Object Recognition in High Clutter Images Using Line Features

Philip David[1,2] and Daniel DeMenthon[1]

[1]University of Maryland Institute for Advanced Computer Studies, College Park, MD 20742
[2]Army Research Laboratory, 2800 Powder Mill Road, Adelphi, MD 20783-1197

Abstract

We present an object recognition algorithm that uses model and image line features to locate complex objects in high clutter environments. Finding correspondences between model and image features is the main challenge in most object recognition systems. In our approach, corresponding line features are determined by a three-stage process. The first stage generates a large number of approximate pose hypotheses from correspondences of one or two lines in the model and image. Next, the pose hypotheses from the previous stage are quickly ranked by comparing local image neighborhoods to the corresponding local model neighborhoods. Fast nearest neighbor and range search algorithms are used to implement a distance measure that is unaffected by clutter and partial occlusion. The ranking of pose hypotheses is invariant to changes in image scale, orientation, and partially invariant to affine distortion. Finally, a robust pose estimation algorithm is applied for refinement and verification, starting from the few best approximate poses produced by the previous stages. Experiments on real images demonstrate robust recognition of partially occluded objects in very high clutter environments.

1. Introduction

Object recognition in cluttered environments is a difficult problem with widespread applications. Most approaches to object recognition rely on first finding correspondences between model features and image features, then computing a hypothesized model pose, and finally searching for additional image features that support this pose. The most challenging part of this process is the identification of corresponding features when the images are affected by clutter, partial object occlusion, changes in illumination, and changes in viewpoint. In fact, once the feature correspondence problem is solved, object recognition becomes almost trivial. A wide variety of features have been employed by object recognition systems, including points, edges, and textured regions.

The surfaces of many objects consist of regions of uniform color or texture. Most of the information available for object recognition is at the boundaries (edges) of these regions. Approaches to object recognition that rely on variations in texture internal to these regions are likely to perform poorly. Furthermore, objects that are composed of thin, stick-like components, such as bicycles, are especially difficult for texture-based approaches because background clutter will be present within a few pixels of any object pixel, thus corrupting local texture templates. This paper presents a simple, effective, and fast method for recognizing partially occluded 2D objects in cluttered environments, where the object models and their images are each described by sets of line segments. A fair amount of perspective distortion is tolerated by the algorithm, so the algorithm is also applicable to 3D objects that are represented by sets of viewpoint-dependent 2D models.

A three-stage process is used to locate objects. In the first stage, a list of approximate model pose hypotheses is generated. Every pairing of a model line to an image line first contributes a pose hypothesis consisting of a similarity transformation. When both the model line and the corresponding image line form corner-like structures with other nearby lines, and the angles of the corners are similar (within $45°$), a pose hypothesis consisting of an affine transformation is added to the hypothesis list, one for each such compatible corner correspondence. Typically, each model-to-image line correspondence contributes one to six poses to the hypothesis list.

We make use of information inherent in a single line correspondence to reduce the number of correspondences that must be examined in order to find an approximately correct pose. For m model lines and n image lines, we generate $\mathcal{O}(mn)$ approximate pose hypotheses. Compare this to traditional algorithms that generate precise poses from three pairs of correspondences, where there are up to $\mathcal{O}(m^3n^3)$ pose hypotheses. An approach such as RANSAC [9], which examines a very small fraction of these hypotheses, still has to examine $\mathcal{O}(n^3)$ poses to ensure with probability 0.99 that a correct precise pose will be found [6]. By starting with an approximate pose instead of a precise pose, we are able to greatly reduce the number of poses that need to be examined, and still find a correct precise pose in the end.

Most of the pose hypotheses will be inaccurate because

most of the generating correspondences are incorrect. The second stage of our approach ranks each pose hypothesis based on the similarity of the corresponding local neighborhoods of lines in the model and image. The new similarity measure is largely unaffected by image clutter, partial occlusion, and fragmentation of lines. Nearest-neighbor search is used in order to compute the similarity measure quickly for many pose hypotheses. Because this similarity measure is computed as a function of approximate pose, the ranking of the pose hypotheses is invariant to image translation, scaling, rotation, and partially invariant to affine distortion of the image. By combining the process of pose hypothesis generation from assumed unfragmented image lines with the neighborhood similarity measure, we are able to quickly generate a ranked list of approximate model poses which is likely to include a number of highly ranked poses that are close to the correct model pose.

The final stage of our approach applies a more time-consuming but also more accurate pose refinement and verification algorithm to a few of the most highly ranked approximate poses. Gold's graduated assignment algorithm [11], modified for line correspondences, is used for this purpose because it is efficient, tolerant of clutter and occlusion, and doesn't make hard correspondence decisions until an optimal pose is found.

