10 A Method for Recognition and Localization of Generic Objects for Indoor Navigation

Dongsung Kim and Ramakant Nevatia

We introduce an efficient method for recognition and localization of generic objects for robot navigation, which works on real scenes. The generic objects used in our experiments are desks and doors as they are suitable landmarks for navigation. The recognition method uses significant surfaces and accompanying functional evidence for recognition of such objects. Currently, our system works with planar surfaces only and assumes that the objects are in a "standard" pose. The localization and orientation of an object are represented with the most significant surface in an "s-map." Some results for laboratory scenes are given.

10.1 Introduction

Our goal is to provide visual capabilities for a robot to navigate in indoor environments such as an office building. For this, not only must the robot be able to sense the objects in its environment for the purpose of obstacle detection but also recognize some of them to be used as landmarks for navigation. One approach to this task could be to provide a detailed map of the objects and structures in the environment to the robot. This allows conventional model-based object recognition techniques to be used for landmark detection and path planning. This strategy, however, has several limitations. First, the objects and their arrangement in an indoor setting are constantly changing. Even normally stationary objects, such as furniture, may be moved occasionally. Also, providing detailed geometric models for all objects even in a single room can be a very difficult and tedious task. When a common object, such as a desk, is replaced by another one, completely new models may have to be provided even if the two objects serve similar functions.

To overcome these difficulties, we propose to represent the objects and structures by some *generic* models. This enables the objects to be recognized as belonging to a certain class without having to also determine which specific one. Such generic modeling allows the robot to navigate in environments without knowledge of the specific instances or their locations.

An example of a scene that the robot must handle is shown in Figure 10.1. The robot may be asked to use a desk as a landmark and to pass through a door after the desk. We only wish to provide generic descriptions of the room and the objects to the robot. A room can be thus characterized by horizontal doors and vertical walls that may have doors. The room may consist of objects, such as desk, whose generic properties are known to the robot, but also other unknown objects which can not be used as landmarks but nonetheless must be avoided during navigation. In this paper, we will focus only on the recognition of the generic classes.

(a) Image 1 (b) Image 2 (c) Image 3

Figure 10.1 Door 1

Little research has been done on recognition of generic classes of objects in computer vision. It is somewhat difficult to precisely define the notion of a generic class, but it surely excludes precise geometric models. The most generic representation of an object is probably in terms of its functionality. Thus, concept of a door is that of an opening allowing passage of objects and a means of closing this opening. The concept of a desk is that of an object that allows a human to place objects on it and work on them in a comfortable posture. The inference of such high level functionality from real images is, however, quite difficult. Some attempts have been made towards this [Star-Bowy91], but these systems do not take images as inputs. In early work, Tenenbaum and Garvey [Garv-Tene74,Tene73] attempted to recognize objects in an office scene by using point properties; however, such properties are not sufficient to distinguish among complex objects in a complex scene, in general. Another approach is to view various instances of a class as generated by varying parameters of a parametric representation. Parametric representations have been studied by several researchers [Broo83,Grim89,Lowe87,Marr-Nish77,Neva-Binf77], however, such approaches do not naturally capture the common variations found in everyday objects such as desks.

In this paper, we propose an approach of representing objects by their significant surfaces and by relations among them, which we believe is sufficiently general for recognizing common objects in an indoor environment and can work with real scenes. Significant surfaces are chosen based on their functional role. A desk is thus characterized by a working surface and some surfaces that correspond to the support structures, that is “legs.” The working surface is characterized by some properties, such as a range of sizes and heights. Besides the observation of surface properties, the surfaces can also be observed by their function, for example, a working surface of a desk could be inferred by the objects placed upon it; we call such evidence as the *functional* *evidence*. Such a representation can allow us to recognize several instances of desks, not all similar in shape and construction, and not necessarily seen previously, in a robust and efficient manner.

The method is robust because significant surfaces aid in detection of the other significant surfaces. A significant surface can confine a domain of another significant surface. The confined domain allows the system to recover a missed significant surface.

Our current system makes two major assumptions. First, it deals with only planar surfaces. Objects with non-planar surfaces must have sufficient planar surfaces to be recognized. Second, it assumes that an object is in a "standard" pose. A standard pose of an object is the one in which the object is usually found in its natural environment. This standard pose allows the systems to recognize objects efficiently. For example, a desk is expected to be placed with its working surface horizontal.

Currently, our system has models for only two classes of objects: doors and desks. We believe that these are sufficient as landmarks for navigation in many office and laboratory environments. Moreover, we believe that our approach can accommodate a wider range of objects easily.

A block diagram of our system is given in Figure10.2. Note that we do not automatically infer significant surfaces from functional descriptions; this translation is currently done by the programmer.

