ATION PAGE

Form Approved
OMB No. 0704-0188

average 1 hour per response, including the time for reviewing instructions, searching existing data sources,
ing the collection of information. Send comments regarding this burden estimate or any other aspect of this
to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson
of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)

DATE
May 1993

3. REPORT TYPE AND DATES COVERED
memorandum

4. TITLE AND SUBTITLE
Data and Model-driven Selection Using Parallel-Line Groups

5. FUNDING NUMBERS
DACA76-85-C-0010
IRI-8900267
N00014-91-J-4038

6. AUTHOR(S)
S. Tanveer F. Mahmood

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
545 Technology Square
Cambridge, Massachusetts 02139

8. PERFORMING ORGANIZATION REPORT NUMBER
AIM 1399

9. SPONSORING/MONITORING AGENCY NAME(S)
Office of Naval Research
Information Systems
Arlington, Virginia 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
None

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution of this document is unlimited

13. ABSTRACT (Maximum 200 words)

A key problem in model-based object recognition is selection, namely, the problem of isolating regions in an image that are likely to come from a single object. This isolation can be either based solely on image data (data-driven) or can incorporate the knowledge of the model object (model-driven). In this paper we present an approach that exploits the property of closely-spaced parallelism between lines on objects to achieve data and model-driven selection. Specifically, we present a method of identifying groups of closely-spaced parallel lines in images that generates a linear number of small sized and reliable groups thus meeting several of the desirable requirements of a grouping scheme for recognition. The line groups generated form the basis for performing data and model-driven selection. Data-driven selection is achieved by selecting salient line groups as judged by a saliency measure that emphasizes the likelihood of the groups coming from single objects. The approach to model-driven selection, on the other hand, uses the description of closely-spaced parallel line groups on the model object to selectively generate line groups in the image that are likely to be the projections of the model groups under a set of allowable transformations and taking into account the effect of occlusions, illumination changes, and imaging errors. We then discuss the utility of line groups-based selection in the context of reducing the search involved in recognition, both as an independent selection mechanism, and when used in combination with other cues such as color. Finally, we present results that indicate a vast improvement in the performance of a recognition system that is integrated with parallel line groups-based selection

14. SUBJECT TERMS
grouping selection
attention

15. NUMBER OF PAGES
[illegible]

16. PRICE CODE

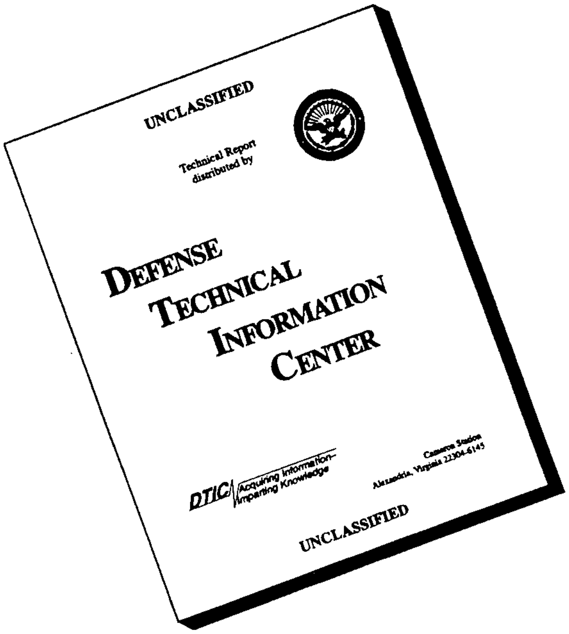
17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT

DISCLAIMER NOTICE



UNCLASSIFIED

Technical Report
distributed by

DEFENSE
TECHNICAL
INFORMATION
CENTER

DTIC Acquiring Information—
Imparting Knowledge

Cameron Station
Alexandria, Virginia 22304-6145

UNCLASSIFIED

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1399 May, 1993

Data and Model-driven Selection using Parallel-Line Groups

S. Tanveer F. Mahmood

Abstract

A key problem in model-based object recognition is selection, namely, the problem of isolating regions in an image that are likely to come from a single object. This isolation can be either based solely on image data (data-driven) or can incorporate the knowledge of the model object (model-driven). In this paper we present an approach that exploits the property of closely-spaced parallelism between lines on objects to achieve data and model-driven selection. Specifically, we present a method of identifying groups of closely-spaced parallel lines in images that generates a linear number of small-sized and reliable groups thus meeting several of the desirable requirements of a grouping scheme for recognition. The line groups generated form the basis for performing data and model-driven selection. Data-driven selection is achieved by selecting salient line groups as judged by a saliency measure that emphasizes the likelihood of the groups coming from single objects. The approach to model-driven selection, on the other hand, uses the description of closely-spaced parallel line groups on the model object to selectively generate line groups in the image that are likely to be the projections of the model groups under a set of allowable transformations and taking into account the effect of occlusions, illumination changes, and imaging errors. We then discuss the utility of line groups-based selection in the context of reducing the search involved in recognition, both as an independent selection mechanism, and when used in combination with other cues such as color. Finally, we present results that indicate a vast improvement in the performance of a recognition system that is integrated with parallel line groups-based selection.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Army contract number DACA76-85-C-0010, under Office of Naval Research contract N00014-91-J-4038 and under NSF Grant IR1-8900267. The author was supported by an IBM Graduate Fellowship during part of the work.

1 Introduction

A key problem in object recognition is selection, namely, the problem of identifying regions in an image within which to start the recognition process, ideally by isolating regions in an image that are likely to come from a single object. Model-based object recognition methods that try to recognize which members of their library of models are present in the scene, usually use geometric features such as points or edges and try to identify pairings between data and model features that are consistent with a rigid transformation of the object model into image coordinates. The large number of such pairings that need to be examined in cluttered scenes leads to a combinatorially explosive search problem. It has been shown that this search can be considerably reduced if recognition systems are equipped with a selection stage where subsets of data features can be isolated that are likely to come from a single object, thus allowing the search to be focused on those matches that are more likely to lead to a correct solution [9]. This isolation can be either based solely on image data (data-driven) or can incorporate the knowledge of the model object (model-driven). Even though selection can be of help in recognition, it has been found difficult to achieve in practice. What makes selection so difficult? In the ideal case, if the appearance of the desired object in the scene were known, and objects in the scene were nicely separated and distinguishable from the background, and the illumination conditions were known, then even simple methods that rely on intensity measurements would work well to extract groups of features. But in reality, the appearance of the object is not known. In addition, illumination conditions and surface geometries of objects present in a scene can cause problems of occlusion, shadowing, specularities, and inter-reflections in the image and make it difficult to interpret groups of data features such as edges and lines as belonging to a single object. Previous approaches have mainly considered data-driven selection by treating as a problem of grouping data features based on some constraint such as parallelism, or collinearity, [17], distance and orientation [13], and regions enclosed by a group of edges [3] to capture some meaningful structure in a scene. Grouping was introduced as a technique of reducing the search involved in recognition by dividing the search for matching features into a search for a matching pair of groups followed by a search for corresponding features within the matching groups. To effectively reduce the search in recognition, however, a grouping scheme must produce a small number of small-sized groups that are reliable (i.e. span a single object)[5]. Existing grouping schemes usually meet some but not all of these requirements. For example, schemes that capture only meaningful structure without reasoning about scene geometry can produce groups that are unreliable such as the grouping of salient image contours described in [25]. In other grouping schemes, the weakness of the constraints used can lead to a large (potentially exponential) number of groups [13]. Other schemes that restrict the number of groups can either cause some relevant groups to be missed leading to false negatives during recognition [11] or cause the groups to become unreliable [3].

So the general problem of selection remains largely unsolved as it is still not obvious how to reliably characterize subsets of data features that will give clues that point to a single object. We have been involved in developing a computational model of selection that proposes that selection can be achieved via an attention mechanism. Specifically, it is an attempt to build a computational model of visual attentional selection in humans, and to propose it as a selection mechanism for recognition. Towards this end, two modes of human attentional behavior, namely attracted-attention and pay-attention modes, have been isolated to serve as paradigms for data-driven and model-driven selection respectively. The *attracted-attention* mode of behavior is spontaneous and is commonly exhibited by an unbiased observer (i.e., with no *a priori* intentions) when some objects or some aspects of the scene attract his/her attention. The *pay-attention* mode is a more deliberate behavior exhibited by an observer looking at a scene with *a priori* goals (such as the task of recognizing an object, say) and hence paying attention to only those objects/aspects of a scene that are relevant to the goal. According to this model, therefore, data-driven selection can be achieved by identifying regions in an image that attract attention (i.e., that are distinctive) with respect to some feature such as color or texture, while model-driven selection can be achieved by paying attention to the model features (i.e., using the model features to decide saliency of features in the image). While it is understandable that paying attention to model features can help isolate areas in the image that could contain subsets of data features that are likely to contain a single object (or the specific model object in this case), it is not immediately apparent how locating salient regions is an appropriate way of performing data-driven selection. Such a choice is, however, motivated by the observation that an object often stands out in a scene because of some distinctive features that are usually localized to some portion of the object. Therefore isolating distinctive regions is more likely to point to a single object, making such regions an appropriate choice in the absence of any specific information about the model object. A number of other approaches have also suggested that selection, at least data-driven, can be performed based on some measure of saliency [24].

The above discussion indicates a framework for achieving data and model-driven selection. But how can salient regions be found for data-driven selection, and how can the object model affect the choice of salient regions for model-driven selection? In earlier work we had presented methods of selection based on color [27, 28] and texture [29]. There it was shown that the number of data features can be greatly reduced by such a selection. But since the regions isolated were rather large, the groups contained a large number of features causing the number of matches between model and image features to be still considerably large. If features within such regions could be grouped further such that only a small number of features fall into a group, then by finding a correspondence between such small region groups on the model and image, the total number of combinations of model and image features can be greatly reduced. In this paper, we explore the use of a property called closely-spaced parallelism that is often exhibited by lines on objects, to perform data and model-driven selection. Specifically, we show how small-sized and reliable groups can be generated using this constraint of closely-spaced parallelism lines and how such groups can be used to perform data and model-driven selection. Even though grouping is still at the heart of a such a selection mechanism, we show that the use of closely-spaced parallelism as a constraint causes it to meet several of the desirable requirements for recognition, albeit at the expense of an increase in the worst-case search complexity over conventional grouping.

The rest of the paper is organized as follows. We first discuss the need for grouping data features into small-sized groups for the purposes of recognition. This gives us a set of requirements that must be met by any scheme for grouping data features. We then briefly review the existing grouping methods in the light of these requirements. Next, we present a method for grouping line features that exploits the property of closely-spaced parallelism among lines. We then explore the use of line groups as a feature by itself for performing data and model-driven selection. In keeping with the general paradigm of attentional selection presented above, data-driven selection is achieved by selecting some salient line groups, while model-driven selection is achieved by utilizing the description of line groups on the model object. Next, we show how such line groups-based selection can be combined with other methods of selection based on cues such as color, to further reduce the search in recognition. Lastly, we present results that indicate the actual improvement in performance of a recognition system that uses line groups-based selection.

2 Role of Grouping in Model-based Recognition

Region selection using color and texture as described in earlier work [27, 28, 29] reduced the search involved in recognition by removing a large number of data features from consideration. Even so, once a set of regions is selected, a large number of matches between features in corresponding model and image regions may have to be tried. Using the alignment method for recognition (in particular, the linear combination of views version of this method [31]) we know that at least four matching features must be found for alignment (and hence recognition). If there are M features (say, points) in a model region and N features in the corresponding image region, then $O(M^4N^4)$ matches per pair of corresponding regions may have to be tried, in the worst case, with such region selection. For the typical number of features ($M \approx 100, N \approx 300$) found in color or texture regions, this is still a very large number of matches to be tried ($\approx 10^{18}$). If the data features within such regions can be further grouped into some meaningful structures or groups consisting of a small number of features each, then the search can be reduced by pairing such groups, and trying combinations of features within matching groups, as before. Previous research has explored the role of grouping in recognition for reducing the search in precisely this fashion [13, 3, 17, 9]. To see how grouping of features can reduce the combinatorics of search drastically, we reproduce here the analysis of grouping given in earlier work [13, 3].

Let us consider the case of grouping being performed both on the model and image features. Let M_g and N_g be the number of model and image groups respectively, and let m_i and n_j be the number of features in the model group i and image group j. If the size of the model and image groups are identical, and each group contains features coming from a single object, then the number of matches that need to be tried are $O(\sum_{i=1}^{M_g} \sum_{j=1}^{N_g} m_i!)$ since all pairs of model and image groups may have to be tried and $m_i!$ accounts for all permutations of feature matches within a pair of matching groups. Further, if the features in a group can be linearly ordered, the number of matches reduces to $O(\sum_{i=1}^{M_g} \sum_{j=1}^{N_g} m_i)$. If the number of features in the model and image groups are not identical or if not all the features in groups come from a single object, then assuming at least one image group contains at least 4 features of a model group, a solution for the pose of the model object can be obtained by trying, in the worst case, all matches of four features within each pair of image and model groups. The number of matches that need to be tried in such case becomes $O(\sum_{i=1}^{M_g} \sum_{j=1}^{N_g} m_i^4 n_j^4 4!)$.
For small-sized groups (say, about 5 features each), this is essentially $O(M_g N_g)$ or linear in the number of groups [1].

From the above analysis, we see that in order to reduce the search involved in recognition, a grouping scheme must possess some desirable properties. Ideally, a grouping method must produce highly reliable (that is, groups coming from a single object) equal-sized groups in the model and image. If this is not possible, it must contain at least a sufficient number of features (four being the minimum) coming from a single object to make recognition possible. When groups satisfy this "minimum reliability", the number of extraneous features in a group must be as small as possible. In other words, it is desirable to have small-sized groups, so that the complexity of search remains linear in the number of groups. Another requirement to keep the number of matches small is to lower the number of possible groups (to a low-order polynomial). The number of groups cannot, however, be reduced by arbitrarily discarding groups as this could create unnecessary false negatives during recognition. That is, it may cause an object to be not recognized because groups corresponding to the model groups were discarded. Finally, since grouping is a pre-processing step to recognition, the group generation process (i.e., the algorithm for assembling the groups) itself must be fast and simple.

The above discussion suggests that one of the keys to making grouping useful for recognition is to group fea-

[1] This ignores the effort required for verifying a match assuming it is the same for recognition with or without grouping.

tures in an image based on constraints that capture some salient and easily detectable structures that point in turn to meaningful structures on objects in scenes. In this way, the number and size of groups can be kept small, since not all tuples of features will be meaningful, and being easily detectable, the groups can be generated by a fast and efficient algorithm. Finally, since such groups point to meaningful structures on objects in scenes, they tend to be more reliable.

2.1 Approaches to grouping

We now examine some of the previous work on grouping in vision in the light of the above requirements for their use in recognition. We will focus here on grouping of edge features, remarking on grouping methods for other data features only briefly. More extensive reviews of grouping are available elsewhere in literature [13], [17].

Grouping was initially studied in psychology, mainly as a perceptual phenomenon. There it was noticed that when we look at an edge image of a scene, we often pick up any structural information present in a collection of edges or lines. Figure 1 illustrates this with examples of line arrangements in which we can identify some perceptual structure. Early Gestalt psychologists demonstrated through a variety of examples that humans use cues such as simplicity, proximity, similarity, symmetry and familiarity for grouping features [33]. Their explanation for the perception of groups based on such cues seemed plausible but lacked a quantitative basis due to the difficulty in precisely defining concepts such as simplicity and familiarity. Later studies tried to make terms such as simplicity a little more concrete by using the concept of minimum entropy from information theory [10]. The explanations put forward by psychologists about this ability to group a collection of features based on constraints all seem to imply that it reflects an underlying knowledge of what makes a collection of edges come from a single object. In other words, the grouping process reflects an inherent bias towards collecting those edges that are likely to belong to a single object.

