

# Structural Descriptions using Feature Context

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

## Abstract

Visual object representations are often thought of a fixed set of rigid parts or features. We show in a computational study, using furniture objects depicted in line drawings, that another valuable source of structural information is feature context. Employing such context evaluation during feature extraction, leads to a rich object description by surfaces and silhouette structures. This redundant description is useful and necessary because the same object can display different conspicuous features in different settings. Category-specific features are loosely expressed in order to match the structural variability existent within categories.

## 1 Introduction

Human visual object recognition is a multitasking process: when we see a novel object we generally assign it to a basic-level category like car, chair or tree [Rosch et al., 1976]. When we are asked to distinguish between cars, we likely assign them to subordinate categories like sports car, van or limousine. If we had to point out our favorite car or the one we drive, then an identification process is at work. These different recognition processes may require different mechanisms or different features for description. Here we solely focus on the basic-level categorization task.

The basic-level categorization process bears multiple aspects [Palmer, 1999] (see also figure 1). One aspect is its viewpoint independence: we are able to recognize an object from different viewpoints despite the different 2D appearance of the object's structure for any given viewpoint. The viewpoints of an object can be roughly divided into canonical and non-canonical [Palmer et al., 1981]. Canonical viewpoints exhibit the object's typical parts and its relations, like the chairs seen in the lower row of figure 1. Non-canonical viewpoints exhibit only a fraction of the object's typical parts and are less familiar to the human observer, like the chairs seen in the upper row of figure 1 (the chairs are seen from the back). Many computational approaches have been developed that are able to deal with

this recognition aspect very well: These systems are able to exactly determine the pose of an individual object [Robert, 1965, Brooks, 1981, Lowe, 1987, ULLMAN, 1990, Grimson, 1990]. These systems are mostly developed for robotic vision, which requires the identification of industrial objects (like machine parts) from any viewpoint and sometimes in a cluttered background. Psychological object recognition research has also primarily focused on the viewpoint independence aspect (e.g. [Tarr and Bulthoff, 1998]).

Yet, another aspect of basic-level categorization is its variability independence: humans are able to recognize objects although each category instance is structurally somewhat different from one another. For example, chair number 1 (in figure 1) has legs that align exactly with the corner of the seating area. So does the backrest. In contrast, in chair number 2 the legs and the backrest do not align exactly. Chair number 3 has additional structure like the arm rest and leg support. These examples express only marginally the variability existent in real world objects, and looking at a furniture catalog vividly demonstrates the immense structural variability. The previously cited systems, managing viewpoint independence so well, are not able to deal with this variability. They were designed to perfectly identify a specific object. These systems therefore rather represent identification systems. In turn, recognition systems that have tried to deal with structural variability are often weak at representing viewpoint independence [SHAPIRO et al., 1984, Lee et al., 1992]: their object domain has several structurally different objects but seen from the exact same, canonical viewpoint. Our study aims at describing objects from different but still canonical viewpoints. Furthermore, in these systems objects are often described the way we humans understand objects, namely by parts [Guzman, 1971, Marr, 1982, SHAPIRO et al., 1984, Biederman, 1987, Lee et al., 1992]. For example, a chair is described by its legs, seating area and backrest. Yet, this is not the only possibility to describe or recognize an object. For example, we can also categorize objects based solely on their silhouettes [Rosch et al., 1976]. Silhouettes are *not* bound to parts, but are descriptors that typically span contour fragments of several parts. We will point out characteristic silhouette structures for furniture objects.

**Object description by shapes and parts.** Guzman was the first to thoroughly discuss the description of basic-level objects [Guzman, 1971]. In his examples, an object is described by individual shapes: For example, a human body is described by a shape for a hand, a shape for a leg, a shape for a foot and so on. His shapes were 2D features: shapes specified in two dimensions. A number of other studies describe objects by a set of 3D parts, each part being a 3D volume. These studies are presumably influenced by work on

simple building blocks: Guzman developed a program that is able to segment a polyhedral scene into its building blocks (mainly cuboids) [Guzman, 1969]. His study trailed a host of other studies refining this type of scene analysis [Clowes, 1971, Huffman, 1971, Waltz, 1975]. Binford came up with a system that measures the depth of a scene by means of a laser-scanning system [Binford, 1971]. The scene consisted primarily of cuboids and cylinders. Such work on simple building blocks presumably led to the belief that basic-level objects are best described by a set of voluminous 3D parts. Marr represented animals by cylinders [Marr and Nishihara, 1978]. Shapiro et al. described furniture by a set of volumetric parts, classified into sticks, plates and blobs [SHAPIRO et al., 1984]. Pentland described natural objects like trees with superquadrics like diamonds and pyramidal shapes [Pentland, 1986]. Biederman proposed an even larger set of parts, like cylinders, cuboids, wedges and more to describe objects [Biederman, 1987]. The problem with a fixed set of parts is that it can not keep up with the variability of real-world objects. For example a chair’s legs can be of cylindrical or cuboidal shape: we call this part-shape variability. It is thus futile to rely on part shapes as a building block for category representations. The object as a whole has a volume or a certain structure in 3D space, but both, the overall structure of the object as well as its detailed voluminous parts are variable. Another short-coming of volumetric approaches is, that it is computationally very expensive to extract the volumetric information of each single object part. In particular, given the astonishing speed of human recognition [Biederman, 1972, Potter, 1975, Thorpe et al., 1996, Schendan et al., 1998], one can assume that such detailed processing or analysis is simply of secondary importance. It is cheaper and easier to interpret merely 2D contours, as Guzman proposed it. For example, the shape and orientation of surfaces gives us hints on the pose of the object without assigning a 2D coordinate to a 3D coordinate system. They would be merely interpreted as 2.5D features, to use Marr’s terminology [Marr, 1982].

**A loose category representation.** Given the large amount of variability, it is difficult to envision a category representation made of a fixed set of rigid features. Our proposal is to view a category representation as a loose structure: the shape of features as well as their relations amongst each other is to be formulated as a loose skeleton. One could argue that loose representations are prone to recognition errors. We think, this is exactly what happens when we look at visual illusions, like the impossible staircase, trident or triangle (<http://www.illusionworks.com>). Immediately, we have a sense of the object’s rough structure. It is only after closer inspection, namely a visual search, that we detect inconsistencies with our world-knowledge. We interpret this as follows: there is enough structure

in an image that can give us an immediate sense of the object category, but the object’s detailed structural relations are to be explored in a visual search. Recognition can thus be understood as a process in which visual features point toward categories triggering in turn specific object knowledge. Such knowledge could consist of a set of rules that relates objects parts in a meaningful manner, for example a chair’s leg has to be connected to the seating area. In this work, we are concerned with illustrating characteristic, (2-dimensional) visual descriptors that are valuable pointers to a category. Guzman has also proposed that a representation needs to be loose [Guzman, 1971]. He developed this intuition by reflecting on how to recognize an object that has deformations like bumps or distorted parts. He termed the required representation as ‘sloppy’.

**Category variability and choice of objects.** We intuitively classify structural variability into three types: part-shape variability, part-alignment variability and part redundancy. Part-shape variability is the above mentioned variability of single parts of an object. Part-alignment variability is the looseness in the alignment of two or more parts (also mentioned previously). Part redundancy is the object’s structure that is not necessary for its successful categorization of the object, as for example the arm rest of a chair. Part redundancy mainly occurs with complex objects possessing a lot of functional parts. Such large variability requires an enormously flexible feature extraction process. In order to simplify our simulations, we reduce the amount of variability as explained in the next paragraph.

Our work focuses on furniture objects depicted in line drawings (figures 13-15). The categories are chair, desk, bed, table, drawers and closet. The line drawings only outline the rough structure of objects. The detailed voluminous structure of each single part, like voluminous shape of a chair leg, is omitted. We chose the following two main restrictions on part-shape variability: 1) Only straight lines are used. 2) Only rectangular surfaces are used. The drawn objects show part-alignment variability and sometimes part redundancy. The objects are shown primarily in canonical views, yet not from fixed viewpoints. This is obviously a vast oversimplification of the existent variability in real-world objects, but the purpose of the simulations is to uncover an efficient representation for a reduced structural variability that is useful for extension to a larger variability.

**Representing a cube.** By design restrictions, the furniture objects in our line-drawings often possess cuboidal structure (which is also true for many real-world furniture objects). It is therefore fundamental to understand how we represent a cube. Figure 2 shows different approaches. In figure 2a the cube is represented using vertex features, as the ones used

in Guzman’s and other’s work [Guzman, 1969, Clowes, 1971, Huffman, 1971, Waltz, 1975, Biederman, 1987, Lee et al., 1992]: Vertex features are classified according to the number of intersecting lines and their approximate geometry. From the arrangement of classified vertex features one can conclude to surfaces and consequently to volumes.

An alternative approach is to represent the cube by surfaces, as shown in figure 2b. There have been a number of suggestions on how to detect surfaces in real-world scenarios: Marr proposed to reconstruct the surfaces of objects using multiple cues like edges, luminance, stereopsis, texture gradients and motion [Marr, 1982]. Barrow and Tenenbaum focused on edges and luminance only [Barrow and Tenenbaum, 1981]. A simpler and straightforward idea is to interpret merely the geometry of the surface contours and determine their orientation in space, like Lee and Fu did [Lee and Fu, 1983]. They used surfaces to represent cars and industrial objects; we used surfaces to determine the perspective of a room scene [Rasche, 2002a]. Expressed as such, a cube consists of three quadratic surfaces, each made of 4 L features. A representation by surfaces is preferred over a representation by vertices (of three intersecting lines or more) because of the existent part-alignment variability in real-world objects: their surfaces do not always align neatly [Lee and Fu, 1983]. However there is an incompleteness regarding a mere surface extraction and object representation (as in figure 2b): the lack of contextual information. For example, the structure shown in the inset of figure 2b contains a cube according to a representation not accounting for context. A human observer however tends to judge this structure as a pattern containing a cube structure inside. Another example is the rectangle: it outlines windows, doors, pictures, the front of cabinets and drawers. But many of these rectangular objects have a typical surround (or context): The rectangle of a drawer is often next to other rectangles of neighboring drawers; In contrast, the rectangular picture on the wall is rather alone. Thus, determining the immediate context of a feature or shape can give us already a hint toward its category. If one starts the context analysis for L features, followed by their classification and grouping according to common regularities in our world, one arrives at two different higher features (or structures) for the cube: surfaces and a silhouettes (figure 2c). This idea will be expanded next.

**Context evaluation of a cube.** We start with finding all possible L features of the cube: it has 15 (figure 3a). Vertices of more than 2 intersecting lines are not considered. The context of each L feature is determined by distinguishing the ‘inside’ and ‘outside’ of a L feature’s corner (figure 3b): the inside is the area outlined by the smaller angle, the outside is the area complementary to the inside area. Looking at these two areas, we can divide L

features into the following three classes:

1) L features that have an additional line in the outside but no line in the inside. Stated generally, the L feature contains some structure outside, but is free of structure inside. We call these *In* features. The cube has 12 In features: numbers 1 to 12.