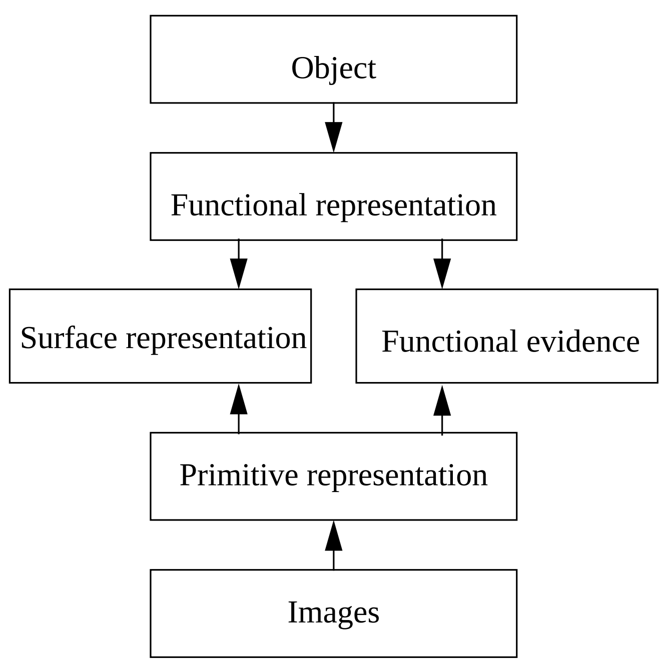


Figure 10.2 Overview of the recognition system

The input to our system is from three cameras arranged to perform trinocular stereo and mounted on a Denning mobile robot. We perform stereo matching on linear line segments detected from the three images. Our matching method is described in detail in [Kim-Neva93] and is similar to one developed at INRIA [Ayac91] but incorporates important junction information. Note that this essentially provides a sparse set of 3-D segments and a somewhat larger set of 2-D segments for the robot to attempt its recognition from.

Detection of the significant surfaces from 2-D and 3-D segments is basically done using perceptual grouping. The detection utilizes the four primitives of the surfaces: orientation, height, shape, and size. The orientation and height are used to reduce search space while the shape and size are utilized for perceptual grouping. Shape of a significant surface may vary for a generic object. Such varying shapes can be detected with help of other significant surfaces. For example, a round desk top surface can be detected with help of legs. The details are described in Section 10.4.

Localization of an object in our system is represented in an *s-map*. The s-map is defined as a map that represents the locations of the visible 3-D surfaces of obstacles in a 2-D space, where 2-D consists of width and length coordinates but does not include a height coordinate. Obstacles are defined as objects that can block the movement of the robot. The s-map is made efficiently from 3-D segments. Further details can be found in [Kim-Neva94].

Section 10.2 explains the significant surface of an object and its primitives. Perceptual grouping for the significant surface is also investigated. Section 10.3 describes recognizing and localizing doors. Section 10.4 resents recognition and localization of desks. Section 10.5 analyzes the recognition system. Finally, Section 10.6 concludes this paper.

10.2 Significant Surface Representation

Significant surfaces follow from the functions that an object performs. Note that one surface may serve several functions whereas a single function may require presence of several surfaces. For example, for a desk, the function of being able to work at a comfortable heights requires a table top within a certain height range as well as some legs for support.

We order the significant surfaces by how essential they are to the functions that they enable. For a desk, we consider the top surface to be more significant than the legs. For an object to be recognized, its most significant surface must be detected.

10.2.1 Primitives of a Significant Surface

A significant surface in a standard pose can be characterized by the four primitives: orientation, range of heights, shape, and size. The orientation and height are decided by the standard pose while the shape and size are determined by the significant surface itself. The orientation and height can reduce the number of candidate segments for perceptual grouping to find the surface having the shape and size.

- Orientation: a significant surface of an object has a fixed orientation relative to a horizontal plane in standard pose. For example, a desk top is horizontal.
- Range of height: all the points in a significant surface are within a certain range in terms of their heights above the floor. For example, desk legs have a height range of between zero and 1 meter above the floor.
- Shape: shape of a significant surface may be given in general form. For instance, desk tops generally have a rectangular shape. However, we may be able to detect desk with non-rectangular shape also, based on evidence provided by legs and objects supported on it.
- Size: size of a significant surface of an object is within a certain range. For example, a desk top should have an appropriate size for a human to work on.

10.2.2 Perceptual Grouping to Detect a Significant Surface

Perceptual grouping is used to find a significant surface from 2-D and 3-D segments. The candidate segments can be limited using orientation and height primitives of the significant surface. Perceptual grouping for the significant surface varies depending on its shape. Thus the details of perceptual grouping for each significant surface are explained in the related sections.

In perceptual grouping, 2-D information as well as 3-D information is utilized. This can reduce grouping errors caused by matching errors. The loss of information caused by missing matches and/or partial matches may be recovered using 2-D segment information in two ways. First, missing matches hinder two less meaningful features from becoming a more meaningful one because of lost information. Such lost information can be recovered if there are 2-D segments that can support the meaningful feature. For example, a rectangle with a 3-D U-shape can be classified as a rectangle with higher confidence if it has 2-D segments that can make the U-shape a rectangular shape. Second, partial matches may prevent two less meaningful features from being grouped into a more meaningful feature when proximity of two features is used as one criterion. The partial matches may lose adjacent portion of two less meaningful features, and prevent them from being a more meaningful feature. This error can be overcome if 2-D segments in addition to 3-D segments are used in checking the proximity of two features.