While the work on grouping in psychology had concentrated on observing it as a phenomenon and developing explanations for it, the work on grouping in computer vision has focused more on ways of making it useful for computer vision. Towards this end, several roles of grouping have been envisaged. In the early work of Marr, for example, grouping was suggested as a way of abstracting information in the raw primal sketch derived from the image [18]. He suggested grouping based on constraints such as curvilinearity, parallelism, and collinear displacements. Later work developed techniques to perform grouping such as the use of the Hough transform to capture collinearity information in points [8]. Grouping was also suggested as a useful step in both geometric and symbolic methods of recognition. Lowe proposed grouping as a way of establishing good primitives for recognition [17]. Jacobs and Clemens showed the extent of search reduction possible using grouping as a pre-processing step in recognition [5]. The role of grouping in geometric methods of recognition was merely to organize the data features, while the actual recognition was done by using the data features from the groups. The role of grouping in symbolic recognition methods, on the other hand, was to provide the groups themselves as high-level match primitives to be used directly for recognition using symbolic reasoning techniques. Early vision systems used grouping in this sense, such as ACRONYM in which edges were grouped into ribbons and the recognition of objects proceeded based on the ribbons and their topology [2]. More recently, grouping has been used for purposes of indexing into a library of objects in geometric methods of recognition [4], [15]. It has also been used for this purpose in symbolic methods of recognition to extract meaningful structures in scenes based on constraints of parallel and skew symmetry [26], [19] and proximity[7].

The role of grouping in extracting meaningful structures in scenes has also been emphasized in grouping schemes based on region and contour features (rather than edge or point features). This can be seen in the work of Shashua and Ullman on the grouping of image contours to capture salient curves [25], of Dolan and Weiss on the grouping of curved lines using proximity [7], and of LeClerc on hierarchical grouping of regions based on the minimum description length principle [16].

A class of approaches in computer vision have attributed the tendency to group features to the ability of humans to recognize the non-accidental occurrence of the relation underlying the groups. That is, the degree to which a relation is unlikely to have arisen by an accident of viewpoint, rather than the knowledge that they belong to a single object, is the motivation behind grouping features based on that relation. This was concluded by Witkin and Tenenbaum [34] after observing that such non-accidentalness was used to interpret groups of features even when the ultimate interpretation of the groups was not known. They also pointed out that since such a relation was expected to remain stable over a large number of viewpoints, it must reflect some meaningful structure in the scene. Lowe extended the non-accidentalness argument behind grouping to identify the set of image relations that are unlikely to occur by an accident of viewpoint [17]. Using the assumption that the viewpoint of the camera is independent of the objects in the scene, he showed that only certain image relations, such as convexity and parallelism, are likely to remain stable over a large range of viewpoints. He also concluded that because of this viewpoint invariance, the detection of such relations in an image implied that they were likely to be the projection of a meaningful and specific 3d structure. Lowe showed that this property can make such groups useful for the recognition of three dimensional objects. For example, using the non-accidentalness of viewpoint, he showed that parallel lines in the image are most likely to come from parallel lines in space. So if the model object contained parallel lines, then this justifies the matching of (groups of) parallel lines in the image to (groups of) parallel lines on the model, thus making such groups useful primitives for recognition.

Another class of approaches in computer vision explored the same argument for grouping that was put forward by psychologists based on the likelihood of the

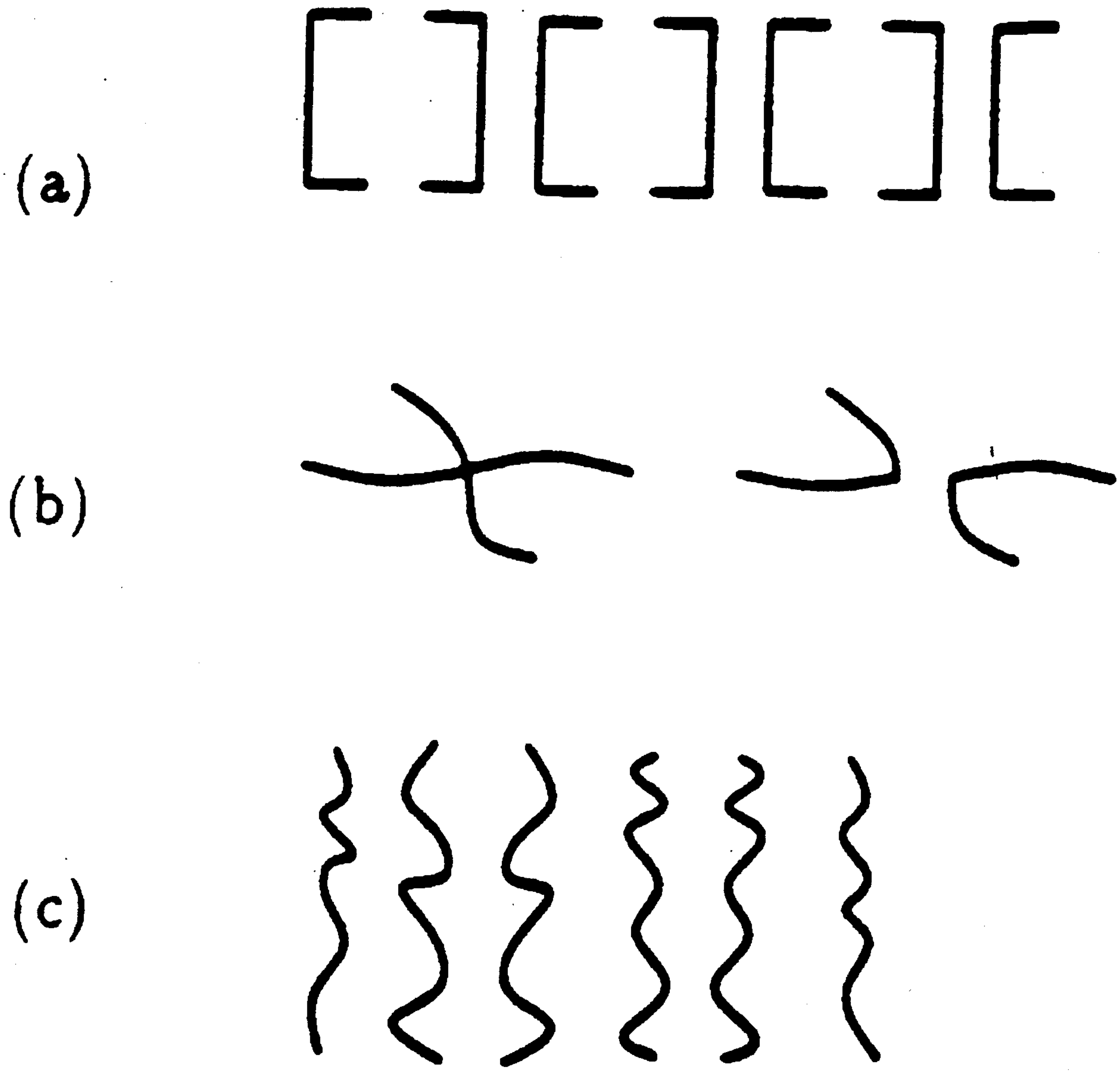


Figure 1: *Illustration of perceptual structure apparent in line arrangements. (a) A group of lines perceived as a collection of squares due to closure (or continuation). (b) Lines seen as crossing due to good continuation. (c) Bilateral symmetry evident from the group of lines shown. (Adapted from Figure 2-1 of [17])*

grouped features coming from a single object. For example, in the early work of Roberts, vertices connected by straight edges were grouped using the rationale that connected vertices were likely to be part of the same object [22]. This argument was given a more quantitative basis by Jacobs who proposed that grouping of data features in an image should be based on a relation that points to the likelihood of such features coming from a single object [13]. Specifically, he used distance and orientation constraints to explore the convexity relation between a group of edges. By doing a statistical analysis of occlusions and merging of edges with background in images, he showed that it is unlikely for a randomly selected group of edges to form a convex polygon, thereby implying that the detection of such a relation between edges pointed to their likelihood of coming from single objects. Other researchers that have used a similar argument for grouping are Bolles and Cain who used the proximity relation to groups features in their local feature focus method [1], Brooks who grouped edges forming ribbons or trapezoids [2], and Clemens who grouped edges enclosing open regions [3].

Let us now evaluate some of the existing schemes for grouping from the point of recognition. Grouping schemes, such as that of Shashua and Ullman [25] for grouping image contours, that attempt to capture meaningful structures in scenes without reasoning about scene geometry, often produce groups that span wide areas of the image, making them unreliable for recognition. Other grouping schemes based on viewpoint invariance that use constraints or relations that are likely to hold over a wide range of viewpoints such as parallelism [17], or convexity[13], [11] also produce groups that attempt to capture meaningful structures in the scene. However, the ease with which such relations are detected in images decides the number of groups generated as well as their size. Since several nearby edges could satisfy relations such as parallelism and convexity, different combination of edges have to be explored by grouping algorithms leading to a very large number of groups and taking time of equal complexity. For example, Huttenlocher grouped edges based on connectivity by considering all possible sequences of edges of length three (leading to $O(N^3)$ groups for N edges)[12]. Later work on grouping tried to generate a smaller number of groups by filtering some of the groups. In Huttenlocher and Wayner [11] for example, a grouping algorithm was presented that works in $O(n\ logn)$ time and generates a linear number of convex edge groups. The filtering was done by using a cost function to rank neighbors of an edge and allowing only the least-cost neighbor to participate in a convexity relation. Although the number of groups are restricted by this method, it is not clear whether such decisions can be made on a purely local basis. Also, since there is no analysis of the kind of groups that will be missed, it is not clear that such groups do not cause a recognition system to make false negative identifications. Another grouping scheme explored by Clemens also restricts the number of groups to be linear in the number of edges [3]. Here the groups are designated by open regions that are enclosed by a group of edges. Such open regions of the image were considered likely to come from a single object because a transition from one object to another almost always caused a change in intensity sufficient to produce an edge that splits a region. The grouping algorithm used assigns an edge to at most four regions thus ensuring that the number of groups remains linear in the number of edges. Due to feature instabilities, imaging artifacts, etc. an open region is rarely bounded by edges forming a complete connected closure, causing such assignments of edges to 4 neighbors using purely local judgment to group together edges that do not necessarily come from the single object. Thus in the existing approaches to grouping, it appears that restricting the number of groups may either cause some relevant groups to be missed or may make the grouping scheme unreliable, causing it to group features that don't necessarily come from a single object.

3 Grouping Based on Closely-Spaced Parallelism

We now present a grouping method that exploits the relation of closely-spaced parallelism commonly occurring between lines on objects, to produce groups that possess many of the desirable properties for purposes of recognition. Many commonly occurring objects in indoor scenes such as books, cups or tables possess some pattern-like structures that often attract our attention. Such structures usually contain groups of closely-spaced parallel lines of a few orientations. For example, printed letters on the surface of an object such as a book, or a bottle, and wooden texture on pieces of furniture such as a table contain groups of closely spaced parallel lines. Sometimes such parallel lines form texture-like patterns as on the bottle in Figure 2a, while in other cases they capture some interesting structures from parts of objects such as the parallel contours in the triangular block of Figure 2a. Even when they can be treated as textures we consider them as a separate cue since the property of parallelism they capture is of direct use in recognition as a grouping method[2]

The groups of parallel lines we want to capture include cases of both explicit and implicit parallelism. Figure 2a shows a scene containing objects showing instances of both types of parallelism. The contour of the triangular block has two explicitly parallel lines as can be seen from Figure 2b, while the letter texture on the bottle has implicit parallelism as can be seen from the group of parallel lines in Figure 2c where only the nearly horizontal lines of Figure 2b are highlighted. As we will see later, the projection of such patterns in images continue to show closely-spaced parallelism among the projected lines over a wide range of viewpoints. This makes it possible to capture closely-spaced parallelism on objects by

[2] Also, in the context of building the attentional selection model, while color and texture have been primary features being extracted directly from the intensity image, parallel-line groups serve as a secondary feature being extracted from the edges or line features. So treating them as a separate cue illustrates an implementation of the model of attentional selection possessing a feature hierarchy.

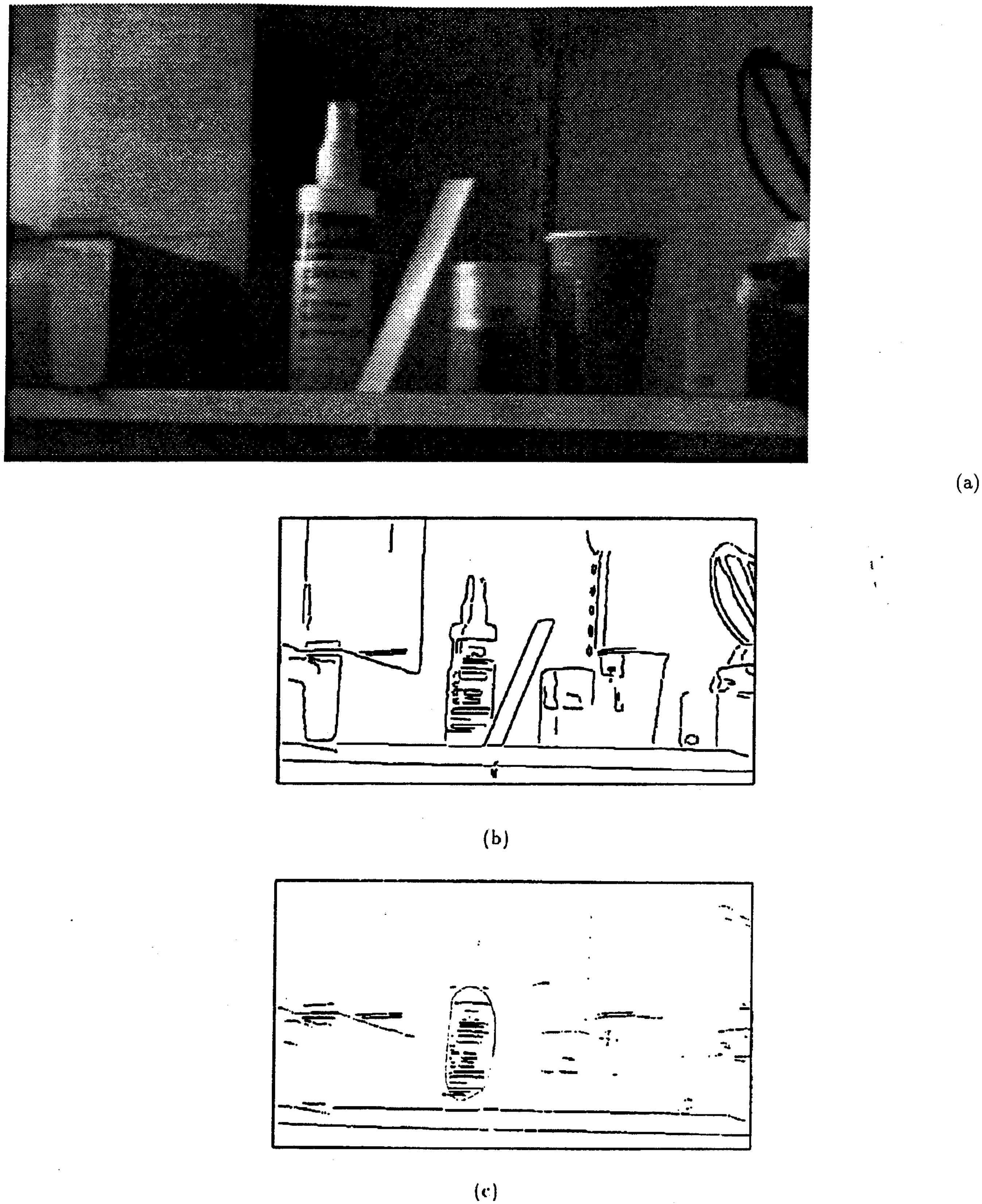


Figure 2: *Illustration of implicit and explicit closely-spaced parallelism on objects. (a) An image of a scene containing objects showing explicit and implicit parallelism. (b) Line segment image of (a). Note the parallelism explicit in the contour of the triangular block. (c) An image showing only the nearly horizontal lines in the image of (b). Note that the parallelism implicit in the letter texture on the bottle in (b) becomes explicit in this image.*

examining such a relation between the edges (or lines) in their projections. Since it is rare that adjacent object regions in an image possess similarly-spaced parallel lines of similar orientation, the detection of closely-spaced parallel lines in images is also likely to point to single objects. Further, since not all edges in the image are likely to show closely-spaced parallelism, this could lead automatically to fewer groups. And for objects showing characteristic textural information that contains such closely-spaced parallel-lines, the groups can be useful clues that point to the identity of such objects. Also, when the spacing allowed between parallel lines is small, such groups capture compact areas in both the image and the object and contain fewer features in a group. Finally, as we will see later, such groups can be easily found in images using a simple algorithm. Thus groups of closely-spaced parallel lines in images capture not only meaningful structures on objects in scenes but also possess the desirable properties required of a grouping scheme for recognition.