Our approach assumes that at least one model line is detected as an unfragmented line in the image. By *unfragmented*, we mean that the corresponding image line is extracted from the image as a single continuous segment between the two endpoints of the projected model line. This necessarily requires that at least one model line be unoccluded. Additional model lines must be present in the image for verification, but these may be partially occluded or fragmented. A potential difficulty with this approach is that line detection software often fragment lines due to difficulties in parameter selection, and they usually don't extract lines completely at the intersections with other lines. The issue of fragmentation resulting from poor parameter selection can be ameliorated through post-processing steps that combine nearby collinear lines. However, this has not been necessary in any of our experiments. The issue of line detection software being unable to accurately locate the endpoints of lines at the intersections with other lines does not cause a problem because a few missing pixels at the ends of a line does not significantly affect the computed model transformations except in the case that the object's image is so small as to make recognition difficult regardless of how well the object's edges are detected. Furthermore, in evaluations of our line detection algorithm on images similar to those described in this paper [7], we have determined that 11% of all partially and fully visible model lines are detected in the images with less than one pixel error in the positions of their endpoints, and that 35% of all model lines

```
Create a structure for nearest neighbor searches of image lines.
Using a range search, identify corners in each model and in the image.
for each model do
  H = Ø.                                  // Initialize hypothesis list to empty.
  for each pair of model line l and image line l', do
    C = Pose hypotheses generated from l, l', and nearby corners.
    H = H ∪ C.
    Evaluate neighborhood similarity of model and image for poses C.
  end for
  P = Sort H based on neighborhood similarity measure.
  for i = 1 to N do
    Apply graduated assignment starting from pose P(i).
    if a suffi cient number of line correspondences are found then
      An object has been recognized.
    end if
  end for
end for
```

Figure 1: Outline of the new object recognition algorithm. The constant N is the number of pose refinements performed; good performance is obtained with $N = 4$ (see Sec. 7).

are detected as image segments where the sum of the errors in the endpoint positions is no more than 5% of the length of the corresponding projected model lines. We find that 5% relative error is small enough to obtain a good coarse pose hypothesis, and that with 35% of the model lines having relative errors no larger than this, there will be many such good hypotheses that will allow the pose refinement stage of the algorithm to recognize an object.

This three-stage approach allows CPU resources to be quickly focused on the highest payoff pose hypotheses, which in turn results in a large reduction in the amount of time needed perform object recognition. An outline of the algorithm is shown in Fig. 1.

In the following sections, we first describe related work, and then describe each step of our algorithm in more detail. Experiments with real imagery containing high levels of clutter and occlusion (see Fig. 4, for example) demonstrate the effectiveness of the algorithm. The new algorithm is faster and able to handle greater amounts of clutter than previous approaches that use line features. The approach is able to recognize planar objects that are rotated by up to 60° away from their modeled viewpoint, and recognize 3D objects from 2D models that are rotated by up to 30° from their modeled viewpoint.

2. Related Work

A wide variety of approaches to object recognition have been proposed since Robert's ground-breaking work on recognizing 3D polyhedral objects from 2D perspective images [16]. Among the pioneering contributions are Fischler and Bolles' RANSAC method [9], Baird's tree-pruning method [2], and Ullman's alignment method [18]. These approaches, which hypothesize poses from small sets of correspondences and reject or accept those poses based on

the presence of supporting correspondences, become intractable when the number of model and image features becomes large.

More recently, the use of rich feature descriptors has become popular as a way of reducing the number of feature correspondences that must be examined. The Harris corner detector [12] has seen widespread use for this purpose; however, it is not stable to changes image scale, so it performs poorly when matching models and images of different scales. Schmid and Mohr [17] have developed a rotationally invariant feature descriptor using the Harris corner detector. Lowe [13] extended this work to scale invariant and partially affine invariant features with his SIFT approach, which uses scale-space methods to determine the location, scale, and orientation of features, and then, relative to these parameters, a gradient orientation histogram describing the local texture. Excellent results have been obtained by approaches using rich features when objects have significant distinctive texture. However, there are many common objects that possess too little distinctive texture for these methods to be successful. Examples include thin objects such as bicycles and ladders where background clutter will be present near all object boundaries, and uniformly textured objects such as upholstered furniture. In these cases, only the relations between geometric features (such as points and edges) can be used for matching and object recognition.

A number of more recent works [15, 4] have also used edges for object recognition of poorly textured objects. Edges are stable features and are easy to locate on both textured and nontextured objects. Mikolajczyk et al. [15] generalize Lowe's SIFT descriptors to edge images, where the position and orientation of edges are used to create local shape descriptors that are orientation and scale invariant. Carmichael's approach [4] uses a cascade of classifiers of increasing aperture size, trained to recognize local edge configurations, to discriminate between object edges and clutter edges; this method, however, is not invariant to changes in image rotation or scale.