10.3 Recognition of Doors

A door has some functions. The most significant function is for a human and other objects such as furniture to pass through. This decides the most significant surface, a door frame. Another significant function can be separation of space when the door is not used as a passage. This determines another significant surface, a door panel. In addition to these significant surfaces, a door may be supported by functional evidence.

The functional evidence consists of objects seen through a door when it is open. This information helps to decide that the detected door is not a simple drawing but a real door.

First, detecting a door frame as the most significant surface is described. Then detecting a door panel as another significant surface is explained. Next, detecting functional evidence is explored. Finally, the results of detecting doors are illustrated.

10.3.1 Detecting a Door Frame

Adoor frame can be characterized by the following four primitives: a vertical orientation, a height range between a floor and 2.5 meters above the floor, a rectangular shape, and passable size. The orientation and height reduce search space for candidate segments. The shape and size decide a perceptual grouping method to detect the door frame from the candidate segments.

A door frame has three components: top bar, left pole, and right pole. A door frame can be detected by finding a U-shape consisting of the top bar, the left pole, and the right pole.

Candidate segments for top bars, left poles, and right poles are searched in a limited space as described below. The candidate segments are then grouped into door frames. The details of collecting candidate segments and perceptual grouping are described below. In addition, the algorithm to detect door is summarized in Figure 10.3.

1. Collect possible top bars
2. Collect left and right poles
3. Hyphothesize doors
   If there is a top bar
      then hypothesize a door with a top bar, left pole, and right pole
      else hypothesize a door with a left pole and right pole
            if there is a 2-D top bar between them
4. Verify a door
   3-D validation with distance and alignment
   2-D validation with distance

Figure 10.3 Algorithm for detecting a door frame

Candidate segments are efficiently collected using orientation and size primitives for the three components described above. The orientation and size confine the search space of the candidate segments to vertical surfaces whose height is between a floor and 2.5 meters above the floor. A top bar should be a horizontal line of sufficient length and height, so that a human can pass under the top bar. Poles should be vertical lines that are high enough for a human to pass, and the distance between the two poles should be wide enough so that a human can pass. First, collecting possible top bars are explained. Collecting possible poles are then explored.

Possible top bars are collected from matched segments. A segment can be a possible top bar if it has sufficient height and width. While checking the length of a matched segment, shrunk 3-D segments due to partial matching can be recovered by considering what portion of 2-D segments are used in reconstructing the 3-D segment.

Left and right poles have the same characteristics in terms of primitives. The poles are also collected among matched segments. A vertical segment can be a possible pole if it is high enough.

Now top bars and poles are perceptual grouped into U-shapes. If top bars of all doors are assumed to be detected, then hypothesizing doors is relatively simple. However, the assumption is not always true. Thus missing a top bar should be considered when hypothesizing doors.

When a top bar is available, a door is hypothesized with a left pole, a top bar, and a right pole. The right pole is in the scope of the top bar. The right pole in the scope of a top bar is a pole below the top bar in an image.

When a top bar is unavailable, a door is hypothesized only with a left pole and a right pole. The right pole is next pole to the left pole. In addition, the distance between them is sufficient for a door. The poles should have a 2-D top bar bridging them. If so, a door is hypothesized with the top bar and two poles.

After hypothesizing a door frame, the door frame is verified with its 2-D and 3-D information. As a 3-D validation, distance and alignment are verified. The distance gap between an end point of a top bar and each pole should be within a threshold value. Moreover, Three components of a door frame should be aligned to a single line in an s-map because they are aligned to a single line at the top view. As a 2-D validation, the distance gap between each end point of a top bar and an upper end point of each pole should be within a threshold value. These verification criteria are also used in selecting a door frame among those hypothesized door frames sharing the same top bar, which can be generated when a door frame has more than two distinct poles. For example, a door frame having two door panels can have more than two distinct poles if a center pole is detected.

10.3.2 Detecting a Door Panel

A door can have several panels. When a door is closed, it is difficult to distinguish panels from a door frame. However, detecting a door panel is easier when a door is open. The panel is attached to a door frame and has a rectangular shape. In the current implementation, our system tries to detect a door panel only when a door is open. The opening of a door is decided using functional evidence described in Section 10.3.3. Detecting a door panel helps to detect a door.

Detecting a door panel is similar to detecting a door frame except that the door panel should be attached to the door frame. Searching for a door panel is done near two poles of the door. After finding an horizontal segment reaching a corner of the door frame, a vertical line reaching the horizontal segment is found. With this horizontal line and the vertical line, a door panel is hypothesized. The same verification used in door frame verification is applied to verify a door panel.