Grouping based on such parallelism, however, has the disadvantage that unlike in conventional grouping, a single pair of matching groups is not sufficient for recognition. This is because recognition methods such as the linear combination of views-based alignment method require at least 4 non-coplanar points for alignment. Since a group of parallel lines in space span a plane, the features such as points or lines derived from them are coplanar, needing at least two matching pairs of groups to be found. However, since more than four corresponding features are needed in practice, other grouping schemes have also found the need for finding more than a pair of matching groups [13].

3.1 Closely-spaced parallelism constraint

So far we have only loosely specified the property of closely-spaced parallelism and have given intuitive arguments about the advantages of grouping based on this relation. We now make the definition more precise to allow the generation of groups from line segments in an image based on this relation. Ideally, the structure in space we want to capture using the closely-spaced parallelism constraint is a set of (3D) parallel line segments on an object with a given inter-line spacing. To see how such a structure appears in an image (i.e. in a projection), we exploit some well-known results in descriptive geometry [14]. These results indicate that under orthographic projection and scale (often used to approximate perspective projection), a parallel-line group in 3D always projects to a group of parallel lines in the image under any view. In practice, because of the noise in the imaging process, and depending on the method used to obtain line segments from edges, such a group appears as a set of closely-spaced lines with slight skew between the lines but with the overall orientation of the group remaining more or less uniform. When perspective effects are dominant, however, parallel lines in 3D appear as a set of converging lines. For most imaging distances, this convergence is slight, so that such lines have only a small amount of inter-line (as well as overall) skew. Thus 3D line segments on objects showing strict closely-spaced parallelism between them actually appear as groups of closely-spaced lines in image that are almost parallel (i.e. with slight inter-line as well as overall skew). However, we will refer to such groups in both the image and object as closely-spaced parallel-line groups (or in short as line groups) with the implication of strict parallelism between lines in 3D and approximate parallelism between their projections.

To precisely define such groups in an image, we begin with some terminology relating to 2d non-intersecting line segments.

3.1.1 Terminology

1. Overlapping lines: Two line segments are said to overlap if the projection of at least one end point of one of the lines lies inside the other line segment. Figure 3a shows examples of overlapping and non-overlapping line segments.

2. Across-the-line-distance: The across-the-line distance between two lines L_1, L_2 whose end points are designated by $p_{l11}, p_{l12}, p_{l21}, p_{l22}$ respectively, is defined as follows. Let S denote the set of pairs $\{(p_{l11}, p_{l21}), (p_{l11}, p_{l22}), (p_{l12}, p_{l21}), (p_{l12}, p_{l22})\}$. Let $d_{min} = \min \{d(p_i, p_j) | \forall (p_i, p_j) \in S\}$, where $d(p_i, p_j)$ is the euclidean distance between the points of the pair (p_i, p_j). Let (p_r, p_s) be the pair in S that has this minimum distance d_{min}. Let $L(p_r)$ and $L(p_s)$ be the lengths of the projection of points p_r and p_s onto lines L_2 and L_1 respectively. Then the across-the-line distance d_{across} between lines L_1 and L_2 is defined as

$$d_{across} = \begin{cases} min\{L(p_r), L(p_s)\} & \text{if } L_1 \text{ and } L_2 \text{ are overlapping} \\ d_{min} & \text{otherwise} \end{cases} \tag{1}$$

Figure 3b shows examples of some non-intersecting line segments and the across-the-line distance between them.

3. Along-the-line-distance: The along-the-line distance d_{along} between two lines L_1 and L_2 is defined as:

$$d_{along} = \begin{cases} \min\{\sqrt{(d_{min}^2 - L(p_r)^2)}, \sqrt{(d_{min}^2 - L(p_s)^2)}\} & \text{if } L_1 \text{ a} \\ 0 & \text{otherw} \end{cases} \tag{2}$$

where the terms d_{min}, $L(p_r)$, and $L(p_s)$ are as given in Definition 2. Figure 3c shows some non-intersecting line segments and the along-the-line distance between them.

3.1.2 A closely-spaced parallel-line group

A closely-spaced parallel-line group in the image, specified by the tuple
$< t_{across}, t_{along}, t_{local-orient}, t_{global-orient} >$,
is the largest group of non-intersecting line segments such that for each line in the group, there exists another line in the group obeying all of the following constraints:

1. The across-the-line distance d_{across} between the lines is no more than the threshold t_{across}.
2. The along-the-line distance d_{along} between the lines is no more than the threshold t_{along}.

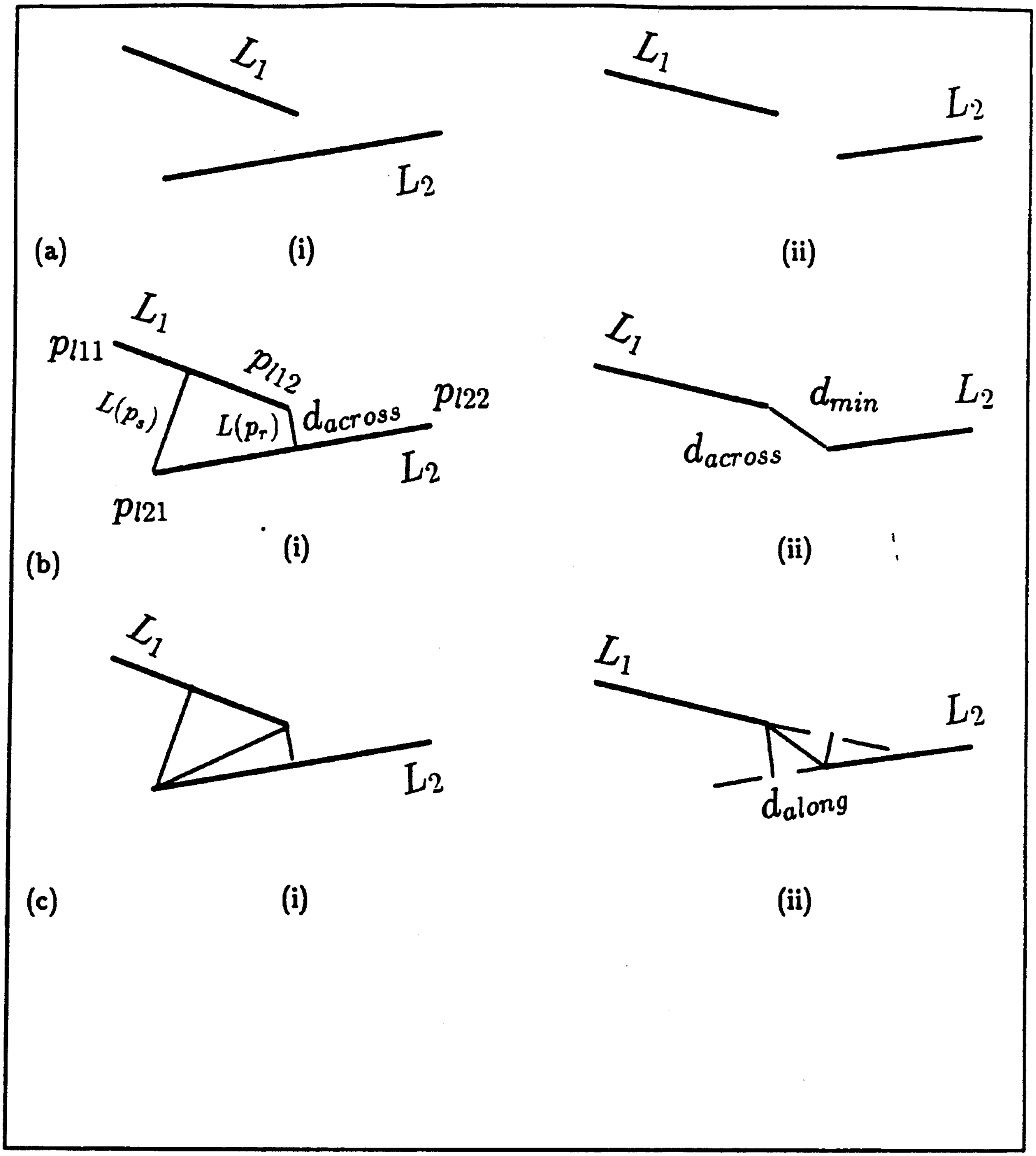


Figure 3: *Illustration of some of the terminology relating to 2D non-intersecting line segments. (a) The difference between overlapping (i) and non-overlapping (ii) line segments according to Definition-1 in the text. (b) Across-the-line distance shown for both overlapping (i) and non-overlapping (ii) line segments. (c) Along-the-line distance shown for both overlapping (i) and non-overlapping (ii) line segments.*

3. The orientation difference between the lines is no more than a threshold $t_{local-orient}$.

Moreover the entire group must satisfy the condition that the maximum orientation change between any two lines in the group is no more than a threshold $t_{global-orient}$.

The above definition of closely-spaced parallelism allows for almost parallel lines to be grouped, which as we said above, is a more useful structure to capture in the image. Further, by allowing non-overlapping lines to be grouped, it can not only capture groups of non-overlapping parallel lines in space, but also allows some occlusions that cover portions of line segments, to be handled. The constraint on global orientation change, $t_{global-orient}$, is imposed to keep the entire group almost parallel since otherwise successive deviations in orientation between lines could lead to a group of fairly skewed lines. This also makes the above grouping constraint different from the one used earlier for assembling groups of parallel lines in a data-driven fashion [21]. Finally, the choice of the thresholds $t_{across}, t_{along}, t_{local-orient}, t_{global-orient}$ dictates the kind of groups that will be generated. We will discuss their choice when using the groups to perform data and model-driven selection.

3.2 Algorithm to generate the line groups

We now present an algorithm to generate closely-spaced parallel line groups in an image for a given choice of thresholds $t_{across}, t_{along}, t_{local-orient}$, and $t_{global-orient}$. It works by first extracting line segments from edges in an edge image using one of the standard algorithms for line-segment approximation [20]. The resulting line segments are used to generate the groups as follows:

1. Each line segment is initially kept in a separate group.
2. For each line segment L, the following operations are done:

   (a) A rectangular neighborhood about L that is $2t_{across}$ in breadth and $2(t_{along}+l)$ in length, where l is the length of the line, is scanned, and all lines that either pass through this neighborhood or have an end point in it are retained.

   (b) Among the lines obtained in step-2a, those that satisfy the local orientation change constraint $t_{local-orient}$ with L are retained.

   (c) A new group is formed by successively merging the enclosing groups of lines obtained after step 2b with the enclosing group of L taking care to see that no enclosing group being added contains a line violating the $t_{global-orient}$ constraint with the currently created group.

3.2.1 Analysis

The grouping algorithm performs steps 1-2 using the union-find data structure to record and update information about line groups [6]. In this data structure, information is organized as a forest of trees. The essential information within a tree is summarized in its root. The basic operations that can be performed on this data structure are ***make-set (a)*** that creates a single node tree with element $\boldsymbol{a}$, ***find (a)*** that finds the root of the tree containing $\boldsymbol{a}$, and ***union(a,b)*** that merges the trees containing elements $\boldsymbol{a}$ and $\boldsymbol{b}$. Using a technique called merging by rank with path compression [6], it is known that m operations of make-set take time $O(m)$, of find take time $O(m)$ while m union operations take time $O(mA(m,n))$ where n is the number of elements in the data structure, and $A(m,n)$ is the Ackerman's function. For most values of m and n, the function $A(m,n)$ is almost constant so that a single one of these operations can be done in constant amortized time.

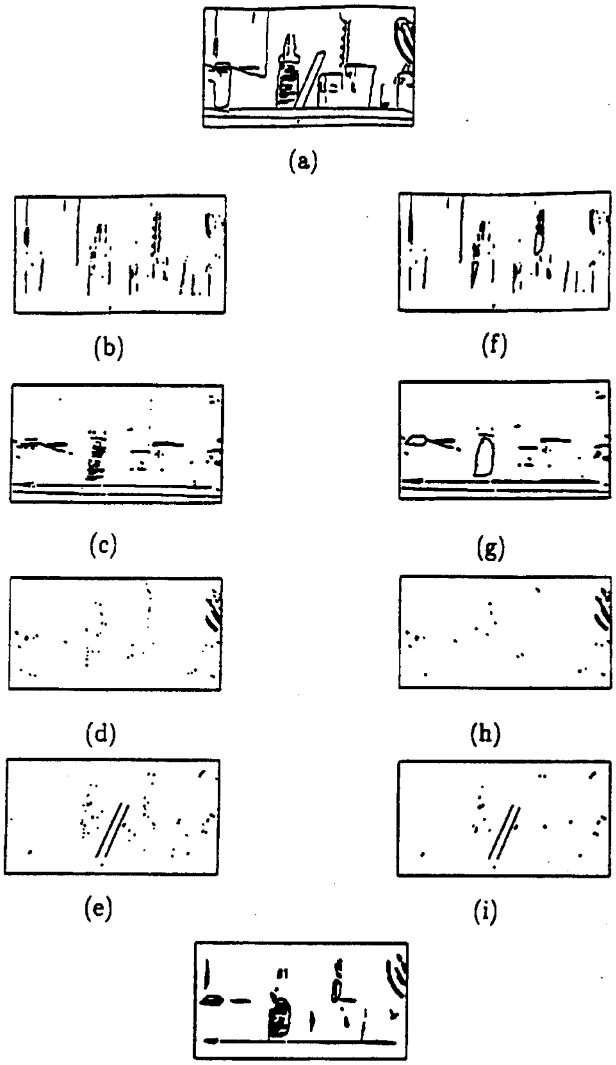
Using the union-find data structure, Step-1 requires n make-set operations for n line segments. For each line L, Step 2a requires all lines to be scanned requiring $O(n)$ time. Similarly Step 2b requires $O(n)$ time, in the worst case, to examine all the retained lines. If the least orientation in a group is stored as part of the information in the roots of trees, then the constraint checking in Step 2c can be done by a simple find operation per line. Finally, the merging in Step 2c can be done by a union operation. Thus the entire step 2 can be done in time $O(n)$ per line with the result that the grouping algorithm itself runs in $O(n^2)$ worst-case time.

3.2.2 Results

We now illustrate the grouping algorithm with a few examples. Figure 4a shows the line segments obtained by doing a line segment approximation to the edges in the image of Figure 2a. The closely-spaced parallel line groups obtained using the grouping algorithm with a constraint specification of $< 10, 5, 6, 10 >$ are shown in Figures 4f-i. These groups are shown along four major orientations (vertical, horizontal, obtuse, and acute) for clarity. The individual groups are highlighted by drawing the convex hull of the end points of line segments. The line segments that are grouped can be seen in the corresponding Figures 4b-e. Similarly, Figure 5 shows another example of grouping performed by the algorithm. By using the algorithm on a number of edge images, the number of groups, their average size (number of constituent lines) and the average area spanned by the groups were recorded. The results are shown in Table 1. From the table it can be seen that the number of groups is linear in the number of line segments, and the size of the line groups tends to be small.