Gold and Rangarajan [11] simultaneously compute pose and 2D-to-2D or 3D-to-3D point correspondences using deterministic annealing to minimize a global objective function. We previously used this method [5] for matching 3D model lines to 2D image lines, and it is used here for the pose refinement stage of our algorithm. Beveridge [3] matches points and lines using a random start local search algorithm. Denton and Beveridge [8] extended this work by replacing random starts with a heuristic that is used to select which initial correspondence sets to apply the local search algorithm. Although we use line features instead of point features, Denton's approach is conceptually similar to ours in a number of ways. Both approaches first hypothesize poses using small sets of local correspondences, then sort the hypotheses based on a local match error, and finally apply a pose refinement and verification algorithm to a small number of the best hypotheses. Significant differences between the two approaches are that ours uses lines instead of points, and also zero or one neighboring features, instead of four, to generate pose hypotheses; so our approach will have many fewer hypotheses to consider, and each hypothesis is much less likely to be corrupted by spurious features (clutter).

Our approach also has some similarities to Ayache and Faugeras's HYPER system [1]. They generate 2D similarity pose hypotheses from correspondences of "compatible" model and image line segments, then rank these hypotheses, and finally use a tree-pruning algorithm to refine the best hypotheses. In contrast, our pose hypotheses are based on affine transformations instead of similarity transformations, our hypotheses ranking is based on a dissimilarity measure that is less affected by line fragmentation because it does not depend on the lengths of lines nor on unique reference points on the lines, and we use the more robust and efficient graduated assignment algorithm [11] for pose refinement.

3. Generating Pose Hypotheses

We wish to generate a small set of approximate poses that, with high certainty, includes at least one pose that is close to the true pose of the object. The smaller the number of correspondences used in estimating a pose, the less likely the estimated pose will be corrupted by spurious correspondences. From a single correspondence of a model line to an image line, where the image line may be fragmented (only partially detected due to partial occlusion or faulty line detection), we can compute the 2D orientation of the model as well as a one-dimensional constraint on its position, but the scale and translation of the model cannot be determined; this does not provide sufficient geometric constraints to evaluate the similarity of a local region of the model with a local region of the image.

On the other hand, if we assume that a particular image line is unfragmented, then from a single correspondence of a model line to this image line we can compute a 2D similarity transformation of the model. This is possible because the two endpoints of the unfragmented image line must correspond to the two endpoints of the model line, and two corresponding points are sufficient to compute a similarity transformation. A similarity transformation will be accurate when the viewing direction used to generate the 2D model is close to the viewing direction of the object. Because we don't know which endpoint of the model line corresponds to which endpoint of the image line, we consider both possibilities and generate a similarity transformation for each. For $\mathbf{p}_1$ and $\mathbf{p}_2$ model line endpoints corresponding to image line endpoints $\mathbf{q}_1$ and $\mathbf{q}_2$, respectively, the similarity transformation mapping the model to the image is $\mathbf{q}_i = A\mathbf{p}_i + \mathbf{t}$

where A and $\mathbf{t}$ are

$$\begin{aligned} A &= \frac{\|\mathbf{q}_1-\mathbf{q}_2\|}{\|\mathbf{p}_1-\mathbf{p}_2\|}\begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix}, \\ \mathbf{t} &= \mathbf{q}_1 - A\mathbf{p}_1, \end{aligned}$$

and where θ is the rotation angle (in the range $-\pi$ to π, clockwise being positive) from $\mathbf{p}_1 - \mathbf{p}_2$ to $\mathbf{q}_1 - \mathbf{q}_2$.

We can obtain more accurate approximate poses with little additional work when the model line and the unfragmented image line (called the *base lines* below) form corner-like structures with other lines: corners in the model should correspond to corners in the image. Corners in the model are formed by pairs of model lines that terminate at a common point, while corners in the image are formed by pairs of image lines that terminate within a few pixels of each other. By looking at corners, we expand our search to correspondences of two line pairs. As before, we assume only that the base image line is unfragmented; other image lines may be fragmented. If a base model line forms a corner with another model line, and if the base image line is unfragmented, then all model lines that share an endpoint with the base model line should be unoccluded around that endpoint in the image, and therefore there is a good chance that these other models lines will appear in the image near the corresponding endpoint of the base image line. Thus, looking at corners formed with the base image lines provides a way of finding additional line correspondences with a low outlier rate.

The model and image lines which participate in corner structures are quickly located using a range search algorithm [14]. To generate pose hypotheses for a particular base correspondence, the angles of corners formed with the base model line are compared to the angles of corners formed with the base image line. An affine pose hypothesis is generated for any pair of corner angles that are within $45°$. As before, this is repeated for each of the two ways that the base model line can correspond to the base image line. Note that these affine pose hypotheses are generated in addition to the similarity pose hypotheses describe above.