10.3.3 Finding Functional Evidence

When a door is open to pass, its opening gives functional evidence that constitutes objects seen through the opening. Thus detecting objects inside a door helps to detect the door. To acquire this functional evidence, we collect segments inside a door frame. Then the segments are checked if they are behind the door by a minimum distance set by expected accuracy of depth determination from the viewpoint of the robot. If so, functional evidence for a door is claimed to be achieved. Moreover, these segments behind the door mean that the door is open.

When a door is closed, the functional evidence is not available.

10.3.4 Localization of a Door

After a door is found, the location of the door is represented in an s-map for navigation. Representing the door in an s-map is very simple. A vertical line becomes a point in an s-map because dropping height information of the vertical line renders a point. Thus the door in the s-map is a line linking two points generated from two vertical poles.

In addition to the location of a door, the facing direction of a door should be known to the robot so that the robot is able to reach in front of the door. The facing direction is decided considering locations of both the door and the robot. At first, two locations, which are perpendicular and a predefined distance away from a door, are computed. Then one between the two possible locations is selected based on the distance between the robot and one position. The nearer location is selected because seeing an object means that the front part is always nearer than the back part.

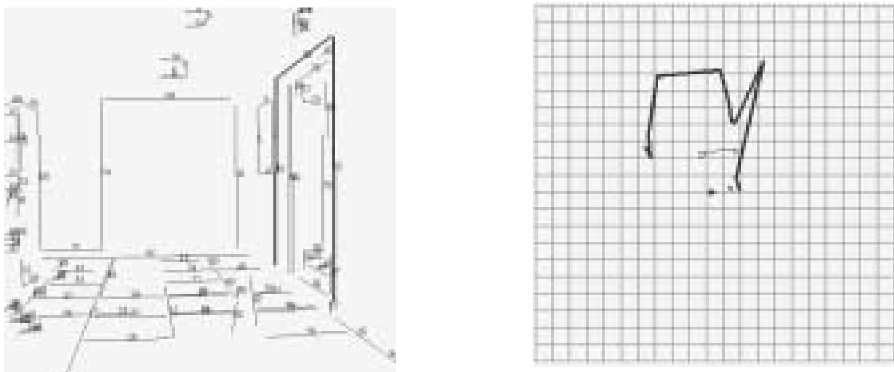
10.3.5 Results for recognition of Doors

Figure 10.4-a and 10.4-b illustrate recognition of an open door for an image scene shown Figure 10.1. In this scene, there are two doors. The right door is open because it has functional evidence of an open door as described in Section 10.3.3. The thicker lines in Figure 10.4-a represents an open door. Moreover, the center position of a door and the front part of a door are represented with small circles bridging the thin line in Figure 10.4-b

The closed door in Figure 10.1 is also detected, but is not shown here separately.

10.4 Recognition of Desks

A desk has the following significant surfaces: the desk top and the legs. The most significant surface of a desk is the desk top. Less significant surfaces are the legs. In addition, evidence of a desk may also be found by detecting the function it performs, namely of supporting objects on the desk top.



(a) Matches (b) S-map

Figure 10.4 Open door

10.4.1 Detecting a Desk Top Surface

The most significant surface for a desk is the top surface. The primitives for a desk top surface are as follows: horizontal orientation, height range between 60 cm and 90 cm, rectangular shape, and workable size. The size is assumed to be from 40 cm to 2 meters in each side. The shape primitive is loosely preserved. Although the system tries to detect the rectangular shape, it allows the desk top to be an arbitrary shape. The desk top is initially detected using perceptual grouping. Then other significant surfaces are used to detect the desk top correctly.

The detection of a desk top using perceptual grouping can be done in two stages: collecting candidate segments for a desk top and perceptual grouping for a desk top. In collecting candidate segments, orientation and height primitives are used to reduce the search space. Segments that are both horizontal and 60-90 cm high above the floor, are collected.

In perceptual grouping, collected candidate segments are grouped into a rectangular shape having workable size. Perceptual grouping is done in four steps: collinearization, L-shapes, U-shape, and rectangular shape. In the first step, possible desk top segments are collinearized based on the angle difference and the gap between two segments. The gap can be as large as the size of a desk because a large part of a desk may be occluded by materials on the desk. In the second step, collinearized lines form L-shapes based on angle and gap between two lines. The angle between two lines should be perpendicular in 3-D. The gap between two lines should be within a threshold value. In checking for a gap, a 2-D gap as well as a 3-D gap is also used to recover errors caused by partial or wrong matches. In the third step, L-shapes form U-shapes. Two L-shape can make a U-shape if they have a common line and the other lines have the same direction. Moreover, the size of a U-shape should be large enough to be a desk. In the final step, U-shapes make rectangular shapes. Two U-shapes can make a rectangular shape if they have two common lines. The rectangular shape should be large enough for a human to work on.