3.2.3 Discussion

The number of groups generated by the grouping algorithm is in fact linear in the number of lines, since each line belongs to at most one group at the end of Step 2. If the constraints did not involve $t_{global-orient}$, it is clear that only a linear number of groups would have been possible (recall that we are considering only the largest such groups). With the fourth constraint $t_{global-orient}$ added, the starting line as well as the order in which lines are examined determines the lines that ultimately belong to



(a)

(b)

(f)

(c)

(g)

(d)

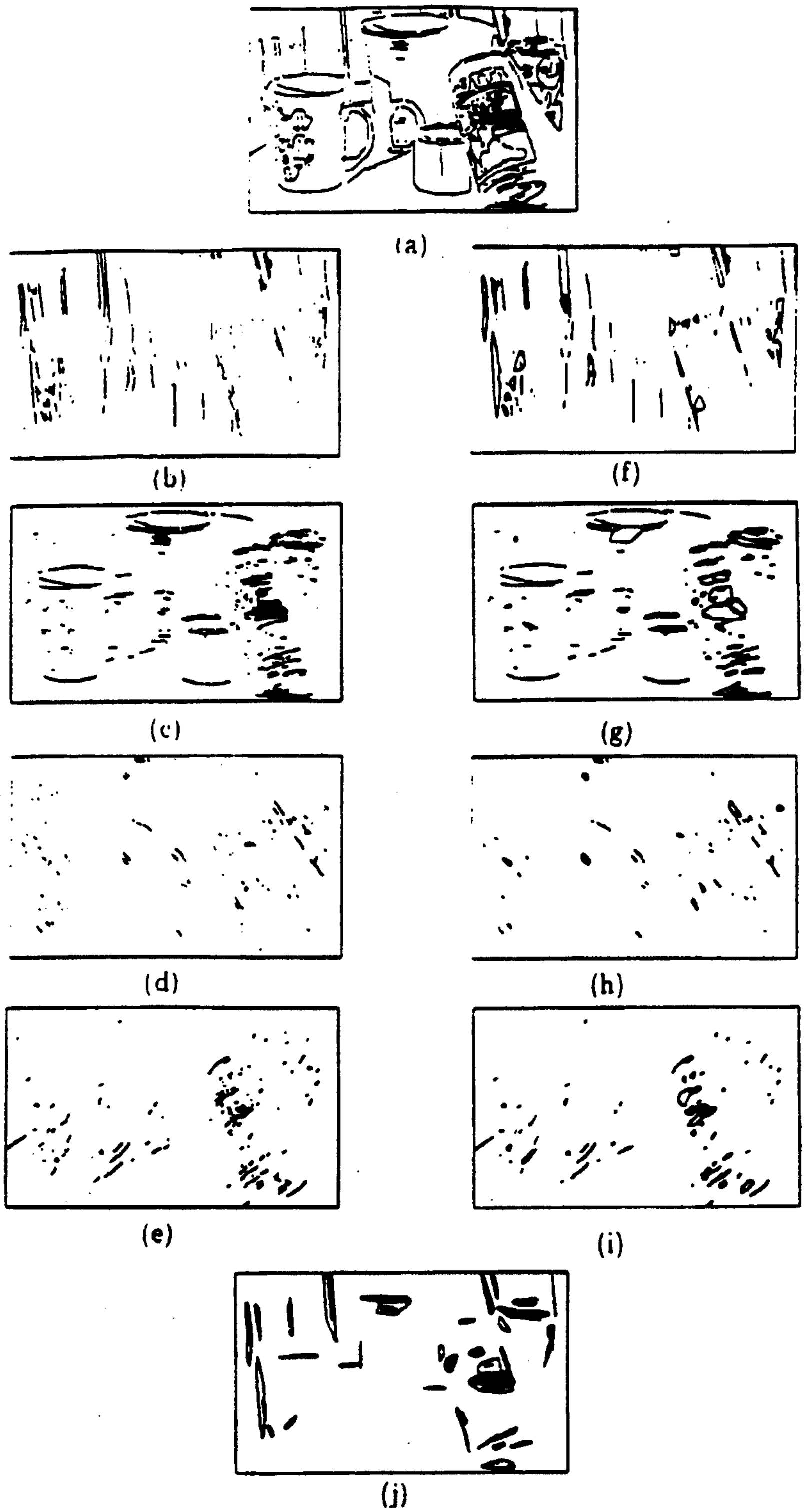
(h)

(e)

(i)

(j)

Figure 4: *Illustration of grouping based on closely-spaced parallelism and salient group detection. (a) Line segments to be grouped based on closely-spaced parallelism. (b)-(e) Line segments shown along four major orientations, namely, vertical, horizontal, obtuse, and acute orientations. (f)- (i) The line groups formed using the algorithm shown also along the respective major orientations for clarity. The thresholds used were* $t_{across} = 10, t_{along} = 5, t_{local-orient} = 6^o, t_{global-orient} = 10^o$. *(j) The 40 most salient groups among the line groups of (f) - (i) found using the saliency measure. Note that none of the salient groups span more than one object.*


(a)

(b)

(f)

(c)

(g)

(d)

(h)

(e)

(i)

(j)

Figure 5: *Illustration of grouping based on closely-spaced parallelism and salient group detection — Another example. (a) Line segments to be grouped based on closely-spaced parallelism. (b)-(e) Line segments shown along four major orientations, namely, vertical, horizontal, obtuse, and acute orientations. (f)- (i) The line groups formed using the algorithm shown also along the respective major orientations for clarity. The thresholds used were* $t_{across} = 6, t_{along} = 5, t_{local-orient} = 6^\circ, t_{global-orient} = 10^\circ$. *(j) The 40 most salient groups among the line groups of (f) - (i) found using the saliency measure. Note that only two of the salient groups span more than one object.*

S.No.	Image	Num.	Group	Num.	Avg. Group	Max. Group	Avg. Group
.	Size	Lines	Constraints	Groups	Size	Size	Area
1.	320 x 576	395	$< 10, 5, 6, 10 >$	34	4	21	0.002
2.	256 x 416	756	$< 6, 5, 6, 10 >$	91	3.0	11	0.001
3.	240 x 240	233	$< 5, 0, 6, 10 >$	22	3.1	6	0.0009
4.	232 x 576	884	$< 5, 5, 6, 10 >$	119	3.9	7	0.0008
5.	200 x 492	1232	$< 5, 10, 9, 12 >$	243	3.77	13	0.0028
6.	224 x 416	316	$< 5, 5, 6, 10 >$	75	3	17	0.003

Table 1: *Characteristics of closely-spaced parallel line groups generated by the grouping algorithm. The average group area is normalized with respect to the image size. Only groups containing more than one line are considered here.*

a group as well as its size. In such cases, more groups than are generated by the algorithm are possible. A case where this happens is shown in Figure 6. Figure 6a shows an arrangement of closely-spaced parallel lines in the image and Figure 6b shows the groups that will be generated by the algorithm. Finally, Figure 6c shows some other groups that are possible from the arrangement in Figure 6a but are not generated by the algorithm. The groups generated by the algorithm correspond to a left to right, bottom to top scan of the line segments in the image. Such a scan often produces groups that resemble the groups we perceive using a frame of reference with the origin at the left hand bottom corner of the image.

In general, if a large number of lines fall within the specified neighborhood of a line (in Step 2a), the possible combinations of lines obeying all 4 constraints could become very large. The grouping algorithm described above generates only a subset of such groups, and in some sense, therefore, does a filtering operation. We mentioned earlier in Section 2.1 that grouping approaches that filtered groups to keep them to a small number could cause a recognition system that subsequently uses these groups to make unnecessary false negatives. We now show that this does not happen with the above grouping algorithm. For this, we notice that the closely-spaced line groups satisfying all 4 constraints of $\{t_{across}, t_{along}, t_{local-orient}, t_{global-orient}\}$ are subsets of groups satisfying the first 3 constraints $\{t_{across}; t_{along}, t_{local-orient}\}$ (called main groups here). The grouping algorithm generates only some of the possible subsets, but such groups (called aggressive groups, henceforth) generate a cover of the main group. That is, every line of a main group belongs to some aggressive group. Suppose that the groups are fed to a recognition system. Assuming an alignment style of recognition (such as the linear combination of views method [32]), we know that at least 4 matching features must be found to solve for the pose of the object. Since parallel line groups in the image that come from parallel lines in space represent coplanar points, we may need two such groups to derive these features. Let us assume that each group provides two features and that the features are derivable from a single line in each of the groups (the end points of a line are the features, say). If there existed a pair of closely-spaced parallel-line groups in the image obeying all 4 constraints that were the correct pair of groups (i.e., they contained sufficient number of features to recognize the object) but were not generated by the grouping algorithm, then a recognition system using the groups given by the algorithm could make false negatives. But since the aggressive groups form a cover, each correct group has partial overlap with at least one aggressive group suggesting that those pairs of aggressive groups would also be the correct groups containing sufficient features to recognize the object thus preventing a false negative identification.

Thus the above grouping algorithm keeps the number of groups small by filtering possible groups, but at the same time, prevents unnecessary false negatives during recognition due to insufficient number of groups being produced.

4 Data-driven Selection using Line Groups

We now discuss the use of closely-spaced parallel-line groups to perform data-driven selection. The goal of data-driven selection is to isolate regions in an image that are likely to come from a single object based on information available in the image and some a priori knowledge about scenes. For a given choice of thresholds, not all the groups generated by the above algorithm represent useful structures in the scene as can be seen from the examples in Figures 4 and 5. Some of the groups may span more than one object, while others come from spurious line segments, or scene clutter rather than objects of interest in the scene. For the purposes of recognition, it would be useful to order and consider only some of the more reliable ones from these groups. In keeping with our general paradigm of data-driven selection, we order the groups using a saliency measure and select a few of the salient groups. In this section, therefore, we describe a measure of saliency for the line groups and then discuss the utility of salient group-based selection in recognition.

4.1 Saliency of parallel-line groups

As in the development of color and texture region saliency, the focus in designing a measure of saliency of parallel-line groups will be on capturing the sensory component of distinctiveness. Thus those properties of lines that are commonly perceived and fairly general will be considered. The strategy for assembling the saliency measure is, as before, to record the factors affecting saliency and to combine them appropriately in a way that reflects their importance. Unlike in the case of color and texture saliency, however, the saliency measure for

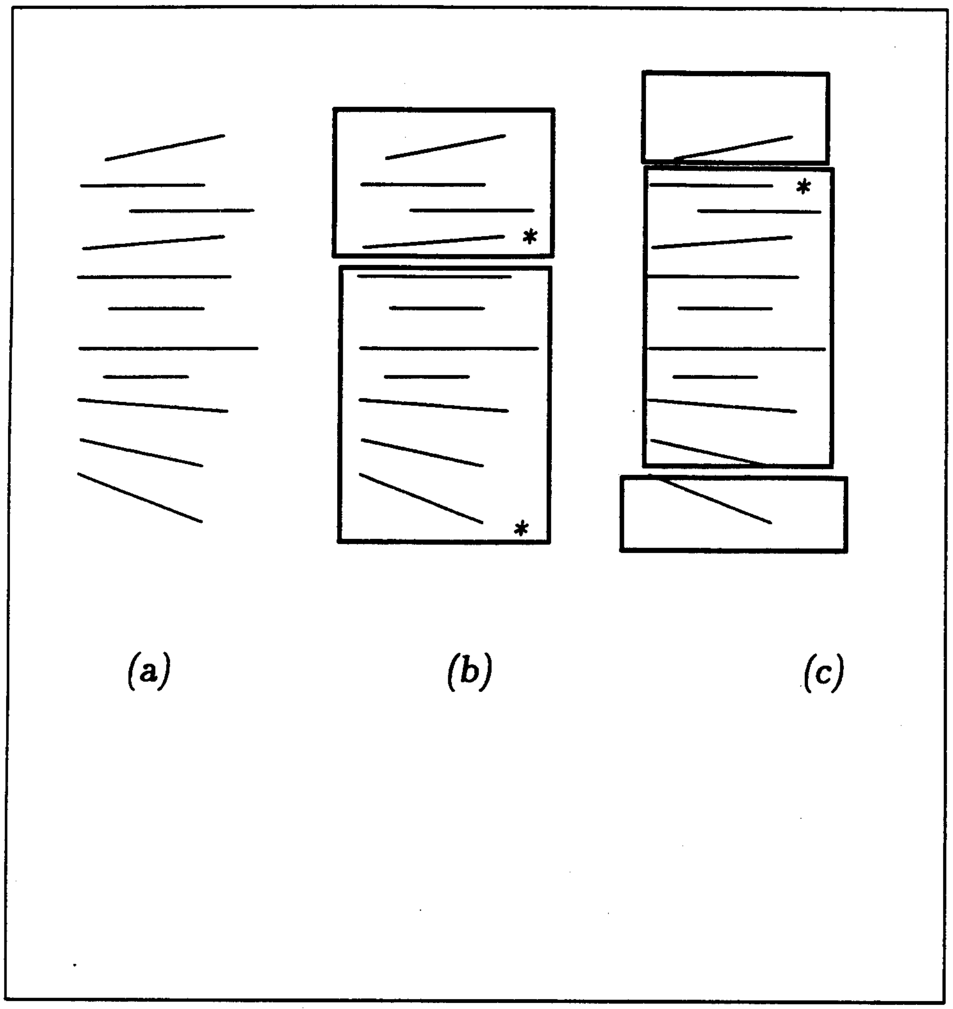


Figure 6: *Example to illustrate some of the line groups that are not generated by the grouping algorithm. (a) An arrangement of closely-spaced parallel lines. (b) The groups generated by the algorithm shown within the two rectangular boxes. The asterisk mark indicates the starting line segment used to assemble the line groups. (c) Another set of groups possible from the arrangement of (a) that also obey all the four grouping constraints. Note the overlap between these groups and those generated by the grouping algorithm.*

line groups is designed to emphasize the inherent reliability of groups more than the match with our perceptual judgment of their importance.

4.1.1 Factors affecting saliency of line groups

Among the properties of line-groups that are often observed are the length of the constituent lines, the region spanned by them, the number of constituent lines, and their overall orientation span. These properties of line groups were chosen as the factors affecting saliency using the following rationale:

1. Length of constituent lines (L): The length of the constituent lines is chosen as a factor because groups with both very long or very short lines are undesirable from the point of recognition. Very long lines in an image are more likely to span multiple objects, while very short lines often tend to be due to spurious line segments resulting from scene clutter or from a very fine line-segment approximation. Since a group has lines of varying length, the average length of lines in the group is taken as a measure of the length of the group.

2. Region span of a group (R): The region spanned by a line group can give indications of its reliability. A large region span often indicates a group spanning more than one object. We measure the region spanned by a closely-spaced parallel-line group by the area of the convex hull of the end points of the constituent lines.

3. Number of constituent lines (N): Groups with a very large number of lines may not be entirely desirable from the point of recognition as they could contain a large number of features. Also, such groups could potentially span multiple objects. A sparse group (with one or two lines), on the other hand, may not indicate any meaningful structure. Either way, the number of constituent lines in a group can affect its saliency.

4. Orientation span (Θ): A low orientation span indicates a greater amount of parallelism in a group. Since the aim in grouping lines here is to capture instances of closely-spaced parallelism on objects in a scene, groups exhibiting a greater degree of parallelism are more likely to indicate a meaningful structure. This also follows from the viewpoint invariance argument of Lowe that we mentioned in Section 2.1, namely, that a group of parallel lines is unlikely to have arisen from an accident of viewpoint, thereby making them more likely to come from either a single object or a single structure. The orientation span is measured by recording the maximum difference in orientation between lines in a group.