2) L features that have an additional line inside but no line in the outside. Stated generally, the L feature has some structure inside, but no structure outside and we therefore call them *Out* features. The cube has 6 Out features: numbers 3, 8, 10, 13, 14 and 15.

3) L features that have no lines around their corner, neither in their inside nor in their outside. They are simultaneously In and Out features and we therefore call them *Lone* features (lone of any other structure in its surround). The cube has 3 Lone features: numbers 13, 14 and 15.

As implied with this classification, we are not so much interested in the exact structure around a L feature, but mainly in finding its empty side. To progress toward complex, global features, we start to connect neighboring L features to three-line polygons (pg3). Pg3s with their classified, two corners are already an enormously valuable source of structural information that we will exploit for object description. Figure 4 shows all possible cases regarding their classified corners. Out of this set of cases there is only a fraction that makes sense in our typical environment. This will be elaborated as we proceed. For the cube example, it is sufficient to realize that we obtain surfaces and silhouettes by two simple rules: 1) If we connect In and Lone features pointing to each other (in a circular manner), we obtain the three surfaces. 2) If we connect Out and Lone features, we obtain the silhouette. In rough terms, with In and Lone features we find surfaces embedded in something. With Out and Lone features we find the silhouette. (Remember, that Lone features have a double role as In and Out features.) We call this type of context understanding 'feature context', as opposed to object or scene context, which refers to the semantic context of neighboring objects.

**Surface classification and grouping.** We are interested in describing the objects in their canonical views because in every day life furniture objects appear mainly standing and in a view as depicted in figures 13 to 15. An exception are chairs and tables, which we frequently view from different sides, but mostly still standing. In these standing poses, surfaces have characteristic geometries, which we classify into certain types. In the cube example, surfaces A and B are both of type *wall*, because they represent standing surfaces. Surface C is of type *tile*, because it appears flat (lying in 3D space) to the observer. Each classified surface tells us something about the orientation of cube. Another typical feature of the cube is the

*folded surface*. It consists of two surfaces forming a folding (figure 5k). The cube has three such folded surfaces, one for each surface coupling: A,B; A,C; B,C.

In the method section we explain how we evolve pg3s and other commonly occurring features, using computer vision methods. We call this feature evolvment *bottom-up* process. In the result section, we uncover features that are distinctive for one category whereas the emphasis lies on silhouette features. Generally, a couple of features suffice to distinguish a category from the others. In some sense, the present category representations can be viewed as feature lists. The recognition system is a matching system: each category-specific structure is matched against the output of the bottom-up process. We call this matching procedure the *top-down* process. We use these computer vision methods for convenience. The human visual system certainly uses a different architecture for recognition, but the present study focuses on the visual descriptors using context.

## 2 Methods

**Images.** Images were drawn in a vector-based drawing program called 'tgif' available under linux (<http://bourbon.usc.edu:8001/tgif>). Image sizes range from 250x250 to 400x600. Lines are 1 pixel wide.

**Contour extraction and line formation.** Contours are extracted using an algorithm connecting points, which have only 2 neighbors. Due to occasional aliasing, we also connect points with 3 neighboring points. This procedure breaks up contours at junctions with three or more intersecting lines. Contours at L corners are often continuous. We therefore break the contours into its straight line pieces using Lowe's algorithm [Lowe, 1987], which recursively cuts contours at their maximum deviation from the line connecting the contour's endpoints.

**Determining a line's neighborhood.** An image-sized matrix is generated with the extracted lines written into it, whereas each line point corresponds to its line list index. For both endpoints of each line, a square neighborhood of the line index matrix is extracted to collect the possibly interesting neighboring lines. The side length of the square neighborhood is half of the line length and is a square for reason of simplicity. For each line, the structural relation to its neighboring lines is determined: We distinguish between T, L and B features (B for branch). Figure 5a shows a schematic example. Line number 1 is the line

being evaluated: line numbers 2 to 7 are determined as neighbors. Lines 2 and 3 form T features with line 1; lines 4,5 and 6 form L features; line 7 forms a B feature with line 1, it is branching away from line 1. As just implied, the neighboring lines are labeled according to their relation to the evaluated line. The labeling is reversed when a neighboring line is evaluated and line 1 appears as a neighbor.

**Determining line context.** In our line drawings we represent legs and handles of furniture objects by lines (of width equal 1). We therefore have line endings with no neighboring lines: Their endpoint context is empty. We call such line ends *end-stopped*, in accordance to the end-stopped lines found in the visual system [Hubel and Wiesel, 1962]. A furniture leg thus has one end-stopped line end, a handle has two. Such end-stopped information is a valuable part of the silhouette analysis.

**Determining L feature context.** The line and L context analysis is shown schematically in figure 5b. In a first step, we eliminate uninteresting features. Firstly, if a line forms several T features at the same endpoint, then only the closest one is used further. In our example, the T feature formed by lines 1 and 2 is dropped, and only line 1 and 3 form a T features (which later is split into two L features). Secondly, L features that can not be classified into either In-, Out- or Lone features are neglected: for example the L feature formed by line number 5 and 1 can not be classified and is therefore eliminated from the list of L features; only lines 4 and 6 form useful L features with line 1 (assuming that line 7 is not to close in the neighborhood of L feature formed by 1 and 6). B and T features are not used for representations: they are split into two L features, whereby the intersected line is kept, see inset in figure 5b. It would be desirable to leave the intersected line away, but the same intersected line can be additionally intersected at other locations. That will create some L features having overlapping legs, which is not in accordance with our idea to analyze feature context. We have not attempted to resolve this insufficiency because it would require a major restructuring of our programs and because it has only minor effects on recognition. Hereafter, this is referred to as the T-split insufficiency.

To determine the context of a L feature, one could analyze the stored structure at both line endpoints, which however is very intricate for some line intersections. We therefore choose to check the context using the line index matrix. Because evaluating the entire context area is time consuming, we merely extract line indices along a circle of fixed radius (25 points) with center equal to the corner of the L feature. This simple fixed-radius context is not complete and will not properly determine the context of all L features. In particular part-



alignment will occasionally cause insufficient context analysis. But the measure is sufficient to demonstrate the usefulness of context-evaluated features. Deficiencies will be pointed out in the result section.

**Pg3 formation and classification.** To find pg3 features (3-line polygons), the list of L features is browsed for couples of L features sharing a leg (figure 5c). Pg3s are firstly divided into *Ulike* and *Zlike* features according to the direction of their outer legs. *Ulike* features have both legs (the outer lines) on the same side of the middle line, *Zlike* features have their two legs on different sides of the middle line. Each division contains six possible combinations of coupled (context-evaluated) L features. Figure 4a and b show the combinations for one orientation with an angle of 90 degrees for both L features. The polygon features should be imagined for varying orientations and angles. Each combination occurs with different frequency and often outlines a specific structure. We sketch the usefulness of the structures:

*Ulike* features can be divided into two groups: a) ii, il and ll combinations point toward surfaces, because they have no structure inside the U-shaped area. We therefore call these *Usurf* features. b) oo, ll and ol combinations point to a piece of a silhouette, because they have no structure outside, but some inside. We therefore call these *Usilh* features. The ll combination is a *Ulike* feature outlining both, a surface as well as a silhouette: It therefore indicates a fragment of a surface being alone in space. The oi combination has no interpretation value, at least not in furniture objects: it simply does not describe structure in a meaningful way. We therefore ignore oi combinations.

*Zlike* features offer much fewer interesting combinations than *Ulike* features: the oi and li combinations represent silhouette pieces of a folded structure like chair or bed. This is an example, in which an *In* feature is part of a silhouette and not a surface. The ii combinations is a common *Zlike* feature, but not interesting because it describes two surfaces touching each other, which is generally represented by two *Usurf* features sharing their middle line. The combinations oo, ll and lo never occur in our line drawings. In summary, from six *Ulike* and *Zlike* features, only five and two respectively, are valuable for object description (figure 5d).

**Geometrical and contextual conditions.** It should be emphasized that structural description now has two distinct components: Geometry and context. An object does not only have characteristic features of specific geometry, its geometrical conditions, but also specific context, its contextual conditions. The differentiation of pg3 features into *Ulike* and *Zlike* features is a first step towards specifying characteristic geometries. The accompanying contextual information of its corners establishes already category-characteristic traits. In

the following we will list features with specific geometrical and contextual conditions that appear frequently in furniture objects.

**Bridge and Arch features.** Many objects possess legs forming, thus a silhouette structure that corresponds to an *Uii* feature (Unlike feature with *ii* context combination) with its open side pointing downwards. In this case, the *Usurf* feature is actually not a surface indicator but rather a silhouette pointer. Desks also have such a feature formed by the space where one puts the legs underneath. This space is generally outlined by the desktop’s chest, its plate and its third structure supporting the other side of the plate (which can be another chest, a leg or a wall). The *Uii* opening formed by legs has generally end-stopped line endpoints. We call this feature *bridge*. The *Uii* opening formed by the desk may have one end-stopped line endpoint or not. We call this feature *arch*. Both features are found by searching for *Uii* features that have two outer legs pointing approximately downwards, which in turn are browsed for those having end-stopped line ends to find the bridge feature (figure 5f). The tolerance for pointing downwards is large, 25 degrees, because some chair legs are not exactly vertical, but spread apart (chair numbers 1,2,4,7 and 8). This is an example of a flexible feature that is able to capture some structural variability.

**Silhouette pointers to cube.** We do not reconstruct the entire silhouette of any object, but we filter out characteristic *Usilh* features that - put together - can very quickly point toward the silhouette of the entire object. Two such *Usilh* features are the *USleft* and *USright* feature. They have a vertical middle line and two legs pointing both either to the left or the right respectively (figure 5g). One *USleft* and one *USright* outline a feature called *cublk* (cuboid-like) if they form a polygon resembling a cuboid silhouette: More specifically, if their outer legs point at each other and if their endpoints lie proximal.

**Rectangle formation and classification.** To search for surfaces (closed polygons) we could simply ring-connect *Usurf* features. Because we have chosen only rectangles (in 3D space) as surfaces by design, we directly search for such 3D rectangles by coupling *Usurf* features as shown in figure 5e. The evolved shapes are then classified: 2D rectangles simply represent frontal views of rectangular surfaces (e.g. surface A, figure 3), parallelograms and trapezoids represent rectangular surfaces, tilted and slanted in 3D space (e.g. surfaces B and C, figure 3). Parallelograms and trapezoids are further classified into the three basic types *tile* (1), *plate* (2) and *wall* (3 and 4), according to their geometry in the 2D coordinate system (figure 5h).

Wall: 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. The 2D rectangle is also treated as of type wall. Walls are sub-classified into *right* (3) and *left* (4) according to the slope of their parallel sides. A negative slope is interpreted as a surface being right of a volume (e.g. surface B in figure 3) a positive slope is interpreted as being left of a volume. Walls are also sub-classified 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.

Tile: 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. Or stated more generally, it appears to the observer as a lying rectangle in 3D space, like the top plate of a desk or table.