An affine pose hypothesis is generated as follows. Let $\mathbf{p}_1$ and $\mathbf{p}_2$ be the endpoints of the base model line, and $\mathbf{q}_1$ and $\mathbf{q}_2$ be the corresponding endpoints of the base image line. See Fig. 2. Assume that a pair of corners is formed with the base lines by model line p and image line q that terminate near endpoints $\mathbf{p}_1$ and $\mathbf{q}_1$, and have angles θ_p and θ_q, respectively. We have two pairs of corresponding points and one pair of corresponding angles. Since a 2D affine transformation has 6 degrees of freedom but we have only 5 constraints (two for each point correspondence, and one for the angle correspondence), we impose the additional constraint that the affine transformation must scale the length of line p in the same way as it does the length of the base model line $\mathbf{p}_1\mathbf{p}_2$. This, defines a third pair of corresponding points $\mathbf{p}_3$ and $\mathbf{q}_3$, on p and q, respectively, as shown in

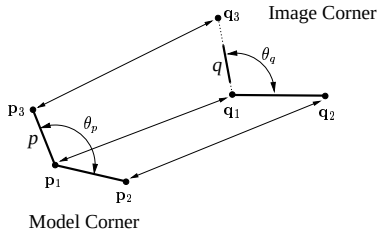


Figure 2: Geometry for calculation of the approximate affine transformation.

Fig. 2. $\mathbf{p}_3$ is the second endpoint of p, and $\mathbf{q}_3$ is the point collinear with q such that

$$\frac{\|\mathbf{p}_2 - \mathbf{p}_1\|}{\|\mathbf{q}_2 - \mathbf{q}_1\|} = \frac{\|\mathbf{p}_3 - \mathbf{p}_1\|}{\|\mathbf{q}_3 - \mathbf{q}_1\|}.$$

$\mathbf{q}_3$ is found to be

$$\mathbf{q}_1 + \frac{\|\mathbf{p}_3 - \mathbf{p}_1\|\,\|\mathbf{q}_2 - \mathbf{q}_1\|}{\|\mathbf{p}_2 - \mathbf{p}_1\|^2}\begin{bmatrix}\cos\theta_q & -\sin\theta_q \\ \sin\theta_q & \cos\theta_q\end{bmatrix}(\mathbf{p}_2 - \mathbf{p}_1).$$

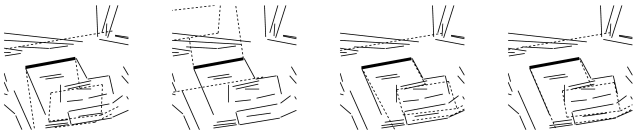
The affine transformation mapping a model point $\mathbf{p}_i = \begin{bmatrix} p_{i_x} & p_{i_y} \end{bmatrix}^\mathsf{T}$ to the image point $\mathbf{q}_i = \begin{bmatrix} q_{i_x} & q_{i_y} \end{bmatrix}^\mathsf{T}$ is

$$\begin{bmatrix} q_{i_x} \\ q_{i_y} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}\begin{bmatrix} p_{i_x} \\ p_{i_y} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \qquad (1)$$

For each correspondence $\mathbf{p}_i \leftrightarrow \mathbf{q}_i$ we have two linear equations in the 6 unknowns a_1, a_2, a_3, a_4, t_x, and t_y. From the three corresponding points, we can solve for the parameters of the affine transformation. Fig. 3 shows the pose hypothesis generated for a particular correct base correspondence.

4. Similarity of Line Neighborhoods

The second stage of the recognition algorithm ranks all hypothesized approximate model poses in the order that the pose refinement algorithm should examine them; the goal is to rank highly those poses that are most likely to lead the refinement algorithm to a correct precise pose. For this purpose, a geometric measure of the similarity between the model (transformed by an approximate pose) and the image is computed. To ensure that this similarity measure can be computed quickly, for any base model line generating a hypothesized pose, only a local region of model lines surrounding the base line (called the base model line's *neighborhood*) is compared to the image lines. Let $\mathcal{M}$ be the set of lines for a single model and $\mathcal{I}$ be the set of image lines. We define the *neighborhood radius* of a line l to be the smallest distance, denoted $r(l)$, such that the two endpoints of at least N_{nbr} lines (excluding l) are within distance $r(l)$ of l. In all of our experiments, the value of N_{nbr} is fixed



(a) (b) (c) (d)

Figure 3: The pose hypotheses generated for a correct correspondence of a real model and image line are shown. The model lines (dashed lines and thick black line) are shown overlaid on the image lines (thin lines). The one thick line in each image shows the base correspondence: a model line perfectly aligned with an image line. (a) and (b) show the two similarity transformations, and (c) and (d) show the two affine transformations. These are the complete set of transformations hypothesized for this base correspondence. Notice the better alignment in images (c) and (d), resulting from the use of corner angle correspondences, compared to image (a).

at 10 lines ($N_{\text{nbr}} \leq |\mathcal{M}|$). The neighborhood of a model line l is the set of N_{nbr} model lines, $\mathcal{N}(l)$, whose endpoints are within distance $r(l)$ of l.