4.1.2 Weighting functions for factors affecting saliency

To develop a measure of saliency for the line groups, each of the factors must be weighted to appropriately reflect their individual contributions in deciding the saliency of a group. The form of the weighting function, in most cases, was derived using three criteria: It should (a) reflect the likelihood of a group coming from a single object, (b) it should be a smooth function so that discontinuities do not indicate an abrupt change in judgment, (c) it should be verifiable from statistical experiments. The weighting functions chosen for the factors are as follows:

1. Length of constituent lines (L): Using the rationale given earlier, the weighting function is chosen to de-emphasize both very long and very short lines, while giving about equal importance to intermediate length lines. The decision of very short lines is made on an absolute basis, i.e., lines shorter than 5 pixels are considered very short, while long lines are decided relative to the size of the image (by choosing the diagonal length in the image as the normalizing factor). The weighting function is as follows:

$$f_1(L) = \begin{cases} -\frac{ln(1-L_n)}{c_1} & 0 \leq L \leq l_1 \\ 1 - e^{-L_n c_2} & l_1 < L \text{ and } L_n \leq l_2 \\ 1.0e^{-c_3(L_n - l_2)} & l_2 < L_n \leq 1.0 \end{cases} \qquad (3)$$

where $L_n = \frac{L}{L_{max}}$, L_{max} = diagonal length of the image, and the various thresholds are $l_1 = 5$ (pixels), $l_2 = 0.4$, $c_1 = -\frac{ln(1-\frac{l_1}{L_{max}})}{l_2}, c_2 = \frac{6ln10}{l_2}, c_3 = \frac{3ln10}{1-l_2}$.

The form of the weighting function was derived by performing the following experiments. Groups were formed from several edge images using the grouping algorithm and the average line lengths of groups were recorded. The scenes of the images varied in complexity having different amounts of scene clutter, contained several objects, and showed illumination artifacts such as specularities, interreflections, etc. Figures 2a and 5a shows examples of some typical images tried. A histogram of the number of groups with a given normalized length L_n (using 200 bins) was plotted. From this, the number groups that came from a single object and had normalized line length falling in a given bin were noted. The ratio of the number of groups of a given L_n coming from a single object to the total number of groups of that line length was taken to represent the weighting function $f_1(L)$. This ratio was plotted against L_n and smooth functions were fit to the resulting curve. These functions were described by the parameters c_1, c_2, c_3. Finally, the thresholds l_1, l_2 were found from the breakpoints in this ratio curve.

2. Region span of a group (R): The weighting function for the region span was chosen to emphasize small and

compact groups. The form of the weighting function was derived by performing studies similar to the one described above. Here, the ratio of the number of groups with a given normalized region span that came from a single object to the total number of groups with the given span was taken to represent the weighting function. The weighting function derived from this ratio was:

$$f_2(R) = \begin{cases} -\frac{ln(1-R_n)}{c_4} & 0 \leq R_n \leq r_1 \\ 1 - e^{-c_5 n} & r_1 < R_n \leq r_2 \\ s_2 - c_6 ln(1 - R_n + r_2) & r_2 < R_n \leq r_3 \\ s_3 e^{-c_7(R_n - r_3)} & r_3 < R_n \leq r_4 \\ 0 & r_4 < R_n \leq 1.0 \end{cases} \tag{4}$$

where $R_n = \frac{R}{R_{max}}$, and R_{max} = image size, $r_1 = 0.1$, $r_2 = 0.4$, $r_3 = 0.5$, $r_4 = 0.75$, $s_1 = 0.8$, $s_2 = 1.0$, $s_3 = 0.7$, $s_4 = 10^{-3}$ and $c_4 = -\frac{ln(1-r_1)}{s_1}, c_5 = -\frac{ln(1-s_1)}{r_1}, c_6 = -\frac{(s_2-s_3)}{ln(1+r_2-r_3)}, c_7 = -\frac{ln\frac{s_4}{s_3}}{(r_4-r_3)}$. Here again the thresholds are chosen in a manner similar to the one described for the weighting function for the length of constituent lines.

3. Number of constituent lines (N:) Since the average size of a group is small in the groups generated by the grouping algorithm, the weighting function is chosen to emphasize densely packed groups, i.e. groups with a larger number of constituent lines, as they often indicate some textural information. If such groups span a large area then they will be de-emphasized in the weighting function $f_2(R)$ for the region-span. The weighting function chosen was:

$$f_3(N) = \frac{N}{N_{max}} \tag{5}$$

where N = number of constituent lines in a group, and N_{max} = maximum number of lines in any line group in the given edge image.

4. Orientation span (Θ): Using the rationale given earlier, the weighting function here is designed to emphasize groups showing a greater degree of parallelism, i.e., a smaller orientation span. To avoid unfair bias towards groups of single lines (which will have an orientation span of zero), we assign a small penalty toward single line groups. The resulting choice of function is:

$$f_4(\Theta) = \begin{cases} 0.1 & \theta = 0 \text{ and } N = 1 \\ (1 + \frac{\theta(1.0 - ct_1)}{t_{global-orient}}) & \text{otherwise} \end{cases} \tag{6}$$

where Θ is the orientation span, and ct_1 =0.4.

4.1.3 Saliency measure for a closely-spaced parallel-line group

The saliency measure for a closely-spaced group of line segments is obtained by combining the weighting functions reflecting the contributions from the various factors. Since the factors record independent properties of line groups, we chose to combine them linearly to give the following saliency measure:

$$\text{Saliency of a line group} = f_1(L) + f_2(R) + f_3(N) + f_4(\Theta) \tag{7}$$

4.1.4 Results

We now illustrate the use of the saliency measure to judge the reliability of closely-spaced parallel-line groups. Figures 4f-i show the line groups found by the grouping algorithm in the image of Figure 4a. Among these, the 40 most salient groups found using the above saliency measure are shown in Figure 4j. As can be seen from the figure, all of the 40 salient groups come from single objects. Figure 5 shows another example in which the line groups generated are shown in Figures 5f-i and the top 40 salient groups among them are shown in Figure 5j. Here only two of the salient groups did not come from single objects. Table 2 shows the results of performing saliency experiments on a number of images whose average complexity is indicated by the number of constituent line segments listed in the table. Here, the last column lists the percentage of unreliable groups in the top 100 salient groups found using the saliency measure. From these studies we conclude that the saliency measure captures reliable groups and can, therefore, be useful in a data-driven selection mechanism for recognition.

In the discussion so far, we have not analyzed the extent to which the groups selected by the saliency measure match our perceptual judgment of the importance of such groups. We chose not to emphasize this aspect for several reasons. First, in an edge image, other relations in addition to closely-spaced parallelism, may exist between lines. For example, long smooth curves may be more salient than parallel-line groups in an edge image. For comparing the performance of the saliency measure, the subjects should be made to look at only closely-spaced line groups and ignore other cues for grouping, a task difficult to achieve in practice. Even if scenes showing only instances of closely-spaced parallelism were examined, there is the additional problem due to the grouping algorithm generating only a subset of the possible groups. Thus not all the groups perceived by a subject may be generated and this affects the groups selected by the saliency measure. Finally, perceptual judgments may be based on a collection of groups of different orientation (this could indicate groups of curves, for example), and this is not considered by the saliency measure.

4.2 Use of salient line groups-based selection in recognition

Data driven selection based on salient line groups is primarily useful when the object of interest has at least one parallel-line group that appears among the selected salient line groups. In such cases, the search for data features that match model features can be restricted to salient groups thus avoiding needless search in other areas of the image. In order for a model group to be found among the salient line groups, however, it should first be generated by the grouping algorithm. That is, the choice of the four constraints characterizing an image

S.No.	Image Size	Num. Lines	Group Constraints	Num. Groups	Avg. Salient Group Size	Avg. Salient Group Area	% Unreliable Groups
1.	320 x 576	395	$< 10,5,6,10 >$	219	6	0.009	1
2.	256 x 416	756	$< 6,5,6,10 >$	382	8	0.006	2
3.	224 x 416	316	$< 5,5,6,10 >$	207	6	0.008	7
4.	200 x 492	1232	$< 5,10,9,12 >$	552	4.25	0.003	3
4.	232 x 576	884	$< 5,5,6,10 >$	454	5.3	0.004	6

Table 2: *Characteristics of salient closely-spaced line groups ranked by the saliency measure described in text. In each case, the top 100 salient groups are considered. The number of groups listed here include single line groups.*

group should be such that a model group, if it exists in the image, can be captured by these constraints. Since no specific knowledge of model objects can be used in data-driven selection, the thresholds can be set based on some rough a priori knowledge about expected objects in scenes, the distances at which they are imaged, and some general knowledge about the parameters of the 3D structures that are meant to be captured in line groups in the images of such scenes. Among the thresholds, the local and global orientation thresholds, $t_{local-orient}$ and $t_{global-orient}$, are essentially independent of the objects in the library, and can be chosen based on an analysis of the imaging noise and the noise in line-segment approximation. Since a group captures almost parallel lines, there is not much leeway in their choice, in that they can be only low values. We chose to model this noise by allowing 5-7 degree skew in between lines ($t_{local-orient}$) and an overall skew ($t_{global-orient}$) of 10 degrees. The values of t_{across} and t_{along} however, have a major effect on the lines that are ultimately grouped, and cannot in general, be chosen independent of objects in the library. Larger values of these thresholds allow somewhat widely-spaced parallelism to be captured, such as the parallelism inherent in the contour of the triangular block in Figure 2b. However, this would also decrease the reliability of the groups, as it allows lines belonging to separate objects to be grouped. Since our aim (at least in data-driven selection) is to capture letter and wooden texture occurring on objects in indoor scenes, we found that for the distances at which the objects are typically imaged, an interline separation (t_{across}, t_{along}) of 5 to 10 pixels is sufficient for capturing most such parallel-line groups in images. Better methods for choosing these thresholds, could be devised, however.

4.2.1 Search reduction using salient line groups

We now estimate the search reduction that can be achieved by using salient-line groups-based data-driven selection. Following the analysis of the number of matches using grouping given in Section 2, if only S salient groups are retained for M_i image groups, then the number of matches to be tried using the 4-point alignment scheme of recognition is $O(\sum_{i=1}^{M_i}\sum_{j=1}^{S} m_i^4 n_j^4 4!)$ where m_i and n_j are the number of features in the model and the selected salient image groups, respectively. To estimate the number of matches using such salient line groups, we chose a few model objects exhibiting closely-spaced parallelism between lines on the object, generated line groups using the grouping algorithm and retained a few of the groups. By placing these objects in various scenes, and using the grouping algorithm and the saliency measure, we retained a few (about 40) salient line groups in the images of these scenes. The number of features (i.e. the end points of line segments) in both model and data groups were recorded. The number of matches with salient groups-based selection was found using the above formula. For purposes of comparison, the number of matches without any grouping was also computed using the formula $O(M^4N^4)$ where M and N are the total number of features found in all the model and data line groups. The results are summarized in Table 3. As can be seen from the table, the search is always considerably lower when salient groups-based selection is done prior to recognition. The number of matches with this type of selection scheme can still be large, however, as all pairings of model and salient image groups are tried. Also, some of the salient groups are large in size thus increasing the number of matches that need to be tried within a pair of groups.

5 Model-driven Selection using Line Groups

So far we have considered the use of closely-spaced parallel line groups for data-driven selection which required the object of interest to have a salient line group. Further, it was assumed that such a group could be detected by the use of some default threshold values for the constraints characterizing closely-spaced line groups. This will not be of much help when the object of interest has closely-spaced parallel line groups but they are either not salient or cannot be captured using the default thresholds. In such cases, the description of the line groups present on the model object can be used to perform selection. We now describe one such line groups-based model-driven selection mechanism. The approach adopted here is to selectively generate line groups in the image based on the description of closely-spaced parallel line groups on the model object. Thus the group generation process is constrained here so that only the likely matching groups are generated. Once the candidate matching line groups in the image are obtained, recognition can proceed by examining pairs of model and image groups as usual.

The criteria developed for a model-driven selection mechanism in the earlier work [28] are also relevant to line groups-based model-driven selection. That is, it must be sufficiently selective to avoid considering ob-

S.No	M	N	M_g	Avg. m_i	Group Constraints	Avg. n_j	Num. Matches	
							No Selection	Data-driven Selection
1.	466	1768	9	6	$< 5,5,6,10 >$	12	$1.88\text{x}10^{22}$	$5.1\text{x}10^{11}$
2.	140	1768	10	4	$< 10,5,6,10 >$	12	$1.49\text{x}10^{20}$	$1.0\text{x}10^{11}$
3.	140	2464	10	4	$< 10,5,6,10 >$	10	$5.6\text{x}10^{20}$	$4.66\text{x}10^{10}$
4.	358	1453	8	16	$< 2,2,6,10 >$	18	$2.98\text{x}10^{21}$	$1.38\text{x}10^{14}$
5.	130	790	5	23	$< 10,5,6,10 >$	12	$4.38\text{x}10^{18}$	$4.46\text{x}10^{13}$

Table 3: *Estimated search reduction using salient line groups-based selection. The terms M, N, M_g, m_i, n_j are as explained in text. Here $N_g = 40$, i.e., the top 40 salient groups are retained for selection.*

viously impossible matches, but at the same time, be sufficiently flexible to take into account the various problems in imaging that may cause a model line group in an image to appear different from its original description. The object may appear different in a scene because it has undergone pose changes, or because it is occluded, or because illumination changes as well as artifacts in the scene such as specularities, interreflections have altered the appearance of the object. These changes (called observation conditions, henceforth) can also alter the appearance of the line groups on the model object. We first examine, therefore, the effects of these changes on the model line groups. This will then be used to design a description of model line groups as well as a strategy for generating matching line groups in the image.

5.1 Effect of observation conditions on a model line group

Consider a closely-spaced parallel (3D) line group on a model object. It can be specified by a tuple $< t_{3d-across}, t_{3d-along} >$ with the following interpretation: Between any two consecutive parallel lines in the model 3D group, the across-the-line distance $d_{3d-across}$ is no more than the threshold $t_{3d-across}$ and the along-the-line distance $d_{3d-along}$ is no more than the threshold $t_{3d-along}$. The distances $d_{3d-across}$ and $d_{3d-along}$ for parallel lines in 3D are defined in a way analogous to that of d_{across} and d_{along} given in Section 3.1.1 [3]. We now analyze the effect of observation conditions on the appearance of such 3D line groups on model objects when they placed in a scene.

Pose changes: If the allowed transformation of the model object is restricted to a 3D affine transformation, then under orthographic projection and scale (to approximate perspective projection) and assuming no imaging noise and no errors in line segment approximation, it is known that closely-spaced parallel 3D line groups project to a set of closely-spaced parallel lines in the image [30][4]. Further, the order of the lines in the group is preserved, although the resulting orientation of the parallel lines in the image can be arbitrary. The inter-line spacing d_{across} and d_{along} between the projected lines will, however, be different from the inter-line spacing $d_{3d-across}$ and $d_{3d-along}$ between the corresponding 3D lines. If no scale change has occurred during the transformation, then the spacing between the projected parallel lines in the image can only decrease. This is because d_{across} forms a side of a right triangle whose hypotenuse is the orthographic projection of the inter-line spacing $d_{3d-across}$ (as shown in Figure 7), and will always be less than or equal to $d_{3d-across}$ [5]. Using a similar argument, we can show that the along-the-line spacing in the image d_{along} is less than or equal to $d_{3d-along}$. If the transformation includes scale changes, then the inter-line spacing between parallel lines varies proportional to the scale. That is, for a scale change s (s > 1.0 or < 1.0) we have $d_{across} \leq s * d_{3d-across}$ and similarly $d_{along} \leq s * d_{3d-across}$. Thus the effect of pose changes is to vary the inter-line spacing between lines while still maintaining the property of parallelism.

Occlusions: The most common effect of occlusions is to corrupt the projected model line group. That is, depending on the geometry of the occlusion, some lines in the group may either partially or totally disappear. But unless a group is completely occluded, the visible lines of the group maintain the same relative ordering and the inter-line spacing is dictated by the pose changes undergone by the model. In rare cases, when the occluding object has similar closely-spaced parallel line groups, it may cause the two groups to be merged, and may even affect the inter-line spacing of the model line group. Thus the effect of occlusions on a model line group is mainly to change the number of constituent lines in the model groups, and in rare cases to even alter the inter-line spacing in the groups.

Illumination changes and other imaging artifacts: When the wavelength characteristics of the light source illuminating the scene is different from the one illuminating the model object, it may cause a change in the apparent color of the object's surface. But since we are looking at an edge image to generate the groups, the edges tend to remain more or less stable. When the

[3] The distance $d_{3d-across}$ is simply the distance between two consecutive parallel lines defined as the length of projection of the end point of one 3D line on the other. The along-the-line distance $d_{3d-along}$ is as defined for 2d-lines except that the distance d_{min}, $L(p_r)$, and $L(p_s)$ are distances between points in 3D.