The detection of a desk top with help of other significant surfaces is done while detecting the other significant surfaces. The other significant surfaces can confine the domain of the desk top. The details are described in Section 10.4.2.

10.4.2 Detecting Legs

Detecting legs of a desk either adds confidence to the desk recognized with a desk top surface, or can help to detect a desk if a desk top surface is not detected. Missing a desk top surface occurs when there are no rectangular shapes, U-shapes, and L-shapes that are large enough because of occlusion by material on the top surface, or by material in front of the desk. This missed top surface may be recovered by detecting legs. While detecting legs, a domain of a desk top is acquired. The details are described below.

Legs have four primitives: vertical orientation, height range between a floor to the desk top surface, no common shape, and size that does not exceeding the 2-D desk domain with respect to an s-map. Orientation and height reduce possible candidates for legs. Then only size primitive is applied to group the candidates. The details of detecting legs are described below. In addition, the algorithm of detecting legs are summarized in Figure 10.5.

1. Collect vertical segments
2. Filter those reaching a desk top surface from the collected segments
3. Select those inside the 2-D domain of a desk from the filtered segments.
   (a) Find the 2-D domain of a desk with respect to an s-map
      i. Find the 1-D domain for a column coordinate
         A. Acquire an initial column domain
         B. Collect possible legs inside the initial column domain
         C. Confine a more accurate column domain from the collected possible legs.
         D. Collect more possible legs inside the more accurate column domain
      ii. Find the 2-D domain in an s-map
         A. Collect the segments inside the column domain
         B. Grow a rectangle containing the collected segments if they are within a reasonable range
   (b) Select those inside the 2-D domain

Figure 10.5 Algorithm for detecting legs

Detection of the legs of a desk is done in three modules: collecting vertical segments, filtering the vertical segments that can reach desk top height, and selecting the vertical segments inside a 2-D domain of a desk top. Collecting vertical segments is done by checking verticality of a segment. The second module is easily implemented by checking 3-D information of vertical segments. The final module needs to find the 2-D domain of a desk with respect to an s-map. After finding the 2-D domain of a desk, more possible legs are selected as legs if they are inside the 2-D domain of a desk. We describe a method of finding the 2-D domain of a desk.

Finding the 2-D domain of a desk is done in two steps: finding a column coordinate domain of a desk in an image, and finding the 2-D desk domain in an s-map.

In the first step, a column coordinate domain becomes more accurate when it interacts with legs. The method attempts to acquire an initial column domain. Then the acquired column domain is used to collect possible legs. Next the possible legs confine a more accurate column domain. Finally, the more accurate column domain is used to collect more possible legs. Among these procedures, we describe acquiring an initial column domain and confining a more accurate column domain because collecting legs in a column domain can be done simply by checking column coordinates of a segment.

Initial column domain is acquired either from segments at a desk height or from a detected desk top surface. The initial column domain should contain all the segments at desk height or segments of the desk top surface.

Confining to a more accurate column domain is done by using a presence row. The presence row is a single row indicating if a column coordinate of the row is occupied by a desk. This presence row is constructed by dropping the row coordinate of possible segments of a desk top surface and marking its coordinate as a filled cell. Thus the region occupied by a desk is marked with filled cells. After making the presence row, the extra possible legs are selected among all the possible legs using the presence row. In the presence row, a band of continuously filled cells is considered as a desk if the band is sufficiently wide. Thus possible legs under such a band become extra possible legs. Conversely, the band having extra possible legs can be considered as a desk. Therefore, such band becomes a column domain of a desk.

In the second step, a 2-D desk domain is computed in an s-map represented in terms of width and depth. A segment with desk top height is collected as a possible desk top segment if it is inside a column domain computed at the first step. After collecting possible desk top segments, a 2-D desk domain is grown by attempting to contain the desk top segments if they are within a certain range. The growing of a 2-D desk domain is described below. In constructing a rectangle containing the segments of a desk, computation time is reduced by transforming segments into another coordinate system. The coordinate system allows checking if a segment is contained to be easily performed by checking its row and column coordinates. Among all the segments of a desk top surface, the longest segment is selected as a reference segment. Then row and column coordinates are rotated so that the reference segment is parallel to row coordinate. This reference segment generates a desk rectangle of which one side is made with the reference segment and the other parallel side is made with a small perturbation of the reference segment. Now other segments are also transformed and their coordinates are compared to see if they are inside the desk rectangle. If a segment is not inside the desk rectangle, the desk rectangle is updated so that it can contain the segment unless the segment is too far. Finally, the desk rectangle is inversely transformed to a world coordinate system.

10.4.3 Finding Functional Evidence

The function of a desk, to work on, can generate functional evidence. When some objects are on the desk, these provide functional evidence. Thus objects on the desk can help recognize the desk. The objects on the desk should be inside the 2-D domain of the desk and reach the desk top.