For a hypothesized approximate model pose $\{A, \mathbf{t}\}$ generated for a base model line l, let $\mathcal{T}(\mathcal{N}(l), A, \mathbf{t})$ denote the neighbors of l transformed by the pose $\{A, \mathbf{t}\}$, and let $d(l', l'')$ denote the distance (defined in Sec. 5) between two lines l' and l'' in the image. Then, the geometric similarity between a model neighborhood $\mathcal{N}$ transformed by the pose $\{A, \mathbf{t}\}$ and the set of image lines $\mathcal{I}$ is

$$S(\mathcal{N}, \mathcal{I}, A, \mathbf{t}) = \sum_{l' \in \mathcal{T}(\mathcal{N}, A, \mathbf{t})} \min \left\{ S_{\max}, \min_{l'' \in \mathcal{I}} d(l', l'') \right\}. \tag{2}$$

The smaller the value of $S(\mathcal{N}, \mathcal{I}, A, \mathbf{t})$, the more "similar" a model neighborhood $\mathcal{N}$ is to the image $\mathcal{I}$ under the transformation $\{A, \mathbf{t}\}$. The parameter $S_{\max}$ ensures that "good" poses are not penalized too severely when a line in the model is fully occluded in the image. This parameter is easily set by observing the values of $S(\mathcal{N}, \mathcal{I}, A, \mathbf{t})$ that are generated for poor poses (that should be avoided), and then setting $S_{\max}$ to this value divided by N_{nbr}.

As explained in Sec. 5, the distance between a single model neighbor and the closest image line can be found in time $\mathcal{O}(\log n)$ when there are n image lines. Since $|\mathcal{N}| = N_{\text{nbr}}$, the time to compute $S(\mathcal{N}, \mathcal{I}, A, \mathbf{t})$ is $\mathcal{O}(\log n)$.

5. Distance Between Lines

For any image line l' (which is typically a transformed model line), we wish to efficiently find the line $l'' \in \mathcal{I}$ that minimizes $d(l', l'')$ in Eq. 2. This search can be performed efficiently when each line is represented by a point in an N-dimensional space and the distance between two lines is the Euclidean distance between the corresponding points in this space. Assuming that we have a suitable line representation, a tree data structure storing these N-dimensional points can be created in time $\mathcal{O}(n \log n)$ and the closest image line can be found in time $\mathcal{O}(\log n)$. This tree structure need only be created once for each image, and is independent of the model lines.

Thus, we want to represent each line as a point in an N-dimensional space such that the Euclidean distance between two lines is small when the two lines are superposed. We would also like the distance function to be invariant to partial occlusion and fragmentation of lines. We could use the midpoint and orientation of a line, but a short line superposed on a longer line (think of the short line as a partially occluded version of the longer line) could be assigned a large distance because their midpoints may be far. Further, there is problem associated with line orientation because a line with an orientation of θ should produce the same distance as when its orientation is given as $\theta \pm 2k\pi$ for $k = 1, 2, \ldots$. For example, two lines with identical midpoints but orientations $179°$ and $-179°$ should produce the same distance as if the orientations of the two lines were $1°$ and $-1°$. It is not possible with a Euclidean distance function to map both of these pairs of angles to the same distance. A solution to these occlusion and orientation problems is to generate multiple representations of each line.

Let l be a line with orientation θ (relative to the horizontal, $0 \leq \theta \leq \pi$) and endpoints $[x_1, y_1]$ and $[x_2, y_2]$. When l is a line in the image ($l \in \mathcal{I}$), l is represented by the two 3D points

$$\left[\frac{\theta}{r_\theta}, \frac{x_{\text{mid}}}{r_m}, \frac{y_{\text{mid}}}{r_m}\right] \quad \text{and} \quad \left[\frac{\theta - \pi}{r_\theta}, \frac{x_{\text{mid}}}{r_m}, \frac{y_{\text{mid}}}{r_m}\right] \tag{3}$$

where $[x_{mid}, y_{mid}] = [x_1 + x_2, y_1 + y_2]/2$ is the midpoint of the line, and where r_θ and r_m are constant scale factors chosen to normalize the contribution of the orientation and position components to the distance measure [7].