[4] If the transformation takes some of the object out of view, then this can be treated as the case when some projected parallel lines coincide.

[5] This is also true when perspective projection is approximated by orthographic projection and scale.

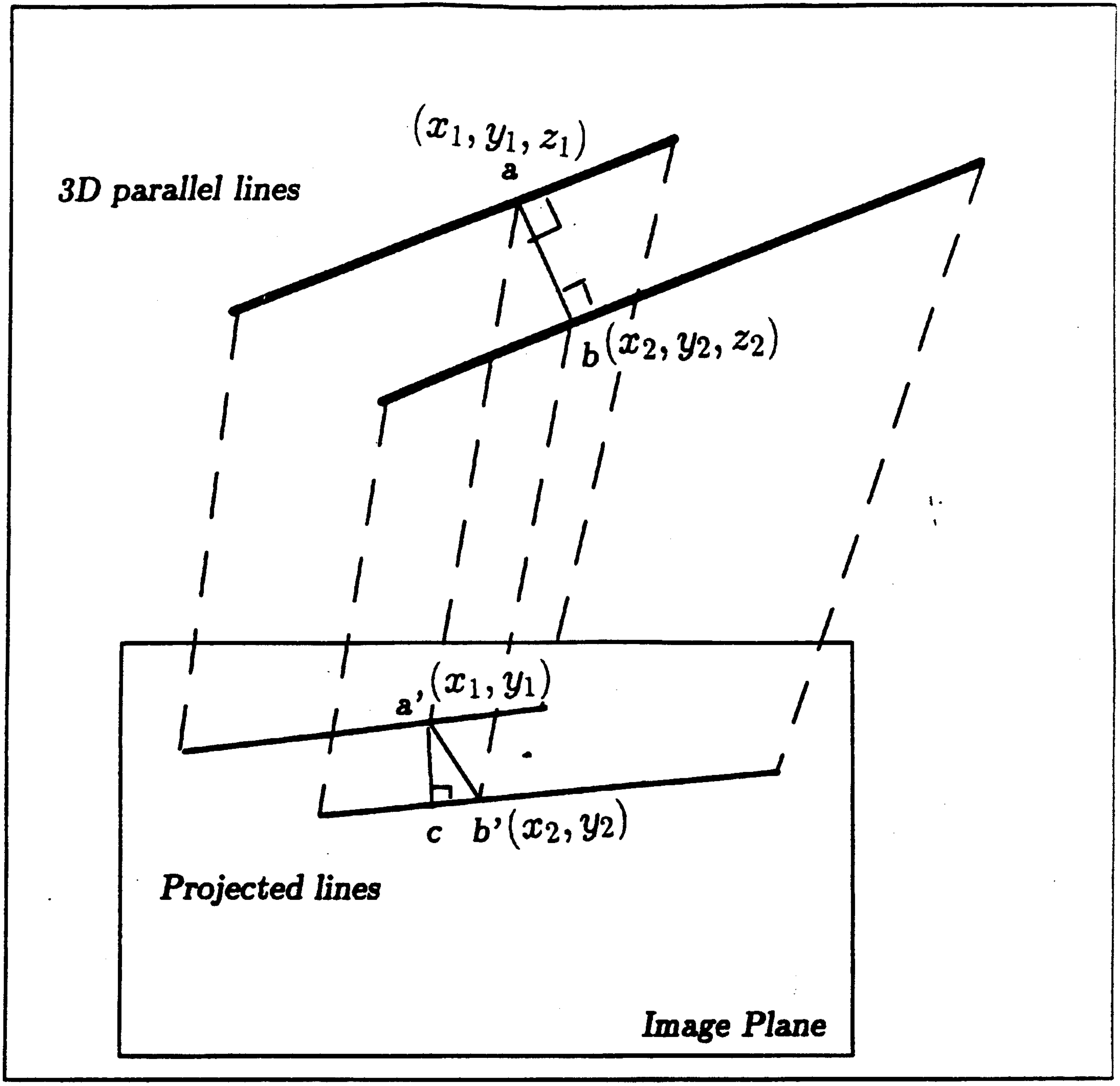


Figure 7: *Illustration to show that the distance between the projections of 3D parallel lines is less than or equal to the 3D distance between the lines. The projection of the 3d distance $d_{3d-across}$ (line ab) is given by the line $a'b'$. The length of $a'b'$ is $\leq$ length of ab since $ab = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$ and $a'b' = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. The distance between the projected lines given by $a'c$, is $\leq a'b'$ by the hypotenuse of a right triangle rule.*

light source location is changed, however, then depending on the position of the 3D parallel line groups relative to the light source, lines in the projected group may be either partially or totally hidden in shadow. Further, if the surface of the object is specular, then specularities occurring in the region of the line group may cause the group to be corrupted by the partial or total masking of the lines in the group. Finally, interreflections can cause spurious line segments to appear as part of the model line group depending on the pattern on the object that is being reflected from the model's surface. Thus the effect of illumination changes is similar to occlusions in that both the number of constituent lines and the inter-line spacing in the model groups may be changed.

5.2 Model line group description

We now develop a description of the model line groups taking into account the effect of the observation conditions. From the above analysis, we know that a closely-spaced strictly parallel line group on the model specified by $< t_{3d-across}, t_{3d-along} >$ also occurs as a group of parallel lines in an image with an inter-line spacing that is decided by the scale change involved in the undergone transformation. By restricting the scale changes to lie between $< s_1, s_2 >$ (with $s_1 <= 1.0$ and $s_2 > 1.0$), the variation in the inter-line spacing d_{across} that can be handled for any model line group appearing in any image can be restricted to lie in the range $[s_1 * t_{3d-across}, s_2 * t_{3d-across}]$ and similarly, the spacing d_{along} to lie between $[s_1 * t_{3d-along}, s_2 * t_{3d-along}]$. So far, the effect of imaging noise and errors in line segment approximation were not taken into account. As we remarked earlier, their effect is to cause the lines in the model group to be slightly skewed in the image. The thresholds $t_{m-local-orient}$ and $t_{m-global-orient}$ can be used to specify the tolerable inter-line skew and the overall skew in the line group. The resulting description of a model line group as it appears in any image can be given by the tuple $< t_{3d-across}, t_{3d-along}, t_{m-local-orient}, t_{m-global-orient}, s_1, s_2 >$. Different line groups on the model will differ in the first two terms (as the thresholds $t_{local-orient}$ and $t_{global-orient}$ are independent of the line group), and the scale changes are specified with respect to the model object.

To generate candidate line groups in a given image using the above model description, however, the inter-line spacing and the tolerable scale changes must be expressed in terms of pixel spacing. For this, the distance at which the model is imaged for building the general model description can be used as the reference distance. That is, the pixel spacing corresponding to $t_{3d-across}$ and $t_{3d-along}$ can be taken to be the one existing between the projected lines in the image when the model is placed at the reference distance and oriented in such a way that the model line group is parallel to the image plane. By moving the object closer or farther than the reference distance by the scale factors s_1 and s_2 respectively, the corresponding change in the pixel spacing can be recorded. If $t_{m-2d-across}$ and $t_{m-2d-along}$ represent the pixel spacing corresponding to $t_{3d-across}$ and $t_{3d-along}$ respectively, and p_1 and p_2 represent the change in pixel spacing corresponding to the scale changes s_1 and s_2, then the description of the model line group that can be used to generate the line groups in the image becomes $< t_{m-2d-across}, t_{m-2d-along}, t_{m-local-orient}, t_{m-global-orient}, p_1, p_2 >$

5.3 Selective generation of matching line groups

We now present an algorithm for selectively generating line groups in the image that match a given model line group description. Since the model description places a bound on the tolerable scale and pose changes, the basic strategy is to generate line groups with successively increasing inter-line (both d_{across} and d_{along}) spacing until the upper bound specified in the model description is reached. That is, given a model line group description $< t_{m-2d-across}, t_{m-2d-along}, t_{m-local-orient}, t_{m-global-orient}, p_1, p_2 >$ the maximum across-the-line spacing allowed between the lines varies from $t_{m-2d-across} - p_1$ to $t_{m-2d-across} + p_2$ (and similarly for the along-the-line spacing). The matching groups are generated by hypothesizing a value of spacing between the lines lying in the above range and generating all groups with inter-line separation specified by that value. In particular, successive integer pixel spacing from $t_{m-2d-across} - p_1$ to $t_{m-2d-across} + p_2$ are used to generate the groups. Such groups, called augmented closely-spaced parallel line groups, can be specified by the tuple $< t_{across-low}, t_{across}, t_{along-low}, t_{along}, t_{local-orient}, t_{global-orient} >$ with the following interpretation: It is the largest group of non-intersecting line segments such that for each line in the group, there exists no line in the group such that $d_{across} < t_{across-low}$ and $d_{along} < t_{along-low}$ and there exists at least one line in the group obeying the following constraints:

1. The across-the-line distance d_{across} between the lines is such that $t_{across-low} < d_{across} \leq t_{across}$.
2. The along-the-line distance d_{along} between the lines is such that $t_{along-low} < d_{along} \leq t_{along}$.
3. The orientation difference between the lines is no more than a threshold $t_{local-orient}$.

and the entire group satisfies the assumption that the maximum orientation change between any two lines in the group is no more than a threshold $t_{global-orient}$.

If the observation conditions include only pose changes, and if all the lines in a model group are equi-spaced, then the model group appearing in the image (the visible part of it, that is) is bound to be present in one such augmented groups because its consecutive lines would exhibit an inter-line spacing within the specified range. In the presence of occlusions and other observation conditions, and when the inter-line spacing in the model group is not uniform, the model group can still be captured in the augmented groups, albeit in a fragmented form. That is, the model group may be partitioned (and possibly merged with adjacent lines) into several augmented groups. But as long as two adjacent lines in the model group are visible in the image, they can still be captured in one of the augmented groups and this should, in theory, be sufficient for recognition (as the two line end points can provide four features).

5.3.1 Algorithm for generating augmented line groups

The algorithm for selectively generating the groups satisfying a model description $< t_{m-2d-across}, t_{m-2d-along}, t_{m-local-orient}, t_{m-global-orient}, p_1, p_2 >$ proceeds by first generating the line group $< t_{m-2d-across} - p_1, t_{m-2d-along} - p_1, t_{m-local-orient}, t_{m-global-orient} >$ using the grouping algorithm of Section 3.2. This can capture model groups that have unequal inter-line separation in the case of the model object undergoing pose change specified by the lower limit of $t_{m-2d-across} - p_1$. Then augmented closely-spaced line groups specified by $< t^{i-1}_{across}, t^{i}_{across}, t^{i-1}_{along}, t^{i}_{along}, t_{local-orient}, t_{global-orient} >$ are successively generated using a modified version of the grouping algorithm of Section 3.2 as follows:

1. Let $t^0_{across} = t_{m-2d-across} - p_1$ and $t^0_{along} = t_{m-2d-along} - p_1$.
2. For i = 1 to p2 - p1 do
    - Let $t^i_{across} = 1 + t^{i-1}_{across}$. and $t^i_{along} = 1 + t^{i-1}_{along}$.
    - The grouping algorithm of Section 3.2 is applied to the line segments with the following modifications to Steps 2a and 2c:
        - In Step 2a, an annulus of neighborhood lying between the rectangles $2t^{i-1}_{across}$ x $2(t^{i-1}_{along} + l)$ and $2t^i_{across}$ x $2(t^i_{along} + l)$, where l is the length of the line, is scanned and all lines passing through this annulus but not through the inner rectangle are retained.
        - In Step 2c, in addition to checking for $t_{m-global-orient}$, each enclosing group being merged with the current group is checked for violations against the annulus neighborhood constraint mentioned above.

5.3.2 Analysis

The above algorithm makes $(p_2 - p_1)$ passes over the line segments in generating the matching groups for each model line group. However, the total number of matching groups generated is still linear in the number of line segments since each line can belong to at most $(p_2 - p_1)$ groups, one for each line lying in the annulus of neighborhood between the outer and inner rectangles of dimensions $2t^{i-1}_{across}$ x $2(t^{i-1}_{along} + l)$ and $2t^i_{across}$ x $2(t^i_{along} + l)$. The above procedure can be repeated for generating matching groups for each model group separately. Alternatively, the allowable inter-line spacings for all model line groups can be pooled together to form ranges of pixel spacing for all the model groups, and the search for matching groups can be done for each such range using the above algorithm. The time to generate the augmented line groups for an iteration i is still O(n^2) (n is the number of line segments) since the grouping algorithm is the same as before, and Step 2c examines each pair of line segments at most once. For the allowed scale changes, the range $(p_2 - p_1)$ is small enough so that the entire operation of selective group generation can be done in $O(kn^2)$ time, where $k \ll n$ is a constant representing the number of passes over the line segments.

An Example

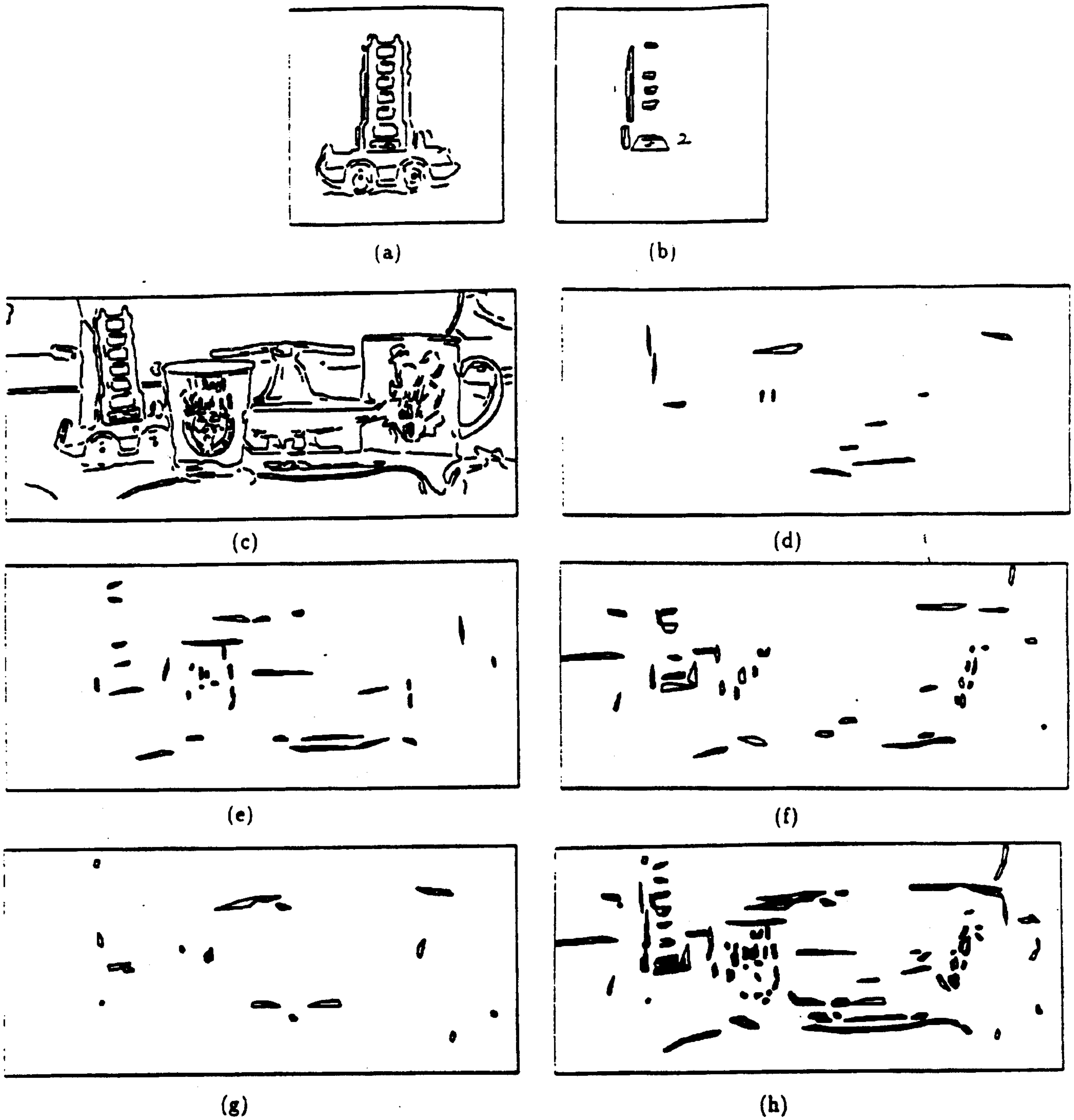
We now illustrate model-driven selection using parallel-line groups with an example. Figure 8 shows model line groups being used to perform selection. Here the parallel-lines on the ladder part of the toy fire truck serve as the model line groups and are specified by the constraints $< 5, 0, 6, 10, 3, 0 >$ implying that the allowable scale changes are from a maximum across-the-line spacing of (5-3 =) 2 pixels up to 5 pixels (in other words, allowing the object to be imaged farther than it is in the model description). Here no variation is allowed in the along-the-line spacing as the model lines all overlap (i.e. have $t_{m-2d-along} = 0$). The model-line groups with the given specification are shown in Figure 8b. Figure 8c shows (an edge-image of) a scene in which the model object appears at a different orientation and has a portion of it occluded. Figures 8d-g show the matching augmented closely-spaced line groups obtained using successively increasing line-spacing as specified by the range $< 0, 2, 0, 0, 6, 10 >$, $< 2, 3, 0, 0, 6, 10 >$, $< 3, 4, 0, 0, 6, 10 >$, and $< 4, 5, 0, 0, 6, 10 >$, respectively. These matches to the indicated model line groups are shown collectively in Figure 8h. As can be seen from the figure most of the model line groups are captured in the matching groups, although there is evidence of fragmentation in two of the groups marked 1 and 2 as shown in Figure 8b.