Detecting materials on a desk top is accomplished in two steps. In the first step, segments inside a 2-D desk domain are collected. These segments can be efficiently collected using 1-D and 2-D filtering. For a 1-D filtering, segments inside the column domain of the desk are collected. Then the collected segments are further checked if they are inside the 2-D desk domain in an s-map. In the second step, segments reaching the desk top are selected among segments inside. In the current system, segments one or both of whose ends reach the desk top are considered as segments reaching the desk top.

10.4.4 Localization of a Desk

After a desk is found, localization is done in an s-map. Location of the desk is simply represented in the s-map by dropping height information of the four sides of the top surface.

In addition to location of a desk, the front direction of the desk should be known to a robot so that the robot can reach the desk and do some other work, such as getting a pencil in a drawer. To find the front part of the desk, common posture is utilized. A desk top has a rectangular shape. Moreover, either of longer sides of the desk top is a front part of a desk. In common posture, the front part of a desk faces a direction that is easily accessible. This implies that the front part of a desk is nearer than its rear part. Now detecting a front part becomes detecting the longer side facing a robot. The detecting of the front part is accomplished in two steps. At first, longer sides of a desk are selected. Then the nearer side between the longer sides is selected as the front part. This selection can be further verified when a robot approaches the desk and acquires more details of the desk front.

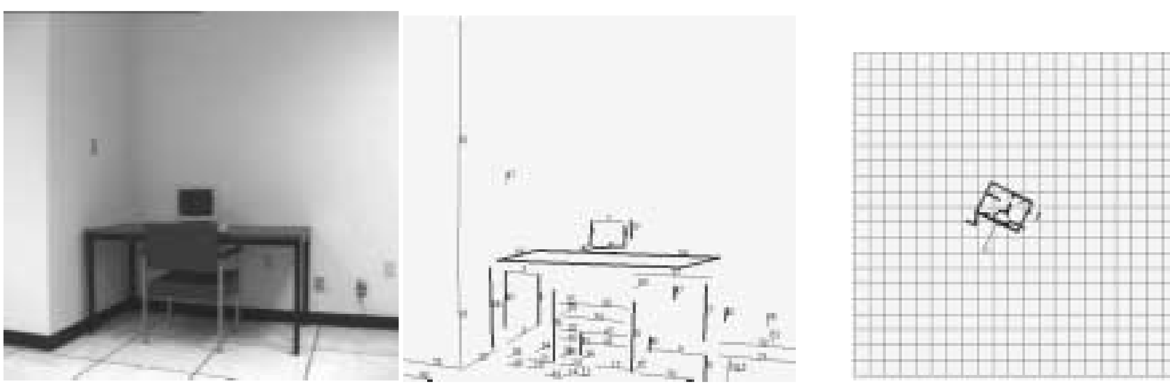
10.4.5 Results for Recognition of Desks

Several tens of experiments have been conducted to recognize four different kinds of desks in many different viewpoints, distances, and settings. Three of the four desks have a rectangular desk top, but have different shaped legs. One of them has drawers in both sides. Another has drawers in one side. The other has no drawers. The desk having a round desk top has only a single leg. The recognition system has successfully recognized the four desks in the experiments with settings of a monitor and a chair. The errors in orientation and size were within 20 percent in experiments done. Results of recognizing desks are given below.

Figure10.6-a shows an image taken by our robot, Antigone. Detecting significant surfaces and functional evidence is given in Figure 10.6-b, -c, and -d. Figure 10.6-b

represents a detected desk top surface on top of matched segments. The thin lines are matched segments. The thicker lines are detected desk top boundaries. The boundaries are generated from a U-shape candidate of a desk top surface because the other side is occluded by a monitor. Figure 10.6-c displays legs under a 2-D desk domain. The current algorithm to detect legs has loose criteria for deciding whether legs are reaching to a desk top because self occlusion and partial matches may prevent legs from reaching to a desk top. Figure 10.6-d illustrates the functional evidence of a desk. In collecting functional evidence, the current algorithm collects only segments reaching a desk top. These significant surfaces and functional evidence allow our recognition system to recognize and localize a desk. The recognized desk is localized in an s-map in Figure 10.6-e. The front part of the desk is represented with the thin line and the small circles.