Plate: If the parallel sides are neither vertical or horizontal, then we classify the 3D rectangle as a plate. In other terms, it is a tile seen from a corner viewpoint. 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. For asymmetric trapezoids we take the larger angle between the side and the orthogonal of the parallel sides. For plates we take a very crude measure: we measure the angle of the bottom L feature and subtract  $\pi/2$ .

**Rectangle grouping.** 3D rectangles are grouped according to commonly occurring patterns. The list of 3D rectangles is searched for 'adjacent' groupings, in which two 3D rectangles share one or more sides - either fully or partially - or are parallel and close. A 3D rectangle can have several adjacent 3D rectangles, which are grouped into a list. Adjacent 3D rectangle groupings are classified into 'nested', 'parallel' and 'folded'.

Nested: If two 3D rectangles share two or more sides, then the smaller one is nested within the larger one. We call the larger 3D rectangle the 'principal' rectangle. There are two frequently occurring cases, which we call 'sliced' and 'embedded'. In the 'sliced' case, the smaller rectangle touches two opposite sides (figure 5i, 1). In the 'embedded' case, three sides are adjacent (figure 5i, 2). Nested groupings are useful for representing drawers for

example.

**Parallel:** Two 3D rectangles are parallel if all sides of one rectangle are parallel to the corresponding sides of the other, but are distant (they are not adjacent). We distinguish two cases: aligned and shifted. In the aligned case, two opposing parallel sides of each 3D rectangle are collinear and they likely share the same plane (figure 5i,3). This grouping occurs for desks with two chest of drawers. In the shifted case, no such collinearity exists, and the 3D rectangles are likely to be in two different planes in 3D space (figure 5i, 4). This grouping occurs for the surfaces (head- and footwall) encasing the mattress of a bed.

**Folded:** Two adjacent 3D rectangles are considered as a folded surface when they differ in their geometrical classifications (left wall, right wall, plate and tile).

### 3 Results

Throughout the following category descriptions, an object is firstly described in terms of its surface features, followed by a description of silhouette features. The list of features should be read bearing for example a probabilistic view in mind: a feature can be typical for a category and occur with a high likelihood, but does not have to, due to the existent structural variability. Most of the features are shown in figure 6. The silhouette features should be understood with the following scenario in mind: One enters a dark room and is able to see only the silhouette of furniture objects against a bright background: We will circumscribe parts of such silhouettes.

**Bed.** Surfaces: A bed consists of four surfaces that we term 'plate' (the lying surface), 'head' and 'foot wall', as well as 'side'. The surface plate is of rectangle type plate. This geometrical condition alone is not very distinct: many other objects possess a surface of type plate, like table, desk, some drawers. The bed's plate has a characteristic context: all its L features are of type In, because in their outside is structure from the foot or head wall. Adding this contextual constraint returns us plate features for only beds and chairs. (The chair structure is basically the same as the bed structure apart from their geometrical proportions.) The other three surfaces are of type wall. There are 4 folded surface couplings amongst these surfaces, some of which are very distinct for the bed category. The most distinctive rectangle grouping is the parallel, shifted grouping of the head and foot wall (figure 5i,4). It is only existent in the bed category.

**Silhouette features:** The 3D arrangement of the plate and head wall possesses a characteristic Zsilh feature, which outlines one corner of the head wall and the folding corner

between the head wall and the plate. We call this the 'lean' feature, (because one can lean against the head wall when sitting on the bed). Its middle line is vertical and does not change its orientation for different, standing object poses. Using only this geometrical constraint returns us lean features for beds, chairs and desks. To further select we add contextual constraints as follows: The headwall's corner is a Lone feature, the folding corner is an In feature; we check only if either corner has the appropriate context. That returns only lean features for beds and chairs.

Another typical silhouette feature is outlined by the head wall: it is a surface standing alone in 3D space and its top contour forms a Ull feature. A third characteristic silhouette feature is formed if the head and foot wall are both high-rising walls extending the plate surface: then an area is formed corresponding to an Uii feature whose opening points upward (not shown in figure 6). In other terms, it is a flipped arch feature and we call it *pit*.

Figure 7 shows the features extracted from bed number 4. This bed contains five Zsilh features. Two are lean features formed by the high-rising head and foot wall (L features 3 and 18, as well as 8 and 23 respectively). One Zsilh feature is wrapped around the lower right corner of the headwall (assumed to be 3D rectangle number 3) made of L features number 5 and 25; the remaining two are accidental due to the T-split insufficiency (method section, determining L feature context) and are formed by the footwalls' upper right corner, L feature number 7, and the L features (14 and 10) formed by the lines branching away from the vertical footwall contour. L features 18 and 23 form a pit feature. The head and footwall form the shifted, parallel grouping of two rectangular surfaces that is unique to the bed category. L features 8 and 4, as well as 7 and 3, form each the downward pointing Ull feature.

L feature number 2, the lower right corner of the foot wall, is an example for the deficiency of the simple circular context measure we use: The L's outside is to a large part empty, but has not been classified as such, because the lower line of the side wall is in the range of the circular measure. In other terms, it is large alignment variability that occasionally causes insufficient context determination.

**Chair.** Surfaces: The seating area and the backrest are two surfaces that form the same characteristic folded surface as the one for the object bed formed by its head wall and its plate. Because the seating area is actually only a small region on the 2D plane, we ignore it for our present implementation.

Silhouette features: Because of the existence of the folded surface structure, the object chair has the same characteristic Zsilh feature, the lean, as the bed and it is here formed

by the backrest and the seating area. Another characteristic Zsilh feature is formed by the contour spanning the backrest, the seating area and one leg: it is a oi context combination and we call this the *seat* feature. Its outer lines are defined only approximately vertical, because both the backrest as well as the leg show some variability in their orientation. This geometrical condition is already highly specific and the seat feature as such is additionally found only in one desk, one drawer and three beds. Using a contextual condition, for example either corner has to correspond to the appropriate context then the seat feature is only found in chairs and two beds (3 and 6). If a chair possesses an armrest like in chairs 5 and 7, then the seat feature is formed by the contour spanning the backrest, the armrest and its vertical support stick (number 2 in figure 6, Chair). This is somewhat accidental but nevertheless a characteristic feature of chairs. Similarly, in those chairs the lean feature is actually formed by the armrest instead the seating contour.

The backrest possesses the same characteristic Ull feature, as does the head wall of the bed.

The bridge features of a chair are generally formed by its legs and the contours of the seating area. Bridge features are formed by any two legs dangling in space and do not need to represent two physically adjacent legs. For chair numbers 6 and 8, the bridge features are formed by its legs and the leg support structure instead the seating contour: they are low-rising bridge features.

The bridge features of a chair form a structure that we call *double bridge*: two closely spaced bridge features with a minimal angular difference between their middle lines (e.g.  $\alpha > 5$  degrees). All chairs possess a double bridge feature, except chair number 3, whose leg support does not lend to such a formation. For chairs with low-rising bridge features, the double bridge is formed by these and is thus a low-rising double bridge feature. Double bridge features are also found in other objects standing on legs: in one bed (number 1), one desk (number 7, left chest), one drawer (number 5) and most tables.

Another characteristic silhouette feature is the geometrical arrangement of the end-stopped endpoints corresponding to the chair's legs: they span a parallelogram or triangle, if all four or three endpoints (legs) respectively, are visible. The feature was not used in this study.

Figure 8 shows the features extracted for chair number 4. Four bridge features are found (see list on the left): for example L features 1 and 11 represent the bridge formed by the two frontal legs; L features 1 and 22 represent the bridge feature formed by the right frontal leg and the left distal leg (in the corner). Each of these two bridge features forms a double bridge feature with the bridge features made of L features 2 and 4.

Three Zsilh features are found. L features 6 and 14, as well as 12 and 14, correspond to the lean and seat feature respectively. L features 6 and 5 correspond to the downward pointing Ull feature of the backrest.

The redundant structure in chairs numbers 3 and 5 to 8, like the leg support or arm rest, is not properly reconstructed due to the simple context measure.

Because the general structure of beds is virtually the same as the one for chairs - apart from their proportions -, their feature lists overlap to some extent. Bed number 1 almost contained all chair features.

**Closet.** Surfaces: There are only two surfaces: one formed by the doors and one by the side wall, which together correspond to a close-up (high-rising) view of a cuboid, namely two trapezoids. We make use of only the door surfaces. Their geometrical condition is defined as a principal 3D rectangle of type wall, containing nested 3D rectangles of type sliced or embedded, which in turn are of high orientation. Drawers have these structures as well. A contextual condition is therefore added, specifying that the nested (door) rectangles do not contain nested rectangles themselves, which would indicate that these are rather drawers than doors.

Silhouette feature: The silhouette of the two trapezoids, can be well captured by the cublk silhouette.

The object shown in figure 9, closet number 2, has two doors. The two doors are extracted as two separate 3D rectangles (numbers 2 and 3) and one large one (number 1). The large one is the principal rectangle containing the two doors as nested rectangles. The principal rectangle is actually formed due to the T-split insufficiency, which turns out to be beneficial in this case. If we had only extracted the two door rectangles, we had to group them later to form an entire surface. The cublk silhouette is formed by L features 3, 4, 5 and 6.

**Desk.** Surfaces: Desktops are generally made of a plate and one or two chests containing drawers. The geometrical conditions for the plate are defined as a 3D rectangle of type tile or plate, with a maximum tilt and a minimum slant. If we do not constrain tilt and slant, the backrests of chairs are interpreted as plates as well, because they are slightly tilted and slanted in 3D space. The plate's context is defined to consist of three In features and one Lone feature. This returns plates for all desks, tables, one drawer (number 3).

To represent the chest, we merely describe its frontal side containing the drawers. Its geometrical condition is defined as a principal 3D rectangle of type wall, of square shape or high orientation, containing a nested 3D rectangle of type sliced or embedded. This structure

is also found in closets and drawers. A folded surface is formed by the principal 3D rectangle of the chest and the plate, which is already a distinct structure for the desk category.

**Silhouette features:** The previously mentioned arch feature exists only in desk numbers 1,3,5,6 and 8. In desk numbers 6 and 8, the arch feature outlines an accidental feature formed due to the T-split insufficiency.

Desks, whose chest contour touch the ground, like numbers 1, 2, 3 and 5 (meaning not standing on legs), possess USleft and USright features with which one can form almost a cublk feature. They have not been exploited for this category.

Figure 10 shows desk number 2. 3D rectangle number 1 corresponds to the plate feature. Each of the 6 drawers is embedded in several principal 3D rectangle due to the T-split insufficiency. The chest structure is therefore found 18 times. The folded surface between plate and chest is found between both chests (3D rectangle numbers 4 and 5). The three extracted Zsilh features, of which one is accidental due to the T-split insufficiency, are of no representational value. 3D rectangles number 4 and 5 form the shifted, parallel grouping that is unique to this category (figure 5i,3)

**Drawers.** Surfaces: Drawers are similar to closets. They contain 2 to 3 surfaces depending on the height of the drawers. We only use the front surface, containing the drawer pattern, for representation. Its geometrical condition is defined as a principal 3D rectangle of type wall containing 3D rectangles of type sliced or embedded and wide orientation. This condition is only subtly different from the one for desk chests.