When l is a transformed model line (as in l' above), l is represented by the set of 3D points

$$\left\{ \left[\frac{\theta}{r_\theta}, \frac{\bar{x}_i}{r_m}, \frac{\bar{y}_i}{r_m}\right], \left[\frac{\theta - \pi}{r_\theta}, \frac{\bar{x}_i}{r_m}, \frac{\bar{y}_i}{r_m}\right], i = 1, 2, \ldots, N_{\text{pts}} \right\} \tag{4}$$

where

$$N_{\text{pts}} = \left\lceil \frac{\|[x_2 - x_1, y_2 - y_1]\|}{w} \right\rceil + 1, \tag{5}$$

$$\Delta = \frac{[x_2 - x_1, y_2 - y_1]}{N_{\text{pts}} - 1},$$

$$[\bar{x}_i, \bar{y}_i] = [x_1, y_1] + (i - 1)\Delta.$$

In words, two orientations are used for each transformed model line, but the position of the line is represented by a

series of N_{pts} points that uniformly sample the length of the line at an interval w. The reason multiple sample points are required to represent the position of transformed model lines but not the image lines is that when $\{A, \mathbf{t}\}$ is a correct pose for the model, the image line, which may be partially occluded or otherwise fragmented, will in general be shorter than the transformed model line. In this case, the midpoint of the image line will lie somewhere along the transformed model line, but the midpoint of the transformed model line may lie off of the image line. We have found that placing a sample point at approximately every 10^{th} pixel along each transformed model line is sufficient to solve the occlusion problem of the representation. Then, when a transformed model line l' is correctly aligned with a possibly partially occluded image line l'', we will have $d(l', l'') \leq w/r_m$.

Then, for any transformed model line l', to find the image line l'' that minimizes $d(l', l'')$ we simply query the nearest neighbor data structure (generated using points from Eq. 3) with all of the points listed in Eq. 4 and then use the distance of the closest one. Because the complexity of the nearest neighbor search is $\mathcal{O}(\log n)$, the use of multiple points to represent lines does not significantly slow down the algorithm.

6. Graduated Assignment for Lines

The final stage of the object recognition algorithm is to apply a pose refinement and verification algorithm to the few "best" approximate poses. We use the graduated assignment algorithm [11] for this purpose because it is efficient ($\mathcal{O}(mn)$ complexity for m model lines and n image lines), robust to occlusion and clutter, and doesn't make hard correspondence decisions until a locally optimal pose is found. Given m model lines $\mathcal{M} = \{l_j, j = 1, \ldots, m\}$, n image lines $\mathcal{I} = \{l'_k, k = 1, \ldots, n\}$, and an approximate model pose $T_0 = \{A_0, \mathbf{t}_0\}$, we wish to find the 2D affine transformation $T = \{A, \mathbf{t}\}$ and the $(m+1) \times (n+1)$ *match matrix* M that minimizes the objective function

$$E = \sum_{j=1}^{m} \sum_{k=1}^{n} M_{jk} \left(d\left(T\left(l_j\right), l'_k\right)^2 - \delta^2 \right). \qquad (6)$$

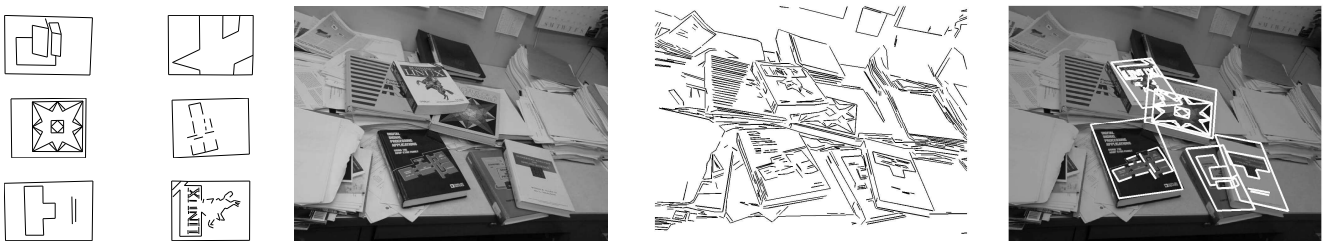
Here, $T(l)$ denotes the transformation of line l by T, and δ is the maximum distance between matched lines. M defines the correspondences between model lines and image lines; it has one row for each of the m model lines and one column for each of the n image lines. This matrix must satisfy the constraint that each model line match at most one image line, and vice versa. By adding an extra row and column to M, *slack row* $m+1$ and *slack column* $n+1$, these constraints can be expresses as $M_{jk} \in \{0, 1\}$ for $1 \leq j \leq m+1$ and $1 \leq k \leq n+1$, $\sum_{i=1}^{n+1} M_{ji} = 1$ for $1 \leq j \leq m$, and $\sum_{i=1}^{m+1} M_{ik} = 1$ for $1 \leq k \leq n$. A value of 1 in the slack column $n+1$ at row j indicates that the j^{th} model line does not match any image line. A value of 1 in the slack row $m+1$ at column k indicates that the k^{th} image line does not match any model line.