5.3.3 Discussion

Model-driven selection using the above algorithm generates enough candidate match groups for a model line group to avoid false negatives under most observation conditions. Further, by requiring the groups to have a minimum inter-line spacing, it avoids generating unnecessary false positive matches to model line groups. This can easily happen if a simpler strategy for group generation were used such as generating ordinary (rather than augmented) closely-spaced parallel line groups by successively increasing the line spacing from $t_{2d-across} - p_1$ to $t_{2d-across} + p_2$. Such a scheme would create successively bigger groups (since a group with small line spacing would always satisfy the constraint of a bigger line spacing) that are often more unreliable and unlikely matches to model groups.

5.4 Search reduction using model-driven line grouping

The model-driven selection mechanism using line groups described above identifies candidate groups in the image that could be potential matches for model line groups under some allowable transformation and taking into account the effect of occlusions, illumination changes, etc. These matching model and image line groups can then



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 8: *Illustration of line-groups-based model-driven selection. (a) Edge image of a model object showing instances of closely-spaced parallelism between lines. (b) Some of the line groups extracted using the grouping algorithm using the constraints of* $t_{across} = 5, t_{along} = 0, t_{local-orient} = 6, t_{global-orient} = 10$. *(c) An edge image of a scene in which the model object appears. (d) - (g) Augmented closely-spaced line groups generated using the description of the model line groups shown in (b). (h) The line groups in the image that are possible matches to the model line groups of (b) under the allowed scale and pose changes.*

be given to a recognition system that will isolate features from the line groups to actually solve for the pose of the model object. To see the search reduction possible with model-driven selection, let M_g model line groups be used to perform model-driven grouping. Let the total number of matching groups be N_g with k_i image groups matching a model group i. Letting (i,j) represent a match between model group i and image group j, and letting (m_i, n_j) stand for the number of features in the matching groups, the number of matches that may have to be tried to align the model object with the image is $O(\sum_{i=1}^{M_g} \sum_{j=1}^{k_i} m_i^4 n_j^4 4!)$. To estimate the search reduction due to model-driven grouping, we selected some model objects (views of them, that is) possessing closely-spaced parallel lines on the surface, generated and retained some line groups, and recorded their specifications as $\{t_{2d-across}, t_{2d-along}, t_{local-orient}, t_{global-orient}\}$. We then used scale bounds of $s_1 = 0.5$ and $s_2 = 2.0$ (this allows objects in the scenes to be imaged at half to twice the distance at which their model descriptions were recorded) to complete the model line group descriptions. By placing these objects in scenes containing clutter, and allowing partial occlusions and illumination changes, we ran the selective group generation algorithm of Section 5.3.1 to record the matching image line groups for model line groups. The end points of line segments were considered as features to be used for recognition, and the number of features in both the model and image line groups were recorded. These experiments gave the values for M_g, N_g, m_i, n_j, k_i in the above formula. Table 4 shows the results of these studies, with column 10 showing the number of matches using model-driven grouping evaluated using the above formula. The number of matches that would be required without grouping is also shown in the table for comparison. As can be seen, the number of matches is far less with model-driven grouping. To get an estimate of the actual search reduction as well as the number of false positives and negatives due to model-driven grouping, however, the grouping mechanism should be integrated with an actual recognition system and its performance evaluated. The results of such experiments will be discussed in a later section.

Even with model-driven grouping, the number of matches when considered on an absolute basis, is still very large. This can again be attributed to the large number of matches k_i for model line groups, and to the sometimes large size of the matching groups. To handle scale changes, large inter-line spacing values have to be examined for group generation, unlike in the case of data-driven selection. This may cause some of the groups to be unreliable or large-sized because of merging across objects. Both these problems can be alleviated if model-driven grouping is used in conjunction with prior region selection done using more reliable cues such as color and texture. One such method of combining grouping with prior selection is discussed in the next section.

6 Line Grouping in Conjunction with Prior Region Selection

So far we have examined grouping of line segments based on the constraint of closely-spaced parallelism as an independent selection mechanism. But our original motivation for grouping lines was to organize the features within prior selected color or texture regions into small groups, primarily for reducing the search in recognition. We now explore the use of line grouping within regions that are selected a priori based on cues such as color and texture.

Data or model-driven selection using line groups can be easily achieved within previously selected regions by modifying the grouping algorithms of Sections 3.2 and 5.3.1 to assemble lines obeying an additional constraint of all lying within the selected regions. This not only restricts the number of groups generated in the image but also their size, by preventing the lines belonging to adjacent objects from being merged, thus making such groups more reliable. Moreover, when model-driven grouping is done within prior selected regions, an additional constraint is provided by the enclosing regions and restricts the possible matches to model line groups even further. For example, when the regions are prior selected based on model color regions, then as shown in [27], a correspondence between model and selected image color regions is also established. The search for image line groups matching a model line group, therefore, can be restricted to color regions in the image that correspond to the model color regions spanned by the model line group.

To estimate the search reduction using line grouping in conjunction with prior region selection, we performed experiments in which model-driven line groups were generated within model color regions. The information about color regions spanned by a model line group was used as an additional constraint in finding matching image line groups. An example of such restricted model-driven selection using line groups is indicated in Figure 9 and Figure 10. Figure 9 shows the result of color-based selection as described in [27]. Figure 9a and b show two views of a model object used to construct its 3-dimensional description. Figure 9c shows the region adjacecny graph describing the color regions in the model object using the color region segmentation algorithm described in [28]. The result of color-based region selection using the model object description of Figure 9c in the scene of Figure 9d is shown in Figure 9e. Next, Figure 10 shows the result of line-groups-based selection within the prior selected regions in the scene of Figure 9d. The model group specification is the same as in Figure 8, namely, $< 5, 0, 6, 10, 3, 0 >$. Figure 10d shows the regions isolated in the image using color-based model-driven selection. Figure 10e-h show the matching line groups generated using the augmented closely-spaced grouping algorithm within the selected regions of Figure 10d. Finally, Figure 10i shows all the matching groups using the allowable transformations for the line groups shown in Figure 10b. By performing similar experiments on a number of model object and scenes.

S.No.	M	N	M_g	Avg. k_i	Avg. m_i	Avg. n_j	Group constraints	Num. Matches	
								No Selection	Model-driven Selection
1.	466	1768	9	150	6	4	$< 5,0,6,10,3,0 >$	$1.88\text{x}10^{22}$	$5.9\text{x}10^{10}$
2.	140	1768	10	111	4	4	$< 7,0,6,10,5,2 >$	$1.49\text{x}10^{20}$	$6.38\text{x}10^{9}$
3.	140	2464	10	94	4	6	$< 7,0,6,10,5,1 >$	$5.6\text{x}10^{20}$	$2.86\text{x}10^{10}$
4.	358	1453	8	48	16	8	$< 3,1,6,10,1,2 >$	$2.98\text{x}10^{21}$	$6.65\text{x}10^{12}$
5.	130	790	5	25	23	6	$< 6,0,6,10,2,1 >$	$4.39\text{x}10^{18}$	$9.00\text{x}10^{11}$

Table 4: *Estimated search reduction using line groups-based model-driven selection. The terms M,N, M_g, N_g, k_i, m_i, n_j are as explained in text. The allowed scale and pose changes in each case are indicated in the augmented group constraints.*

we recorded the resulting values of $M_g, N_g, m_i, n_j, k_i, n_j$ (these terms were defined in Section 5.4) and these are shown in Table 5. The number of matches using model-driven grouping with prior region selection was calculated using the same formula that was given in Section 5.4 and is shown in Column 10 in Table 5. This can be compared with the number of matches using model-driven line grouping without prior region selection given in Table 4. As can be seen from the tables, combining line grouping with prior region selection based on cues such as color can greatly reduce the estimated search involved in recognition. This is also corroborated, as we will see next, by experiments done with an actual recognition system.

Restricting line grouping within prior selected regions has the disadvantage though that it relies on the correctness of the prior selection mechanism. This is not always the case. Color-based selection for example, does identify a good portion of the regions containing the object. But the region isolation is not often very precise so that some spurious line segment-containing groups may still be formed in such a grouping process.

7 Actual Search Reduction in a Recognition System due to Line Groups-based Selection

Although the search is greatly reduced by performing grouping within prior selected color regions, the estimated numbers are still large ($\approx 10^6$). A recognition system that actually does this amount of search is far from practical. These numbers were arrived at using a worst-case scenario in which only two pairs of matching groups could be found at the end after searching through the entire set of possible matching pairs. In practice, we expect to find lot of good matching pairs much sooner in the search. To test the actual search reduction possible in practice, we built a recognition system and integrated the line grouping-based selection mechanism to record the improvement in performance. The linear combination of views-based alignment was used for a testbed recognition system [31]. The 3D models were constructed from two 2D views with full correspondence between them obtained using a method described in [23]. Corner features extracted from both the model and image were used to perform the alignment and line segment features were used for doing the verification. The search for corresponding alignment features was done using an interpretation tree type search driven from the image features [9]. We then used color-based selection to isolate areas in the image that are likely to belong to the object using the method described in [27]. Then line grouping was performed within the selected color regions to obtain line groups that match model descriptions. Two pairs of matching line groups were searched and features within the matching pairs were tried for finding the alignment transform. Sometimes three pairs of matching line groups had to be tried to obtain sufficient features for good alignment. Figure 11 shows an example of recognition being performed with selection based on color and line grouping. The model line groups and the matching image line groups are as shown in Figure 10. A set of matching line groups and the corresponding corner features within them that yield a transformation that is verified by the recognition system to be correct are shown in Figure 11e and f. The projected model overlayed on the original image is shown in Figure 11g. By considering several (around 600) random orderings of the list of groups and features within groups in a number of different scenes, we recorded the average number of matches that needed to be tried before successful verification. Some of these results are shown in Table 6. The models and scenes are the same as those used in Tables 4 and 5, but the features here are corners instead of the end points of line segments. The number of matches actually explored by the recognition system for finding seven corresponding corner features using line grouping-based selection within color-selected regions are indicated in Column 10 of Table 6. The rather larger number of matches for a smaller model object in entry 5 of the table is due to the larger size of groups (the maximum size was 23) even though the number of model groups is small. Compared to the number of matches explored, detailed verification was done for only a few (about a 1000) of the matches. The estimated number of matches that would be explored without selection for seven corresponding features is shown in Column 9 for comparison. From these results, we concluded that line grouping-based selection when performed within prior selected regions leads to a tremendous improvement in the performance of a recognition system.

S.No	M	N	M_g	Avg. k_i	Avg. m_i	Avg. n_j	Group constraints	Num. Matches	
								No Selection	With Prior Region Selection
1.	466	1768	9	26	6	4	$< 5, 0, 6, 10, 3, 0 >$	1.88x10^{22}	1.77x10^{9}
2.	140	1768	10	20	4	4	$< 7, 0, 6, 10, 5, 2 >$	1.49x10^{20}	2.07x10^{8}
3.	140	2464	10	19	4	4	$< 7, 0, 6, 10, 5, 1 >$	5.6x10^{20}	1.87x10^{8}
4.	358	1453	8	17	16	6	$< 3, 1, 6, 10, 1, 2 >$	2.98x10^{21}	2.39x10^{11}
5.	130	790	5	13	23	4	$< 6, 0, 6, 10, 2, 1 >$	4.39x10^{18}	3.89x10^{10}

Table 5: *Estimated search reduction using restricted line groups-based model-driven selection within prior selected color regions.*

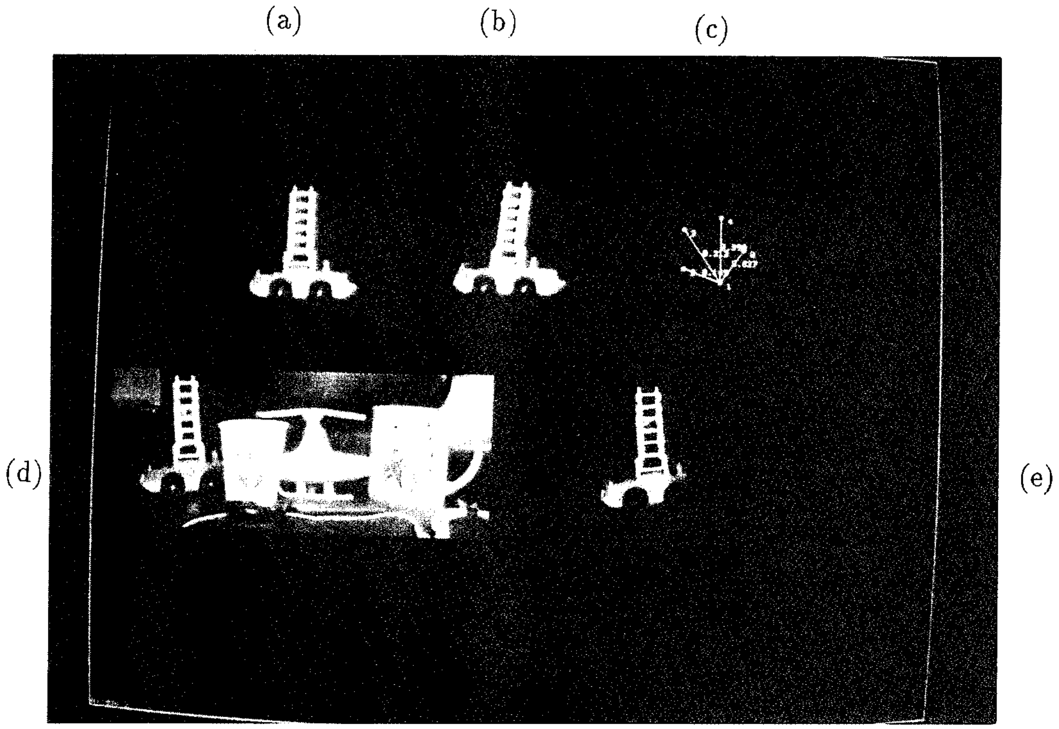


Figure 9: *Illustration of color-based model-driven selection. (a)- (b) Two views of a model object used to construct a three-dimensional description. (c) A region adjacency graph description of the color regions on the model object. (d) A scene in which the object appears. (e) The result of color-based selection using the model description of (c).*