**Silhouettes features:** The drawers silhouette can be simply described by a cublk silhouette.

Drawers number 2 is shown in figure 11. 3D rectangle number 1 is recognized as the drawer front. L features 11, 10, 8 and 3 form the cublk silhouette.

**Table.** Surfaces: A table has only one surface, that has the same geometrical and contextual conditions as the desk's top surface.

**Silhouette features:** The bridge features form double bridges. Bed number 5 does not form such bridges because one bridge-L feature has not been extracted due to the simple context measure.

Figure 12 shows an example. Bridge features are formed by 3 and 4, 3 and 13, 4 and 12, 7 and 9 (see also list on left side of object). Two double bridges are formed between bridge feature 7/9 and 3/13 as well as 3/4.



## 4 Discussion

**Value of context.** Much object recognition research has attempted to describe structure as a geometrical arrangement of a limited set of features or parts ([Palmer, 1999], figures 2a and b). We have now illustrated that another valuable source of structural information is feature context, or in other terms, the space around features. Our recognition system starts with a context evaluation and classification of line endpoints and L features. L features are then integrated to pg3 features of which only a fraction (7 out of 12) make sense in our object world. That fraction is a set of potent visual descriptors of silhouette structures or pointers to surfaces.

In an early simulation study, we represented objects solely by surface features and some silhouette features, meaning *without* accounting for context [Rasche, 2002b]. Because of this lack of context, many category-specific features were accidentally found in other categories and we had to specify a number of geometrical constraints which mainly related to standing poses and geometrical proportions in order to get the categories distinguished. In our present study, including context information, we could drop many of these geometrical conditions in exchange to contextual conditions. Contextual conditions have two advantages. Firstly, features become much more specific because they contain contextual information in addition to their geometrical information. Consequently, the features are much less accidentally detected in other categories. Secondly, contextual conditions are independent of the object’s pose: they only encode space but are not tied to the geometrical orientation of its feature. Indeed, a next step could be to attempt to formulate category representations that are suitable for recognizing less canonical views, for example, furniture objects that are in lying position instead of their regular standing pose.

**Properties of recognition evolvement.** The choice of which features are to be evolved in the bottom-up process (figure 5) and which ones are to be evolved during the top-down process (figure 6) is arbitrary. Some of the features, evolved in the bottom-up process, are obvious pointers toward a category, like the cublk silhouette or the lean feature. The bottom-up process therefore has category-specific traits. An object evolving along one of these traits does not really need to be matched against categories *not* possessing these features. One could possibly try to evolve the objects in a pure bottom-up manner using an indexing approach.

The T-split sufficiency has benefits and drawbacks. Its benefits are that a principal rectangle is not split by its nested rectangles during feature evolvement. The drawback is that some pg3 features do not properly reflect their actual context and do not represent

useful features. These accidental detections do not affect recognition performance, but are one of the reasons why we have originally chosen the matching approach: it is difficult to develop an indexing approach, when there are accidental detections. The solution would be an improved context evaluation. Context should be evaluated at different radii or may be even dynamically, using for example a region analysis algorithm.

**Properties of category features.** As outlined in the introduction, the category-specific features derived in this work are to be understood as visual 2D pointers to category-specific knowledge. Some of the feature structure is loose in order to be able to cope with alignment variability: the bridge feature has a tolerant, angular measure for its vertical legs; it represents any two legs dangling in space; the cublk structure is not required to correspond to the silhouette of an exact cuboid. Folded surfaces are not required to touch each other exactly or represent any exact folding angles.

A few category features represent the typical sizes of the object’s proportions: the principal and nested rectangles formulated for the categories closet, drawers and desk, express the rough proportions by allowing rectangles of certain orientations only (type high/wide). These proportions are necessary in order to get the categories properly distinguished, because of their similarly nested rectangles.

Most of the finer structure, like arm rest in chairs or leg support in chairs and tables is not properly reconstructed using the simple context measure. Even if it had been reconstructed, it had to be analyzed more extensively because it can not be understood by mere surface and silhouette features. This is structure that is rather understood in a visual search using category-specific knowledge. There is other structure that is not understood by our program, for example the side wall of the desk’s chest forming the leg room.

**Category representations.** We do *not* propose that feature lists suffice for category representations. If one attempted to extract the meaning of a room scene (containing furniture objects), then one would possibly not be able to localize or categorize objects since the features might appear at random locations in the scene to the recognition system. The category features therefore have to be in some relation to each other. These relations do not have to be tight, rigid or fixed. They must be loose like some of the category features, in order to be able to match the structural variability. For example, the closet can be represented by a cublk silhouette containing the door surface fitting approximately between the silhouette’s vertical contours. Similar, the drawers are represented by their cublk silhouette with the drawer front fitting in between. The chair is represented by a double bridge with a backrest

or lean feature on top of it. Category representations would be a collage of features standing in approximate relations to each other, rather than a fixed set of features.

**The need for complex and individual features.** While presumably most vision researchers would agree with edge detection followed by integration to some global features, many may be sceptical about the idea of a larger set of complex or individual features as the ones we have evolved here. The problem with choosing only a small set of features (or parts or primitives) is, that it requires an enormously large number of rules to describe an object - whether these rules are envisioned as associations or as an emergence of synaptic weights. In contrast, with more complex features, representations are expressed conciser and recognition evolves faster. Because the human visual system recognizes swiftly and with ease, we believe that it must use complex features for representations. The ones we have derived here, can be understood as possible examples.

**Usefulness of over-determined representations.** Many contour pieces of an object are used for surface and silhouette features simultaneously. Consequently, an object is over-determined: only a fraction of features is necessary to categorize the object. This is not useless redundancy, but advantageous, if not even necessary whenever only a fraction of the set of features is available. This frequently occurs:

1) In real-world scenes in which part of the contours are often of low contrast or ambiguous due to noise and thus not easily detectable [Canny, 1986].

2) When we see a novel category instance that has partially new features. The remaining, familiar features would trigger the category process and the new features could be learned immediately.

3) When objects are in real-world backgrounds, the silhouette contours can be interrupted by contours from objects in the background, e.g. a floor contour. In such situations, surface features are more decisive in the categorization process.

4) Or when the object is in sharp contrast with its background (the 'entering a dark room' situation we have sketched to imagine the silhouette features). In this case, silhouette features are more stringent in determining category type.

This over-representation could also explain, why humans never make categorization mistakes. Humans may not uniquely categorize structurally ambiguous or distorted objects, but we never have to guess the category type for a regular object.

**Flexible Recognition Evolvment.** The list of points in the previous paragraph implied something about recognition evolvment: because there are so many different situations in which an (exact same) object displays a different subset of its features, the recognition process has to start with a given feature subset and still yield the same categorization result. Recognition evolvment can thus not progress along a fixed path. A prominent debate relating to this issue, is the discussion whether local or global structure is interpreted first [Navon, 1977, Palmer, 1999]. If local structure was interpreted first, then the furniture objects had to be interpreted by its local surfaces and gradually integrated to the global object structure. If global structure was to be interpreted first, then one had to start with silhouette features and work toward local surfaces. Given the previously sketched situations, neither recognition evolvment is preferred: its starting point depends on the situation and its resulting displayed features. Thus, recognition evolvment may be a highly flexible process.

**Alternative context evolvment.** The way we evolve surfaces and silhouettes is not so much different from a process called region analysis. Region analysis is a process that determines which parts of an image are separate regions [Palmer, 1999]. Applying such a process to our objects would firstly return all surfaces as separate regions as well as the regions outside the object, its silhouette. In a second step, the shape of the regions had to be determined, in order to interpret their possible orientation in space. This sequence is opposite to our feature evolvment: our system starts with a local region analysis - the context - for each L feature and then integrates these to pg3s, surfaces and silhouettes. Both directions seem valid.

**Extension to real-world objects.** What are the challenges that one faces if one extends this work to real-world objects depicted in grey-scale images? We discuss two main points. Firstly, there is the problem of contour extraction in gray-scale images [Palmer, 1999]: Most of the contours in a grey-scale image can be extracted using different scales, but are usually fragmented across these scales [Canny, 1986]. To obtain more continuous contours one had to either integrate across scales [Witkin et al., 1987] or to automatically select the appropriate scale [Lindeberg, 1998]. Alternatively, it may not be necessary to obtain a perfect contour picture if one uses a tolerant measure of connecting the line segments [Lee and Fu, 1983, Rasche, 2002a]. Secondly, the shape-variability is so vast, that it would be desirable to have a shape-encoding mechanism that bears some tolerance toward part-shape variability. One such mechanism has been created by Blum [Blum, 1973], which he called the symmetric-axis transform (also called medial-axis transform elsewhere): it finds the symmetry axis of

a shape irrespective of its exact contour geometry. This transform has also been applied to grey-level images [Wang et al., 1982].

## **Acknowledgments**

This work was supported by the Engineering Research Centers Program of the National Science Foundation under Award Number EEC-9402726. The author wishes to thank C. Koch for sustained support.

## References

- [Barrow and Tenenbaum, 1981] Barrow, H. and Tenenbaum, J. (1981). Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, 17:75–116.
- [Biederman, 1972] Biederman, I. (1972). Perceiving real-world scenes. *Science*, 177(43):77–80.
- [Biederman, 1987] Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychol Rev*, 94(2):115–47.
- [Binford, 1971] Binford, T. (1971). Visual perception by computer. In *Proceedings of the IEEE Conf. on Systems and Control*, Miami.
- [Blum, 1973] Blum, H. (1973). Biological shape and visual science. *J Theor Biol*, 38(2):205–87.
- [Brooks, 1981] Brooks, R. (1981). Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, 17:285–348.
- [Canny, 1986] Canny, J. (1986). A computational approach to edge-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.
- [Clowes, 1971] Clowes, M. B. (1971). On seeing things. *Artificial Intelligence*, 2(1):79–116.
- [Grimson, 1990] Grimson, W. E. L. (1990). *Object recognition by computer: the role of geometric constraints*. MIT Press, Cambridge, Mass.
- [Guzman, 1969] 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.
- [Guzman, 1971] Guzman, A. (1971). Analysis of curved line drawings using context and global information. In Meltzer, M. and Michie, D., editors, *Machine Intelligence 6*, chapter 20, pages 325–375. Edingburgh University Press, Edingburgh, Scotland.
- [Hubel and Wiesel, 1962] Hubel, D. and Wiesel, T. (1962). Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *J. Physiol.*, 160:106–154.
- [Huffman, 1971] 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 et al., 1992] 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.
- [Lee and Fu, 1983] Lee, H. and Fu, K. (1983). Generating object descriptions for model retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 5(RM118):462–471.
- [Lindeberg, 1998] Lindeberg, T. (1998). Feature detection with automatic scale selection. *Int. J. Comput. Vis.*, 30(2):79–116.
- [Lowe, 1987] Lowe, D. (1987). 3-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395.
- [Marr, 1982] Marr, D. (1982). *Vision*. W. H. Freeman, New York.
- [Marr and Nishihara, 1978] Marr, D. and Nishihara, H. (1978). Visual information-processing - artificial intelligence and sensorium of sight. *Technol. Rev.*, 1(FS127):28–49.
- [Navon, 1977] Navon, D. (1977). Forest before trees: the precedence of global features in visual perception. *Cognitive Psychology*, 9:353–383.
- [Palmer, 1999] Palmer, S. E. (1999). *Vision Science: Photons to Phenomenology*. MIT Press, Cambridge, Massachusetts.
- [Palmer et al., 1981] 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.
- [Pentland, 1986] Pentland, A. (1986). Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331.
- [Potter, 1975] Potter, M. C. (1975). Meaning in visual search. *Science*, 187(4180):965–6.
- [Rasche, 2002a] Rasche, C. (2002a). Recognizing the outline of room scenes. page <http://www.klab.caltech.edu/~rasche>.
- [Rasche, 2002b] Rasche, C. (2002b). Structural description with classified polygon and surface features and their groupings. <http://www.klab.caltech.edu/~rasche>.
- [Robert, 1965] Robert, L. G. (1965). Machine perception of three-dimensional solids. In Tippet, T. e. a., editor, *Optical and Electro optical Information Processing*. MIT Press.