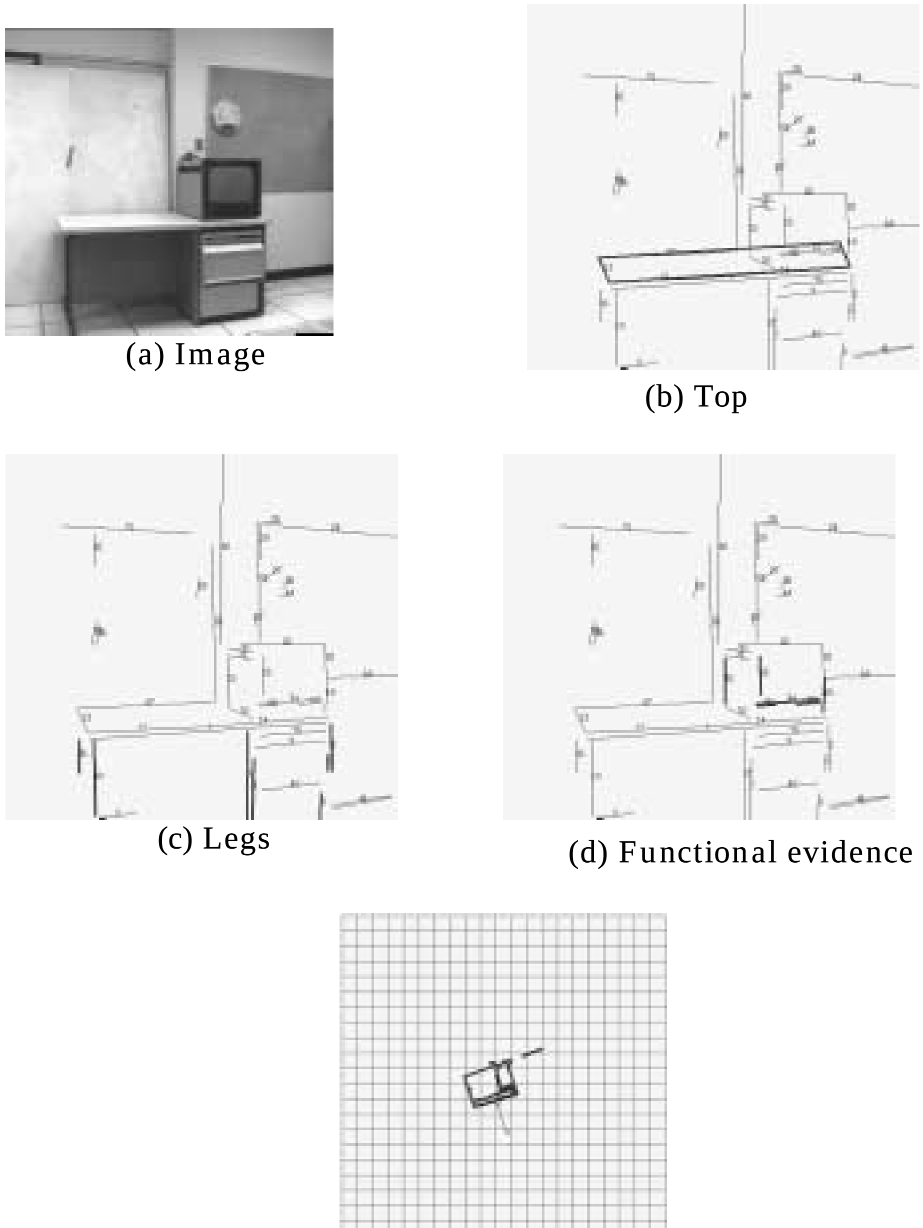
Figure 10.7 displays yet another desk scene. The desk has a monitor on it, and is occluded by a chair. Figure 10.7-b illustrates detected significant surfaces and functional evidence that are represented as thicker lines. All four boundaries of the desk top are successfully recovered although front and rear boundaries of the desk are occluded by a chair and monitor respectively in Figure 10.7-b. Moreover, legs and functional evidence are successfully detected. Figure 10.7-c represents the recognized desk in an s-map. The front part of a desk is rendered with the thin line and the small circles.



(a) Image (b) matches (c) S-map

Figure 10.7 Desk B

Figure 10.8 displays another desk scene. The desk has a monitor on it. Figure 10.8-b illustrates detected significant surfaces and functional evidence that are represented as thicker lines. All four boundaries of the desk top are successfully recovered although rear boundaries of the desk are occluded by a monitor in Figure 10.8-b. Moreover, legs and functional evidence are successfully detected.



(a) Image

(b) Top

(c) Legs

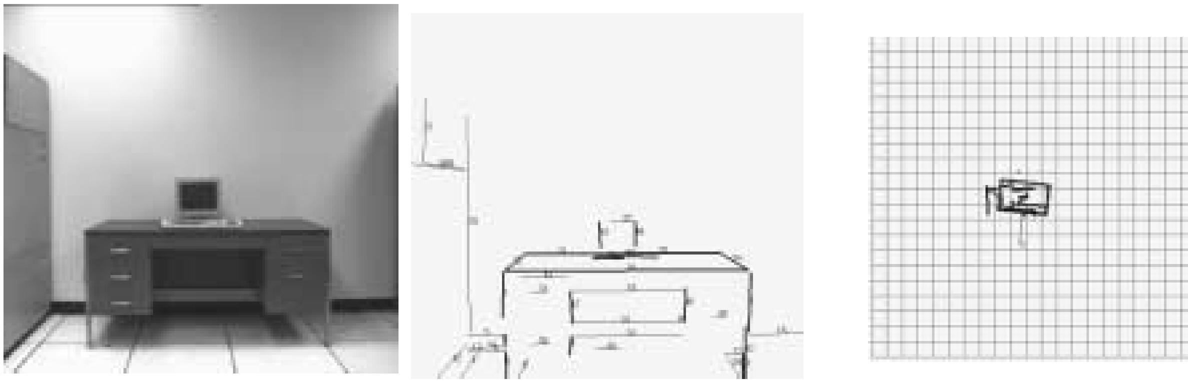
(d) Functional evidence

(e) s-map

Figure 10.6 Desk A

Figure 10.8-c represents the recognized desk in an s-map. The front part of a desk is rendered with the thin line and the small circles.