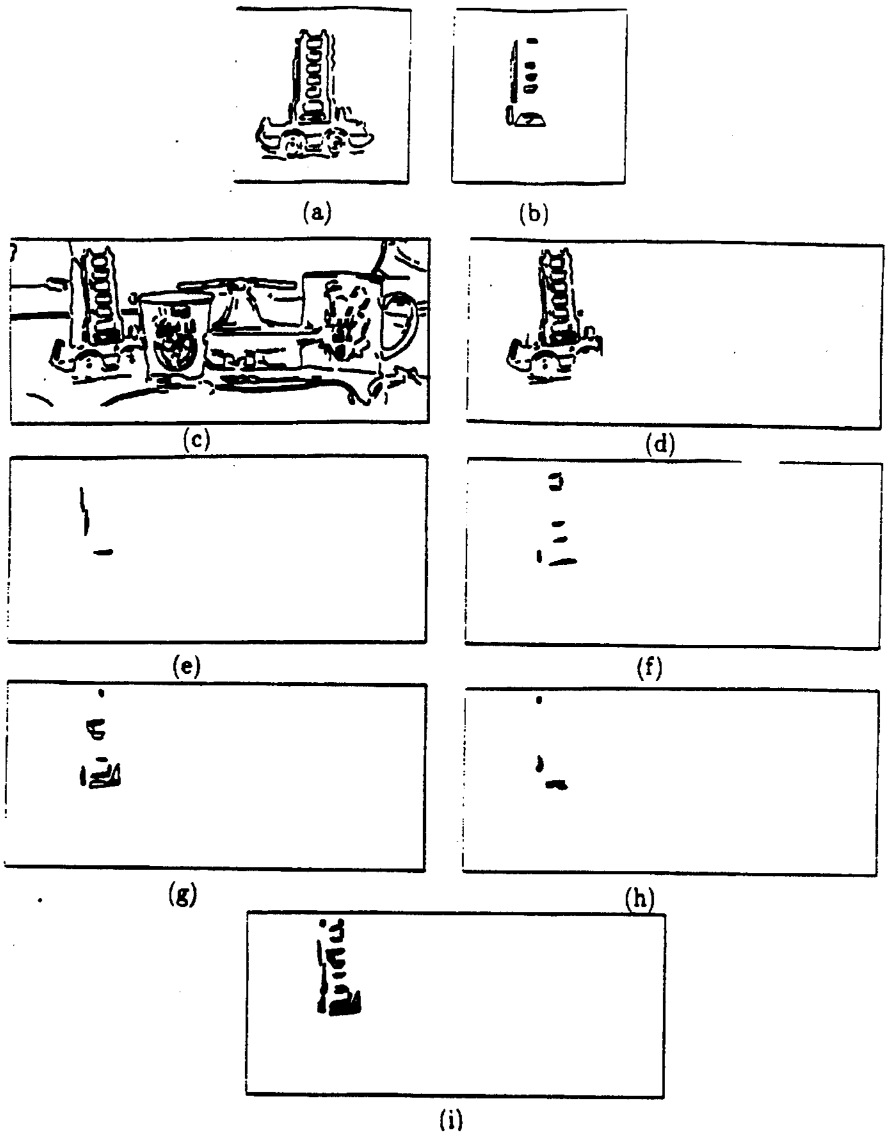


Figure 10: *Illustration of model-driven selection using line group within prior selected color regions. (a) Edge image of a model object showing instances of closely-spaced parallelism between lines. (b) Some of the line groups extracted using the grouping algorithm using the constraints of $t_{across} = 5, t_{along} = 0, t_{local-orient} = 6, t_{global-orient} = 10$. (c) An edge image of a scene in which the model object appears. (d) The region isolated using color-based model-driven selection. (e)- (h) Matching line groups using successively increasing inter-line spacing as described in text. (i) The line groups in the image that are possible matches to the model line groups of (b) under the allowed scale and pose changes.*

S.No	Model corners	Image corners	M_g	Avg. k_i per scale	Avg. m_i	Avg. n_j	Group constraints	Num. Matches	
								No Selection	Explored in recognition
1.	160	580	7	8	4	3	$< 5, 0, 6, 10, 3, 0 >$	9.93x10^{30}	3.9x10^{4}
2.	54	580	10	7	3	3	$< 7, 0, 6, 10, 5, 2 >$	3.77x10^{27}	8.4x10^{4}
3.	54	391	10	9	3	3	$< 7, 0, 6, 10, 5, 1 >$	2.34x10^{26}	1.5x10^{4}
4.	114	484	8	11	4	4	$< 3, 1, 6, 10, 1, 2 >$	2.45x10^{29}	9.2x10^{4}
5.	30	95	5	13	6	3	$< 6, 0, 6, 10, 2, 1 >$	1.13x10^{20}	1.35x10^{5}

Table 6: *Actual search reduction using restricted line groups-based model-driven selection within prior selected color regions. Here m_i and n_j refer to number of corner features (rather than the end points of line segments) within a line group. Seven corresponding corner features were used for recognition.*

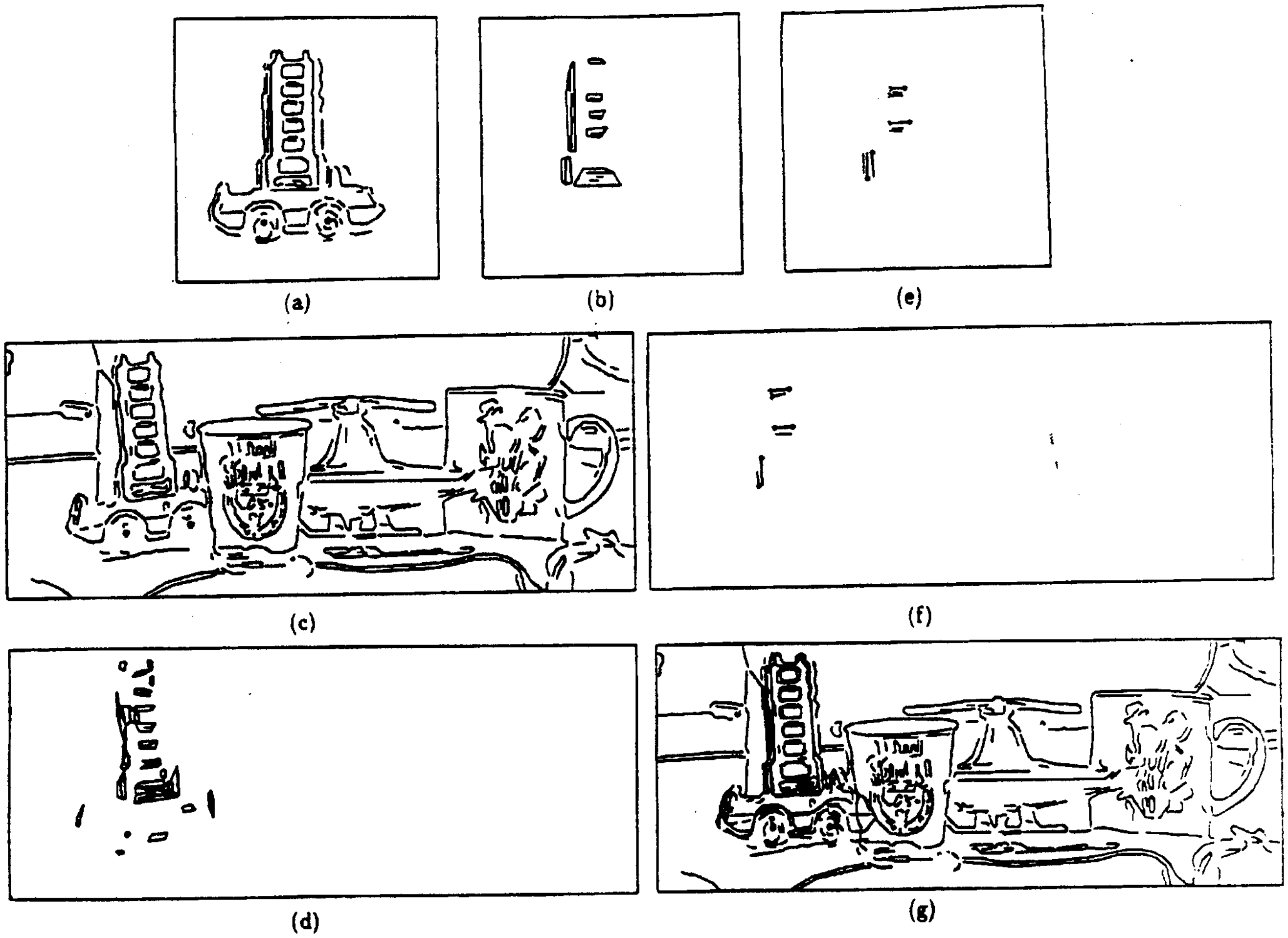


Figure 11: *Illustration of recognition in the regions selected by the attentional selection mechanism using color and line groups. (a) Edge image of a model object showing instances of closely-spaced parallelism between lines. (b) Some of the line groups extracted using the grouping algorithm using the constraints of* $t_{across} = 5, t_{along} = 0, t_{local-orient} = 6, t_{global-orient} = 10$. *(c) An edge image of a scene in which the model object appears. (d) The line groups in the image region selected by color that are possible matches to the model line groups of (b) under the allowed scale and pose changes. (e)- (f) The pairs of matching model and image line groups that found a good alignment transform. The circles here show the matching corner features within the line groups. (g) The model object projected into the image of (c) using the alignment transform given by the correspondence shown in (e) and (f).*

8 Conclusions

In this chapter we have examined the use of the property of closely-spaced parallelism between lines in performing data and model-driven selection. Towards this end, a scheme for grouping line segments was presented that possessed several features that make it compare favorably with other existing schemes of grouping of edges. First, closely-spaced parallelism occurs commonly within letter textures and contours of geometrical objects. Secondly, the groups generated tend to be compact and more likely to come from single objects (particularly in the data-driven mode). Also, the number of such groups is linear in the number of lines, and can be generated by a fast algorithm. Lastly, the size of the groups tends to be mostly small, except when merging across objects occurs (which can be reduced when grouping is restricted to prior selected regions). Thus grouping based on closely-spaced parallelism presented here satisfies most of the desirable requirements of grouping for recognition. Lastly, unlike in existing approaches to grouping, we have also examined the use of grouping in the model-driven mode. In doing so, an analysis of the changes that can occur to model line groups due to observation conditions was done, and this was found useful in performing model-driven selection using such line groups. Finally, since closely-spaced line groups tend to span a small portion of an object, they are good for achieving reliability in selection but may not be as useful in actually solving for the pose of the object, as for example, groups assembled using other constraints such as convexity.

References

[1] R.C. Bolles and R.A. Cain. Recognizing and locating partially visible objects: The local feature-focus method. *International Journal of Robotics Research*, 1(3):57–82, 1982.

[2] R. Brooks. Symbolic reasoning among 3d models and 2d images. *Artificial Intelligence*, 17:285–348, 1981.

[3] D.T. Clemens. *Region-based feature interpretation for recognizing 3D models in 2D images.* PhD thesis, Artificial Intelligence Lab, M.I.T., AI-TR-1307, 1991.

[4] D.T. Clemens and D. Jacobs. Model-group indexing for recognition. In *DARPA IU Workshop*, pages 604–613, 1990.

[5] D.T. Clemens and D.W. Jacobs. Space and time bounds on indexing 3d models from 2d images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:1007–1017, 1991.

[6] T.H. Cormen, C.E. Lieserson, and R.L. Rivest. *Introduction to Algorithms.* New York: McGraw Hill, Cambridge: MIT Press, 1990.

[7] J. Dolan and R. Weiss. Perceptual grouping of curved lines. In *Proc. DARPA IU Workshop*, pages 1135–1145, 1989.

[8] R.O. Duda and P.E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of ACM*, 15(1):11–15, 1972.

[9] W.E.L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* Cambridge: MIT Press, 1990.

[10] J.E. Hochberg. Effects of the gestalt revolution: The Cornell symposium on perception. *Psychological Review*, 64(2):73–84, 1957.

[11] D. P. Huttenlocher and P.C. Wayner. Finding convex edge groupings in an image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 406–412, 1991.

[12] D.P. Huttenlocher. *Three-dimensional recognition of solid objects from a two-dimensional image.* PhD thesis, Artificial Intelligence Lab, M.I.T., AI-TR-1045, 1988.

[13] D. Jacobs. The use of grouping in visual object recognition. Master's thesis, Artificial Intelligence Lab, M.I.T., AI-TR-1023, 1988.

[14] H.H. Jordan and F.M. Porter. *Descriptive Geometry.* Boston: Ginn and Company, 1929.

[15] Y. Lamdan and H.J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proceedings of the International Conference on Computer Vision*, pages 218–249, 1988.

[16] Y.G. LeClerc. Region grouping using the minimum description length principle. In *Proc. DARPA IU Workshop*, pages 473–481, 1990.

[17] D. G. Lowe. *Perceptual Organization and Visual Recognition.* Boston: Kluwer Academic, 1985.

[18] D. Marr. *Vision.* San Francisco: W.H. Freeman and Co., 1982.

[19] R. Mohan and R. Nevatia. Perceptual organization for segmentation and description. In *Proc. DARPA IU Workshop*, pages 415–424, 1989.

[20] T. Pavlidis. *Algorithms for Graphics and Image Processing.* Rockville: Computer Science Press, 1982.

[21] G. Reynolds and J.R. Beveridge. Searching for geometric structure in images of natural scenes. In *DARPA IU Workshop*, pages 257–271, 1987.

[22] L.G. Roberts. Machine perception of three-dimensional objects. In Tippet et al., editors, *Optical and Electro-Optical Information Processing*, pages 159–197. Cambridge: MIT Press, 1966.

[23] A. Shashua. Correspondence and affine shape from two orthographic views: Motion and recognition. Technical report, Artificial Intelligence Lab, M.I.T., AI-Memo-1327, December 1991.

[24] A. Shashua and S. Ullman. Structural saliency : The detection of globally salient structures using a locally connected network. In *Proceedings of the International Conference on Computer Vision*, pages 321–327, 1988.

[25] A. Shashua and S. Ullman. Grouping contours by iterated pairing network. In R. Lippman, J.E. Moody, and D.S. Touretzky, editors, *Advances in Neural Information Processing Systems 3*, pages 335–341. San Mateo: Morgan Kaufmann Inc., 1991.

[26] F. Stein and G. Medioni. Recognition of 3d objects from 2d groupings. In *Proc. DARPA IU Workshop*, pages 667–674, 1992.

[27] T.F. Syeda-Mahmood. Data and model-driven selection using color regions. In *Proceedings of the European Conference on Computer Vision*, pages 321–327, 1992.

[28] T.F. Syeda-Mahmood. Data and model-driven selection using color regions. Technical report, Artificial Intelligence Lab, M.I.T., AI-Memo-1270, February 1992.

[29] T.F. Syeda-Mahmood. Data and model-driven selection using texture regions. Technical report, Artificial Intelligence Lab, M.I.T., AI-Memo-1400, 1993.

[30] A. Tuller. *A Modern Introduction to Geometries.* Princeton: D. Von Nostrand Co., 1967.

[31] S. Ullman and R. Basri. Recognition by linear combination of models. Technical report, Artificial Intelligence Lab, M.I.T., AI-Memo-1152, August 1989.

[32] S. Ullman and R. Basri. Recognition by linear combination of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 992–1006, October 1991.

[33] M. Wertheimer. Principles of perceptual organization. In D. Beardslee and M. Wertheimer, editors, *Readings in Perception*, pages 115–135. Princeton: Princeton University Press, 1958.

[34] A.P. Witkin and J. M. Tenenbaum. On the role of structure in vision. In Beck, Hope, and A. Rosenfield, editors, *Human and Machine Vision*, pages 481–543. New York: Academic Press, 1983.