- [Rosch et al., 1976] Rosch, E., Mervis, C., Gray, W., and Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology*, 8:382–439.
- [Schendan et al., 1998] Schendan, H. E., Ganis, G., and Kutas, M. (1998). Neurophysiological evidence for visual perceptual categorization of words and faces within 150 ms. *Psychophysiology*, 35(3):240–51.
- [SHAPIRO et al., 1984] SHAPIRO, L., MORIARTY, J., HARALICK, R., and MULLGAONKAR, P. (1984). Matching 3-dimensional objects using a relational paradigm. *PATTERN RECOGNITION*, 17(4):385–405.
- [Tarr and Bulthoff, 1998] Tarr, M. J. and Bulthoff, H. H. (1998). Image-based object recognition in man, monkey and machine. *Cognition*, 67(1-2):1–20. (eng).
- [Thorpe et al., 1996] Thorpe, S., Fize, D., and Marlot, C. (1996). Speed of processing in the human visual system. *Nature*, 381:520–522.
- [ULLMAN, 1990] ULLMAN, S. (1990). 3-dimensional object recognition. *COLD SPRING HARBOR SYMPOSIA ON QUANTITATIVE BIOLOGY*, 55:889–898.
- [Waltz, 1975] Waltz, D. (1975). Understanding line drawings of scenes with shadows. In Winston, P., editor, *The Psychology of Computer Vision*. McGraw-Hill, New York.
- [Wang et al., 1982] Wang, S., Rosenfeld, A., and Wu, A. (1982). A medial axis transformation for grayscale pictures. *IEEE Trans. Pattern Anal. Mach. Intell.*, 4(NT735):419–421.
- [Witkin et al., 1987] Witkin, A., Terzopoulos, D., and Kass, M. (1987). Signal matching through scale space. *Int. J. Comput. Vis.*, 2(M2049):133–144.



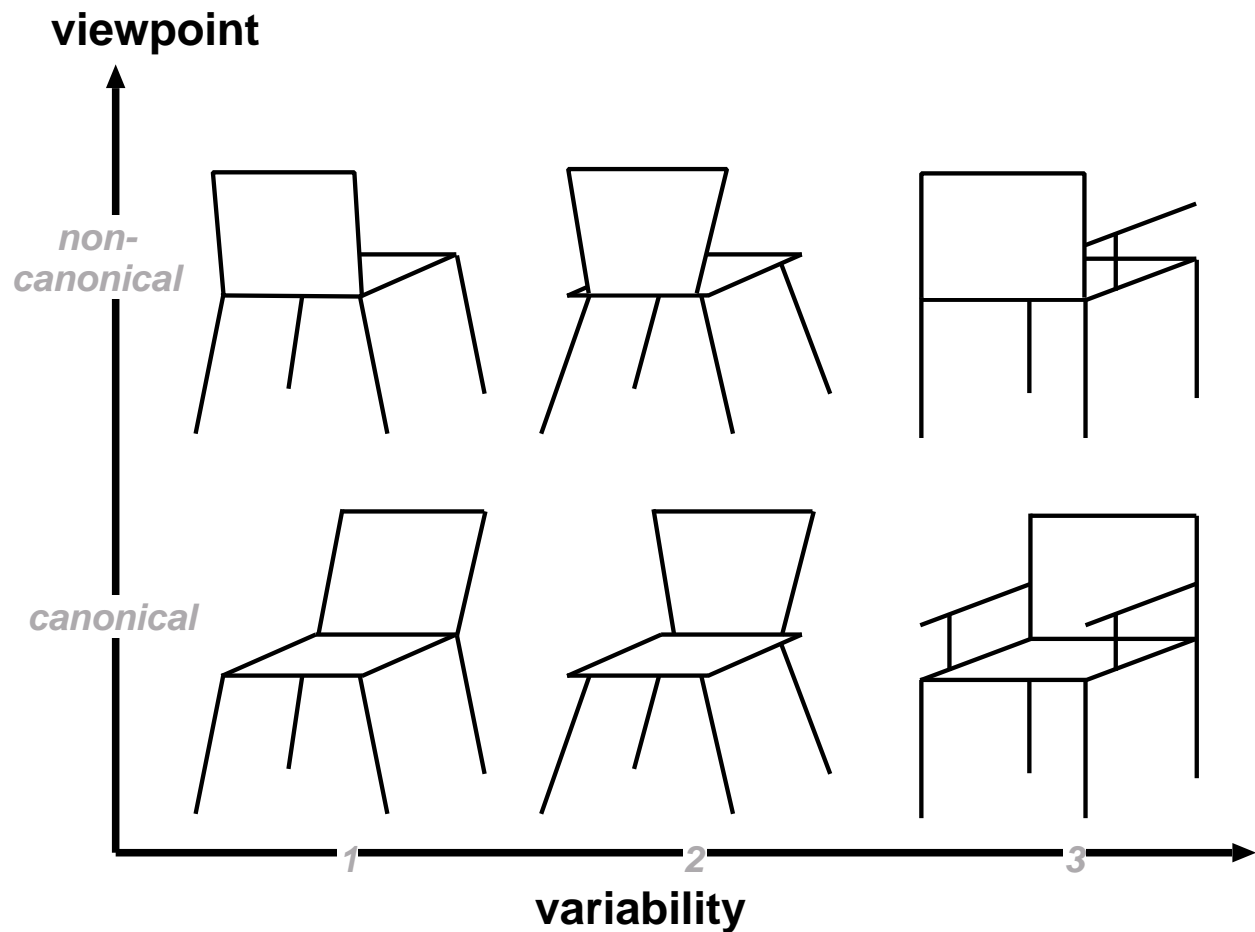


Figure 1: Two recognition aspects, viewpoint and variability independence, are illustrated as if they were two dimensions. The x-axis represents the variability aspect: 3 different chairs are shown reflecting only marginally the blatant variability of real-world objects. The y-axis represents the viewpoint aspect: each chair is shown from a canonical and a non-canonical viewpoint.

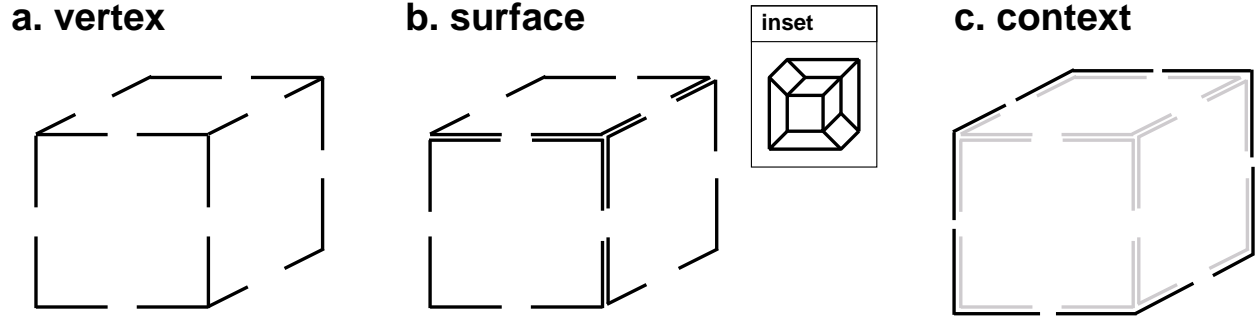


Figure 2: Different approaches to describe a cube. a. Vertex approach: the cube consists of 7 vertices, of which 3 are L features (two intersecting lines) and 4 are 3-line vertices (three intersecting lines). b. Surface approach: the cube is described by three surfaces, each made of 4 L features. c. Cube described by three surfaces and a silhouette: These structures are evolved by analyzing the context of L features (see next figure).

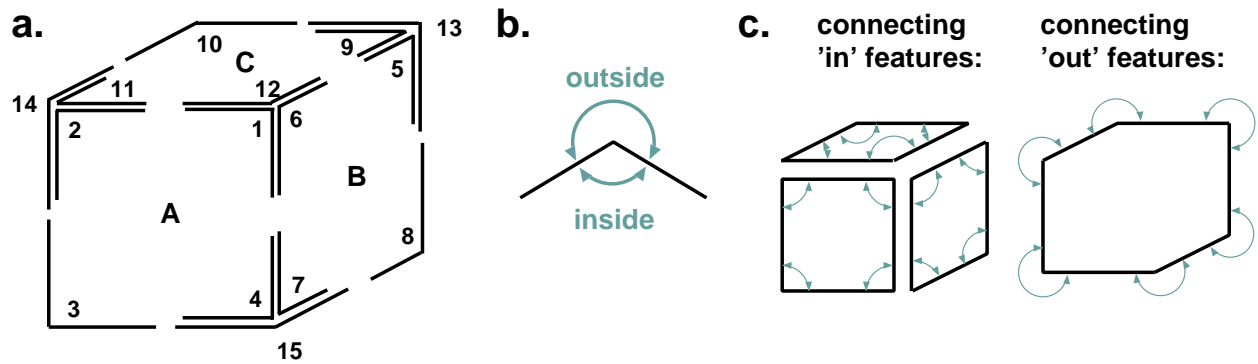


Figure 3: The idea of context illustrated on the cube. a. The 15 L features of a cube: 4 for each square and 3 as part of the silhouette. b. The inside and outside of a L feature. c. L features whose inside is empty (In features) are useful for extracting surfaces. L features whose outside is empty (Out features) are useful for extracting the silhouette.