The graduated assignment algorithm uses deterministic annealing to convert the discrete problem (for a binary match matrix) into a continuous one that is indexed by a control parameter; the control parameter determines the uncertainty of the match matrix, and hence the amount of smoothing implicitly applied to the objective function. The match matrix minimizing the objective function is tracked as this control parameter is slowly adjusted to force the continuous match matrix closer and closer to a binary match matrix. The details of this process may be found in [7].

7. Experiments

To validate our approach, we recognized partially occluded 2D and 3D objects in cluttered environments under a wide variety of viewpoints. All images were acquired at a resolution of 800×600 pixels. 400 to 800 lines were typically detected in an image, and models each had between 20 and 80 lines. First, we used books to test the recognition of planar objects. Fig. 4 illustrates recognition results when our algorithm is applied to an image of a pile of books. For all but one of the five books, the pose hypotheses leading to correct recognition was found in the top 10 hypotheses of the sorted list. One book ("Linux", shown in the lower right of Fig. 4) was not found until the 24^{th} pose hypothesis.

The performance of our algorithm depends on how reliably it can move to the top of the sorted hypothesis list those hypotheses associated with correct correspondences. To evaluate this, we estimate $P_\theta(k)$, the probability that one of the first k sorted pose hypotheses for a model leads to a correct recognition when the viewpoint of the object and the viewpoint used to generate its model differ by an angle of θ, assuming that an instance of the model does in fact appear in the image. Knowing $P_\theta(k)$ allows one to determine how many pose hypotheses should be examined by the pose refinement process before restarting with a new model, either of a new object or of the same object but from a different viewpoint. Because $P_\theta(k)$ is highly dependent on the amount and type of clutter and occlusion in an image, and because the level of clutter and occlusion present in our test was held fixed, $P_\theta(k)$ should be interpreted loosely. The six books shown in Fig. 4a were used to perform this experiment. All six books were placed flat on a table along with a number of other objects for clutter. Each book in turn was moved to the center of the table and then rotated on the plane of the table to 8 different orientations, where each orientation was separated by approximately $45°$. For each of these orientations, the camera was positioned at angles $0°$, $10°$, $20°$, $30°$, $40°$, $50°$, $60°$, and $70°$ relative to the normal of the table ($0°$ is directly overhead) and at a fixed distance from the central book, and then an image was acquired. The



(a) Models of books. (b) Test image. (c) 731 detected lines. (d) Recognized books.

Figure 4: Five books, some partially occluded, are recognized in a cluttered environment.

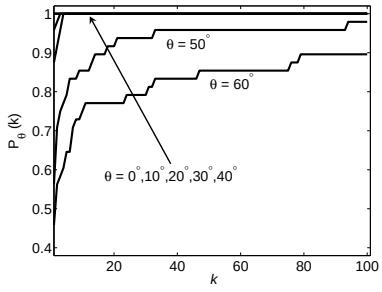


Figure 5: $P_\theta\ (k)$ is the probability that one of the first k sorted pose hypotheses for a model leads to a correct recognition for that model. θ is the difference in viewpoint elevation between the model and the object. For $\theta \leq 40°$, one of the four highest ranked pose hypotheses always leads to correct recognition. The curves for $\theta = 0°$ thru $40°$ are superposed for $k \geq 4$.

center books were unoccluded in these experiments. A separate image was also acquired of each book in an arbitrary orientation with the camera in the $0°$ position; these later images were used to generate the book models. We then applied our algorithm to each model and image pair and determined the position in the sorted hypothesis list of the first hypothesis that allowed the object to be recognized. Up to 100 hypotheses were examined for each model and image pair. The estimated values of $P_\theta\ (k)$ are shown in Fig. 5. From this we see that, for planar objects whose orientations differ by up to $60°$ from the modeled orientation, a probability of correct recognition of 0.8 can be achieved by examining the first 30 pose hypotheses. By examining just the top four pose hypotheses, we can achieve a probability of correct recognition of 1.0 for objects whose orientations differ by up to $40°$ from the modeled orientations. Thus, a good strategy would be to apply the algorithm using a set of models for each object generated for every $40°$degree change in viewpoint.

