1971). On seeing things. *Artificial Intelligence*, 2(1):79–116.
- Grimson, W. E. L. (1990). *Object recognition by computer: the role of geometric constraints*. MIT Press, Cambridge, Mass.
- Guzman, A. (1969). Decomposition of a visual scene into three-dimensional bodies. In Grasselli, editor, *Automatic Interpretation and Classification of Images*, chapter 12. Academic Press, New York.
- Huffman, D. (1971). Impossible objects as nonsense sentences. In Meltzer, M. and Michie, D., editors, *Machine Intelligence 6*, chapter 19, pages 295–323. Edingburgh University Press, Edingburgh, Scotland.
- Lee, C., Pong, T., Esterline, A., and Slagle, J. (1992). Kor: A knowledge-based object recognition system. In Shapiro, L. and Rosenfeld, A., editors, *Computer Vision and Image Processing*, pages 329–362. Academic Press, Boston.
- LIN, W., FU, K., and SEDERBERG, T. (1984). Estimation of 3-dimensional object orientation for computer vision systems with feedback. *JOURNAL OF ROBOTIC SYSTEMS*, 1(1):59–82.
- Lowe, D. (1987). 3-dimensional object recognition from single two- dimensional images. *Artificial Intelligence*, 31(3):355–395.
- Marr, D. (1982). *Vision*. W. H. Freeman, New York.
- Marr, D. and Nishihara, H. K. (1978). Representation and recognition of the spatial organization of three-dimensional shapes. *Proc R Soc Lond B Biol Sci*, 200(1140):269–94.
- Palmer, S. E. (1999). *Vision Science: Photons to Phenomenology*. MIT Press, Cambridge, Massachusetts.



- Palmer, S. E., Rosch, E., and Chase, P. (1981). Canonical perspective and the perception of objects. In Long, J. and Baddeley, A., editors, *Attention and performance IX*, pages 135–151. Erlbaum, Hillsdale, NJ.
- Robert, L. G. (1965). Machine perception of three-dimensional solids. In Tippet, T. e. a., editor, *Optical and Electro optical Information Processing*. MIT Press.
- Rock, I. and Palmer, S. (1990). The legacy of gestalt psychology. *Sci Am*, 263(6):84–90. (eng).
- Rosch, E., Mervis, C., Gray, W., and Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology*, 8:382–439.
- SHAPIRO, L., MORIARTY, J., HARALICK, R., and MULGAONKAR, P. (1984). Matching 3-dimensional objects using a relational paradigm. *PATTERN RECOGNITION*, 17(4):385–405.
- ULLMAN, S. (1990). 3-dimensional object recognition. *COLD SPRING HARBOR SYMPOSIA ON QUANTITATIVE BIOLOGY*, 55:889–898.
- Waltz, D. (1975). Understanding line drawings of scenes with shadows. In Winston, P., editor, *The Psychology of Computer Vision*. McGraw-Hill, New York.
- Wertheimer, M. (1958). Principles of perceptual organization. In Beardslee, D. and Wertheimer, M., editors, *Readings in Perception*, pages 115–135. Princeton, N.J.: Van Nostrand.



<http://www.springer.com/978-0-387-23468-7>

The Making of a Neuromorphic Visual System

Rasche, C.

2005, XI, 140 p., Hardcover

ISBN: 978-0-387-23468-7