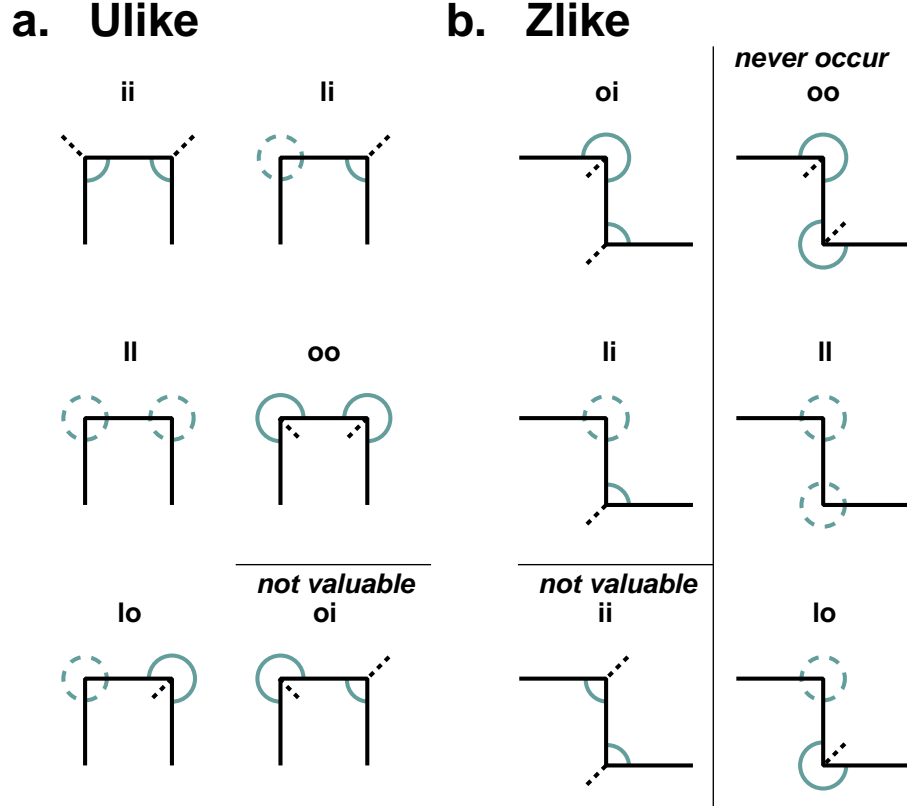


Figure 4: Context evaluation of (open) 3-line polygons (pg3s). Pg3s are shown only for one orientation and an angle of 90 degrees for L features. a. Ulike features (outer legs on same side of the middle line). b. Zlike features (legs on different sides). Annotation. i: In feature, o: Out feature, l: Lone feature (In & Out). The grey arcs at each corner, 1/4 and 3/4, indicate the structure-free side of In and Out feature, respectively. Grey stippled circles denote Lone features. The black stippled line indicates that there is some structure on that side of the corner.

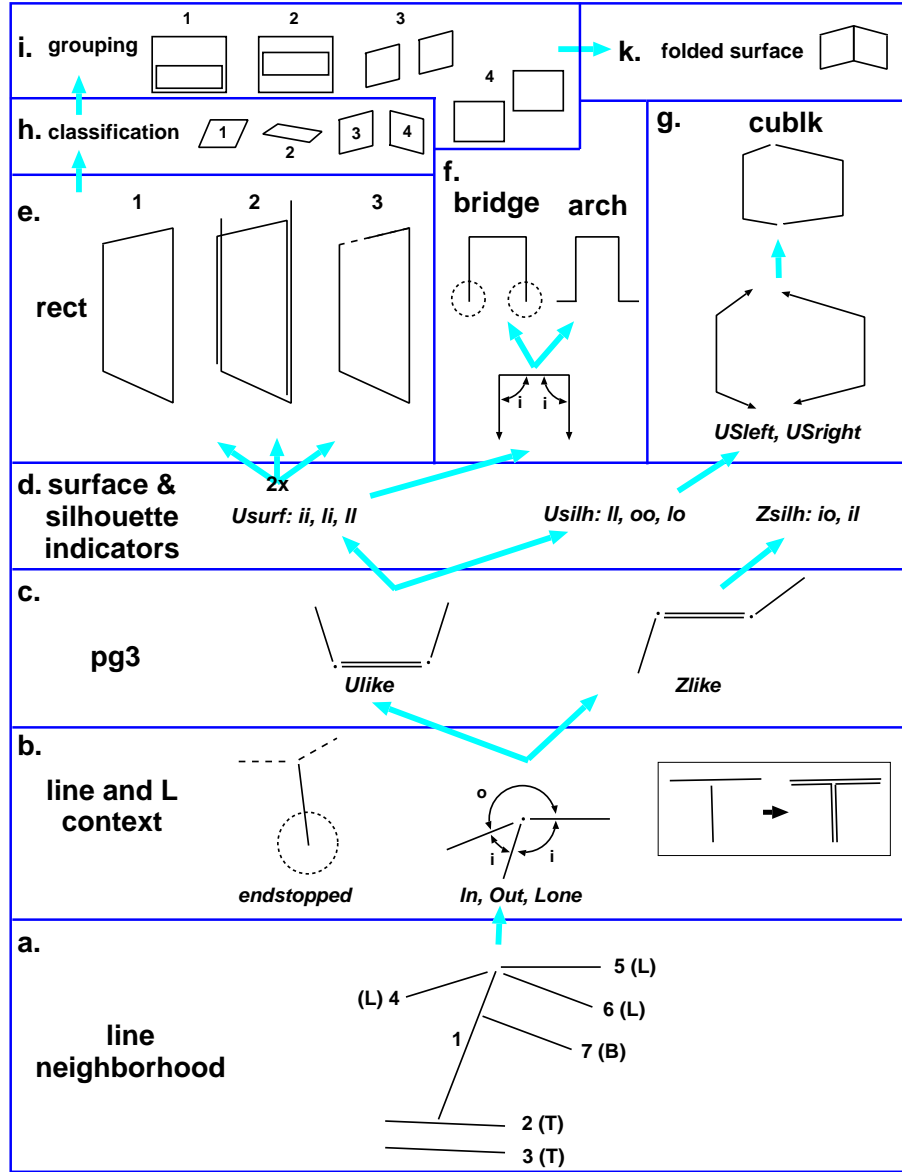


Figure 5: Feature extraction (bottom-up) process. Schematic illustration of the formation of features from lines (bottom) to complex features and structures (top). For details see text.

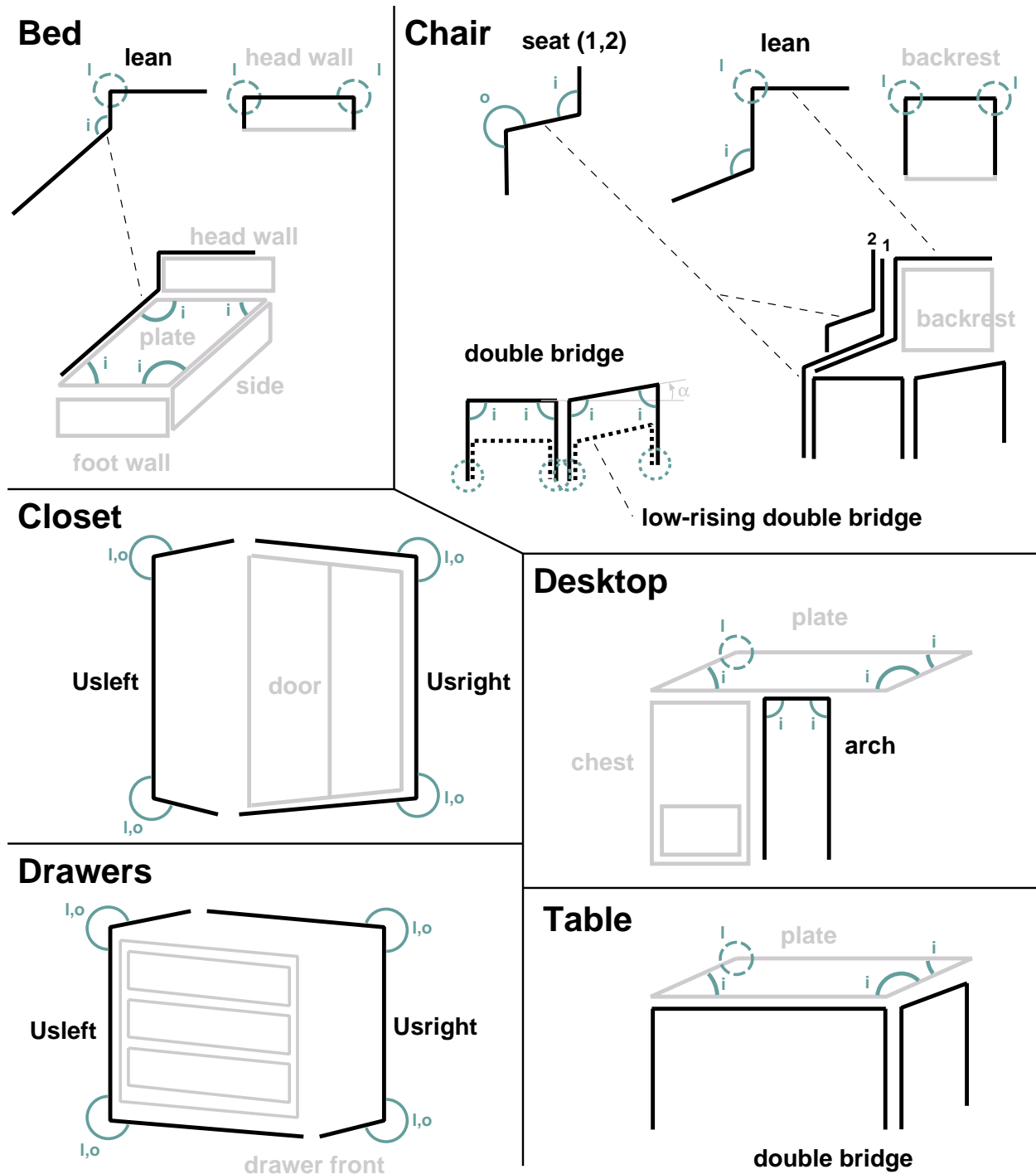


Figure 6: Category-specific features (pointers) used in the matching procedure (top-down process). Grey: surface features. Black: silhouette features. The features do not reflect all the geometrical conditions. See text for further comments.