Figure 10.9 displays a round desk scene. Figure 10.9-b illustrates detected significant surfaces that are represented as thicker lines. As seen in Figure 10.9-b, a part of the round desk top is detected by perceptual grouping of a rectangular shape. Then,



(a) Image (b) matches (c) S-map

Figure 10.8 Desk C

the missed part of the round desk top is recovered with help of legs. Figure 10.9-c represents the recognized desk in an s-map. The missed part as well as the detected part is contained in the 2-D desk domain.

(a) Image (b) matches (c) S-map

Figure 10.9 Desk D

10.5 Complexity of the Recognition System

The complexity of the recognition system depends on a target object for recognition. In most cases, the complexity of the recognition system is decided by that of detecting the most significant surface. Because the most significant surface is a key for recognition and helps to confine domains of other significant surfaces, it should be detected correctly or at least have its domain selected roughly.

In the case of recognizing a door, worst case happens when half the segments are vertical lines and half the segments are top bars whose scopes are whole images. Then

$O(n^2)$ hypothesized door frames are generated. However, the complexity of detecting a door frame can be $O(n)$ in the average case because the scope of a top bar reaches two vertical lines in most cases. The complexity of other significant surfaces and functional evidence is $O(n)$. The total complexity for recognizing a door is $O(n)$ in the average case or $O(n^2)$ in the worst case.

In the case of recognizing a desk, worst case happens when all segments are desk top segments. Then generating L-shapes has complexity of $O(n^2)$. Moreover, generating rectangles from L-shapes also has complexity of $O(n^2)$ in the worst case. Therefore, the complexity of detecting a desk top is $O(n^2)$ in the worst case. However, we can reduce candidate segments for a desk top by utilizing natural posture and primitives of the desk top surface. In most cases, the number of the segments of a desk top is less than some constant number. These constant number of the desk top segments allows the system to have the complexity of $O(n)$. The complexities of other significant surface and functional evidence are also $O(n)$. The total complexity for recognizing a door is $O(n)$ in the average case or $O(n^2)$ in the worst case.

We have analyzed the real computing time for recognition from 3-D segments. The computing time was measured in tens of laboratory scenes using Sun Sparc station 10. Although the current system has been written in Lisp without optimization, it showed promising results in terms of computing time. For the case of doors in laboratory scenes, the computing time for recognition was less than one hundredth of a second. For the case of desks in laboratory scenes, the computing time was less than one tenth of a second. In addition to this computation, the edge detection takes about 40 seconds per image. The matching also needs about 10 seconds. From matched segments, an s-map is constructed in less than one tenth of a second. Among these, we estimate that edge detection and matching may be done within a second with parallel processing at a reasonable cost. Thus we believe that total computing time from images to object recognition can be less than a second with low level parallel processing.

10.6 Conclusion

We have shown some experiments on generic object recognition of desks and doors by using representations inspired by their functionality. Some of the evidence we use is rather weak by itself, however, it suffices in the context in which such objects are found. We believe that our methodology can also be applied to other large objects commonly found in offices and laboratories.

10.7 References

[Ayac91]N. Ayache. *Artificial Vision for Mobile Robots*. The MIT Press, 1991.

[Broo83]R. A. Brooks. Model-based three-dimensional interpretations of two-dimensional images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):140–150, 1983.

[Garv-Tene74]T. D. Garvey and J. M. Tenenbaum. On the automatic generation of programs for locating objects in the office scenes. *Second International Joint Conference of Pattern Recognition*, pages 162–168, 1974.

[Grim89]W. E. L. Grimson. On the recognition of parameterized 2d objects. *International Journal of Computer Vision*, 3:353–372, 1989.

[Kim-Neva93]D. Kim and R. Nevatia. Indoor navigation without a specific map. *Intelligent Autonomous Systems*, 1993.

[Lowe87]D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[Marr-Nish77]D. Marr and K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London*, B(200):269–294, 1977.

[Neva-Binf77]R. Nevatia and T. O. Binford. Description and recognition of complex-curved objects. *Artificial Intelligence*, 8:77–98, 1977.

[Star-Bowy91]L. Stark and K. Bowyer. Achieving generalized object recognition through reasoning about association of function to structure. *IEEE Trans. on the Pattern Analysis and Machine Intelligence*, 1991.

[Tene73]J. M. Tenenbaum. On locating objects by their distinguishing features in multisensory images. *Computer Graphics and Image Processing*, 2:308–320, 1973.