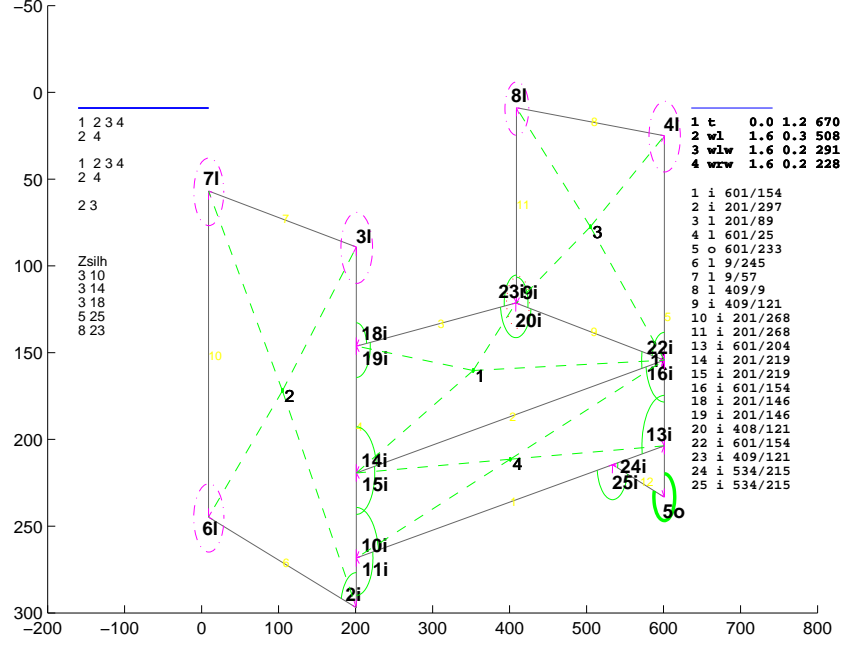


Figure 7: Object bed. **Surfaces:** 3D Rectangles are visualized by their stippled diagonals and are also listed by their size on the top right with corresponding classification: type (wall/tile/plate, left/right/, wide/high), tilt, slant and 3D area (area not used in this study). Angles are given in radians. 3D rectangle groupings are listed on the top left in three blocks: Adjacent groupings (1st block, 2 lines): Each line lists the 3D rectangles that are adjacent to the first one in its line. Folded groupings (2nd block, 2 lines): Each line lists the 3D rectangles that form a foldable surface with the first one in its line. In this example, the 1st and 2nd block happen to be identical. Parallel groupings (3rd block, 1 line): Lists the parallel and shifted groupings. (Caution: x and y axis do not have the same scale, which needs to be considered when interpreting angles). **Silhouette:** L features are denoted by their list number followed by their context classification (i, o or l). They are also listed on the right, below the list of 3D rectangles, with the coordinates of their corner points (x/y). (L features are not continuously numbered because of eliminated, useless L features in the list). Zsilh features are listed on the left, denoted by their two constituent L features. The context of L features is graphically illustrated as before: Lone features are denoted by stippled, gray circles; In features by gray arcs; Out features are marked by gray arcs with a thicker line width than others.

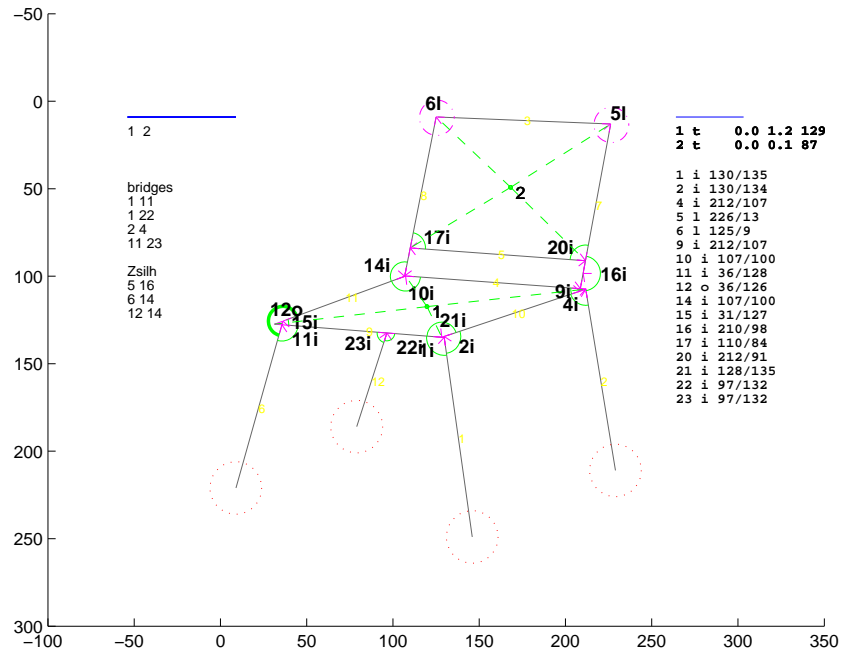
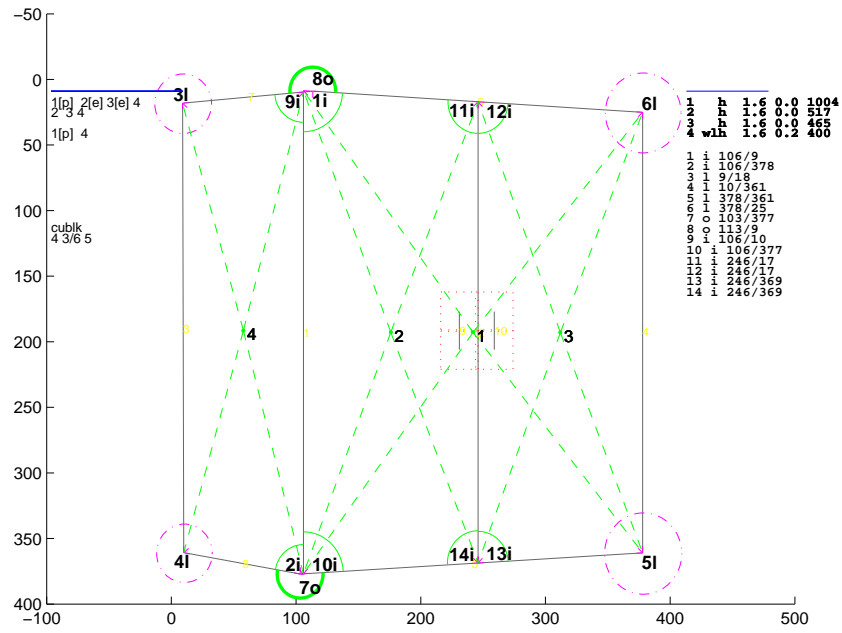


Figure 8: Object chair. The thin stippled circles mark end-stopped lines. Bridge and Zsilh features are listed on the left.





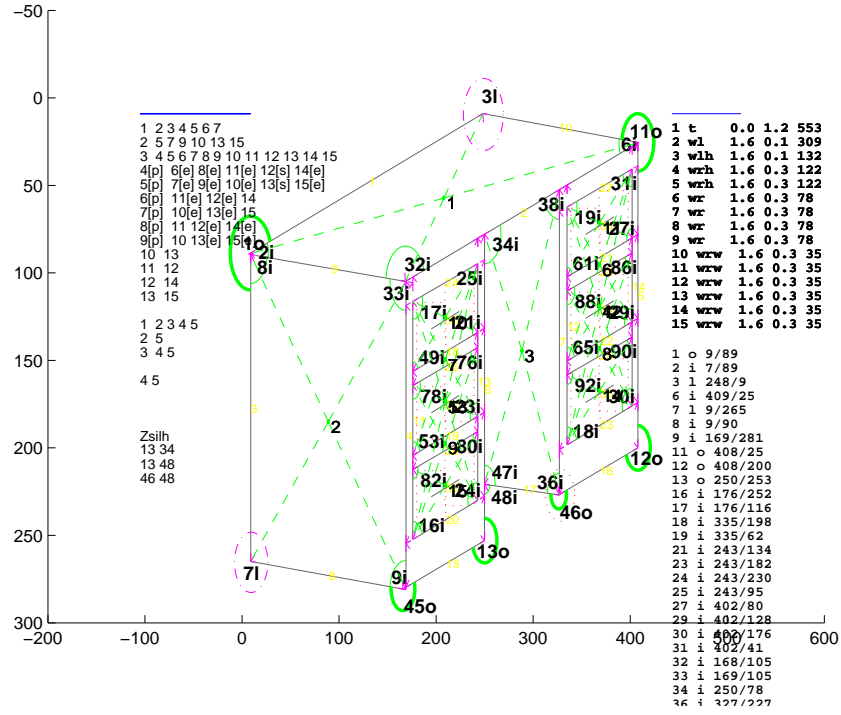


Figure 10: Object desk. 3D rectangle grouping notation: [s], sliced.

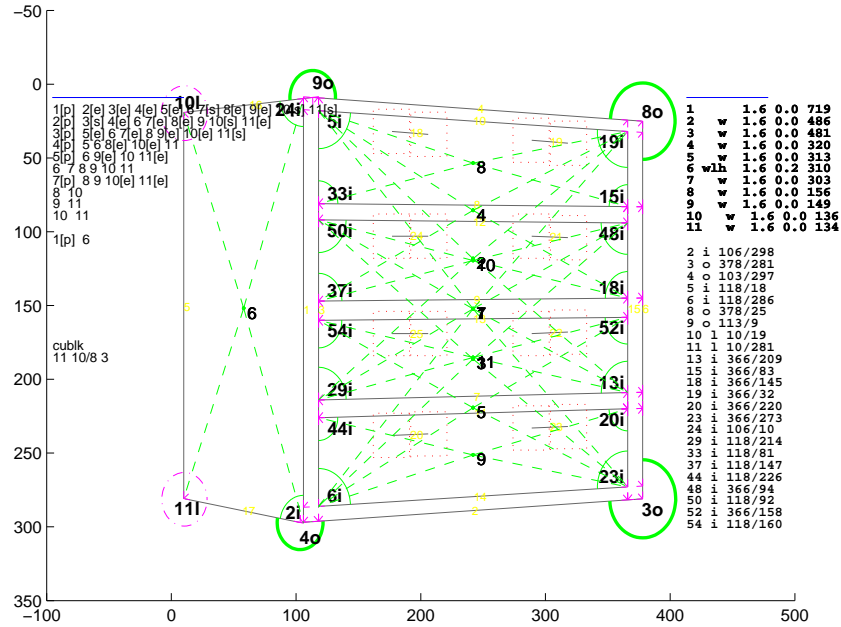


Figure 11: Object drawers.

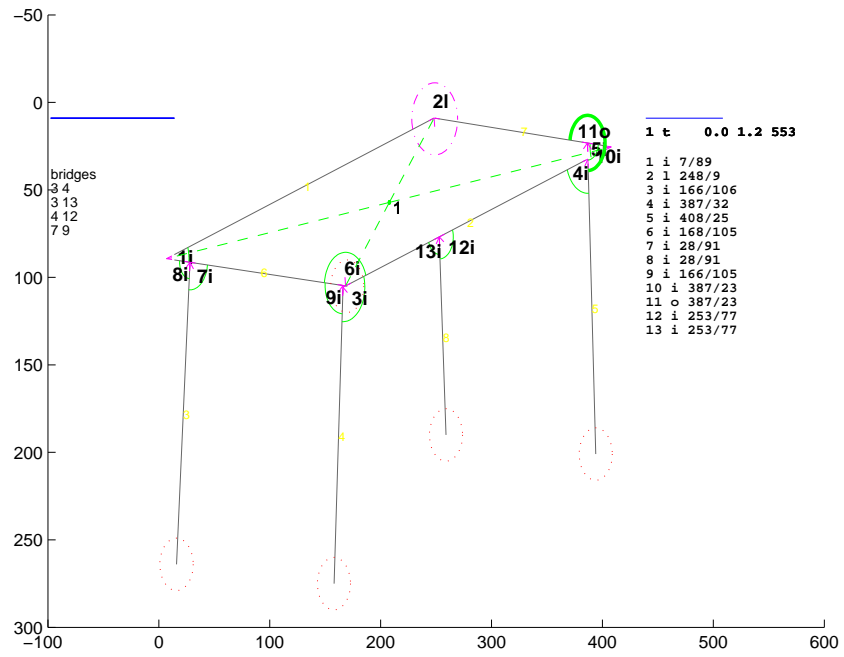


Figure 12: Object table.

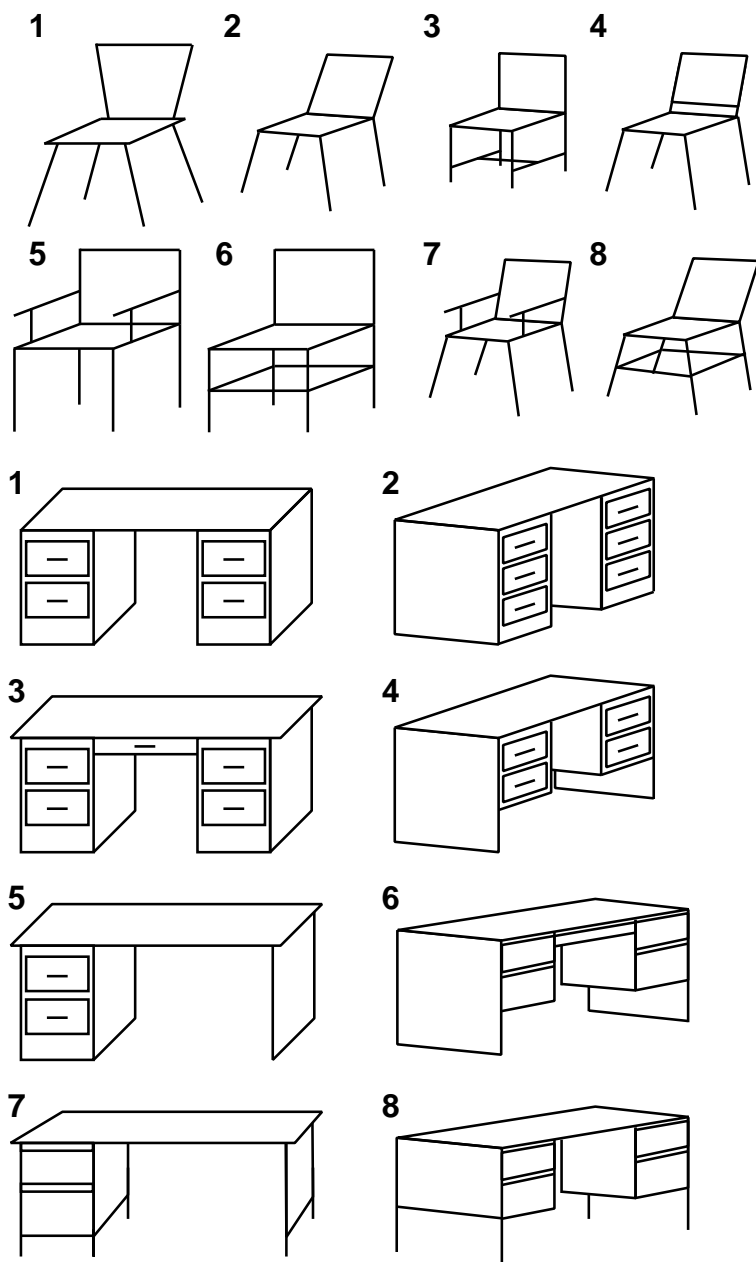


Figure 13: Objects of categories chair and desk.

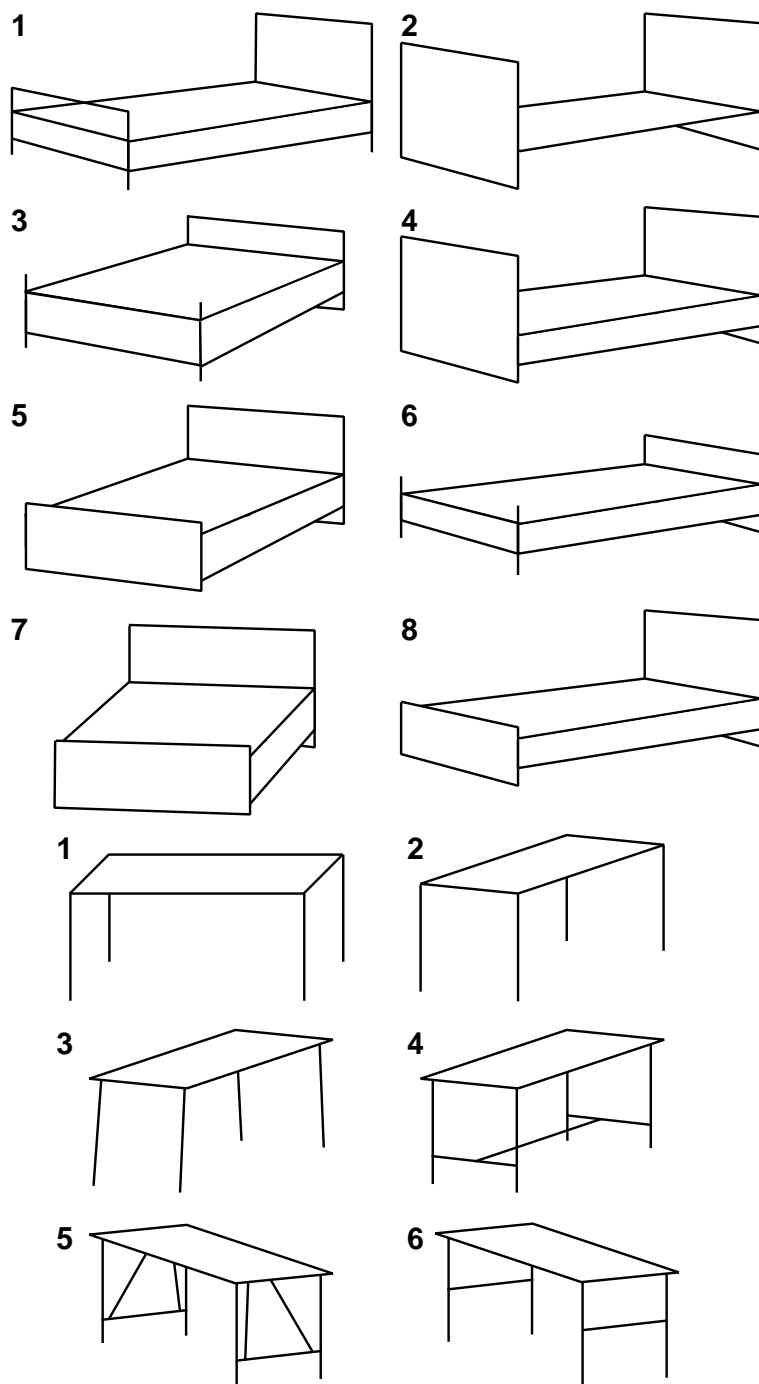


Figure 14: Objects of categories bed and table.

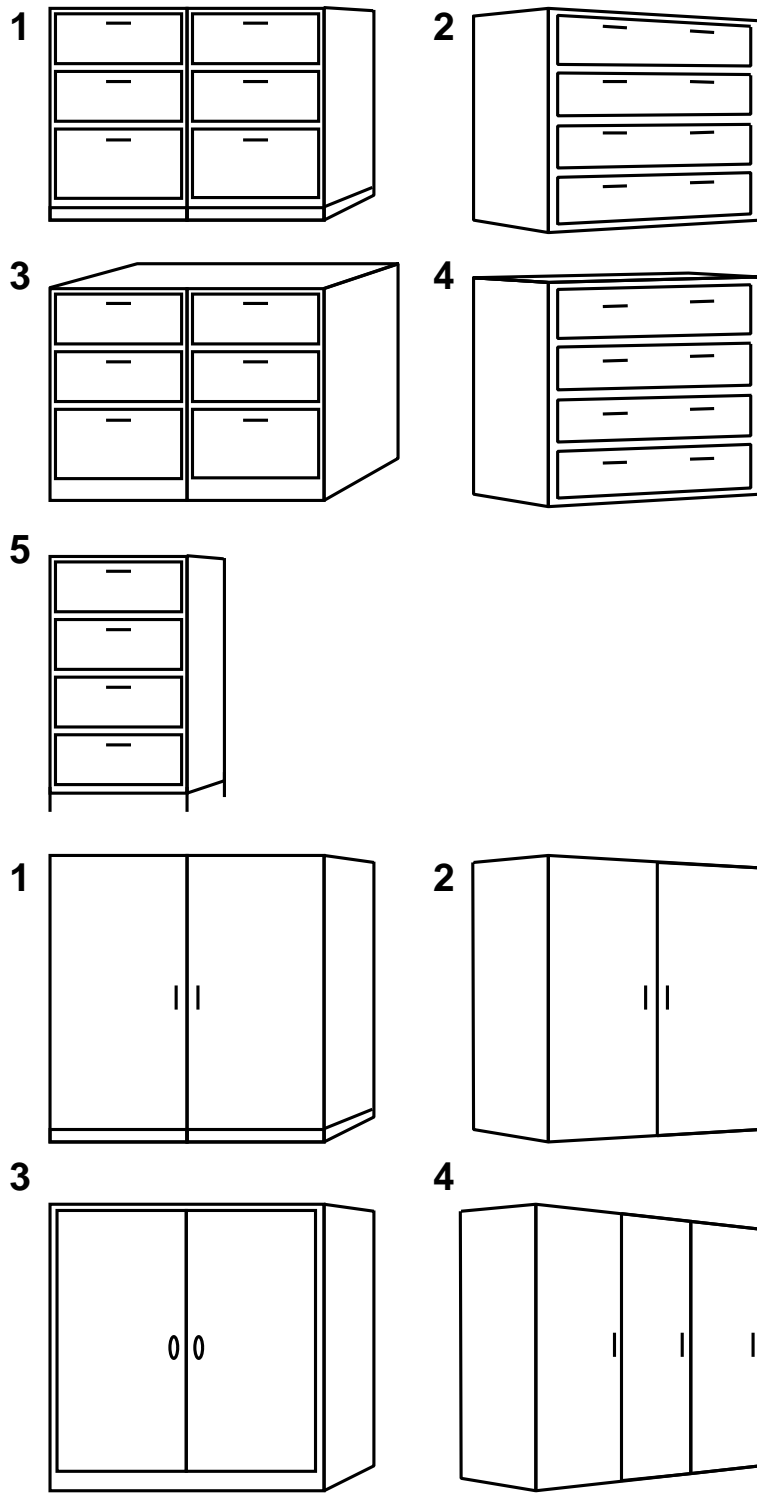


Figure 15: Objects of categories drawers and closet.



<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