Finally, we applied our algorithm to three 3D objects. We acquired 17 images of each object, where the objects were rotated by $2.5°$ between successive images. The first image of each object was used to represent the object, and from this a 2D model was generated by identifying the object edges in that image. Examining only the top 10 sorted pose hypotheses, all three objects were successfully recognized from all viewpoints that differed by up to $25°$ from the modeled viewpoint. Two of the objects (the monitor and hole punch) were also recognized at $30°$ away from the modeled viewpoints. Fig. 6 shows the object models, the images, the detected lines, and the final poses of the recognized objects for the most distant viewpoints that each was recognized. The range of recognizable object orientations could have been be extended somewhat by examining more pose hypotheses, but at some point it becomes more cost effective to add a new model for a different viewpoint.

8. Conclusions

We have presented an efficient approach to recognizing partially occluded objects in cluttered environments. Our approach improves on previous approaches by making use of information available in one or two line correspondences to compute approximate object poses. Only a few model lines need to be unfragmented in an image in order for our approach to be successful; this condition is easily satisfied in most environments. The use of one or two line correspondences to compute an objects pose allows for a large reduction in the dimensionality of the space that must be searched in order to find a correct pose. We then developed an efficiently computed measure of the similarity of two line neighborhoods that is largely unaffected by clutter and occlusion. This provides a way to sort the approximate model poses so that only a small number need to be examined by more time consuming algorithms. Experiments show that a single view of an object is sufficient to build a model that will allow recognition of that object over a wide range of viewpoints.

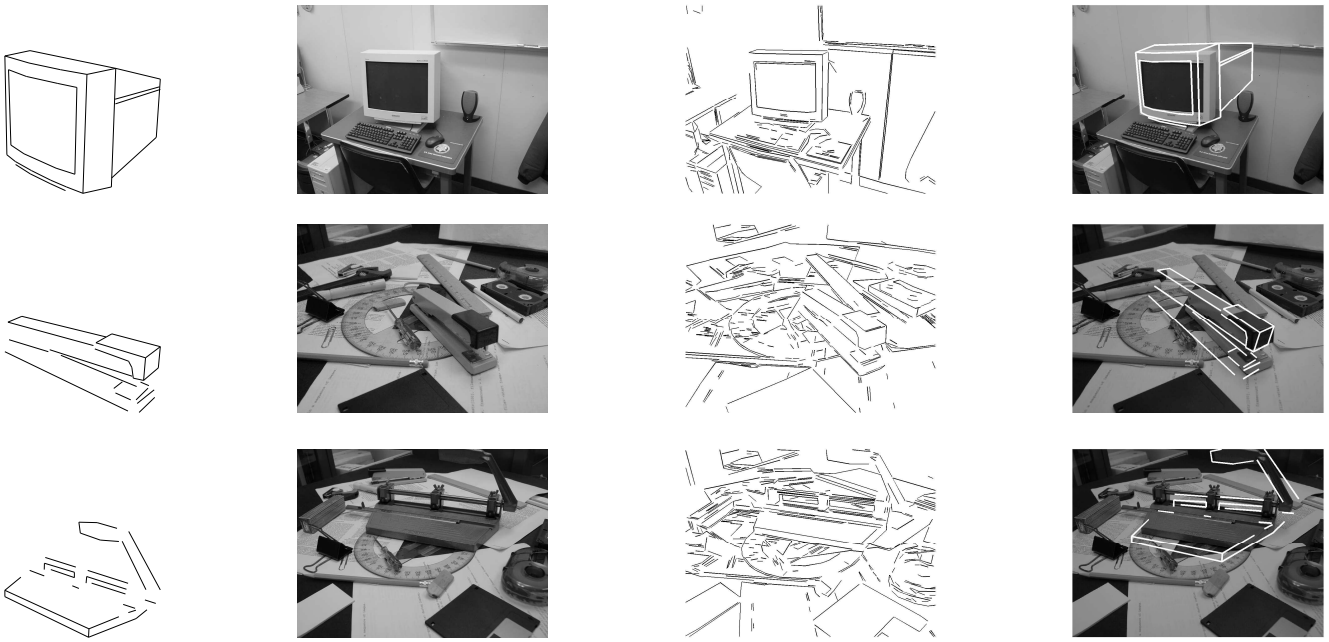


Figure 6: Recognition of 3D objects from viewpoint-dependent 2D models: computer monitor (top row), stapler (middle row), and hole punch (bottom row). Shown in each row, from left to right, is the 2D object model, original image, image lines, and model of recognized object overlaid on the original image. The modeled view of each object differs from the test view by $20 - 30^\circ$.

References

[1] N. Ayache and O.D. Faugeras, HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):44–54, January 1986.

[2] H.S. Baird, *Model-Based Image Matching Using Location*, MIT Press: Cambridge, MA, 1985.

[3] J.R. Beveridge and E.M. Riseman, Optimal Geometric Model Matching Under Full 3D Perspective, *Computer Vision and Image Understanding*, 61(3):351–364, May 1995.

[4] O. Carmichael and M. Hebert, Shape-Based Recognition of Wiry Objects, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(12):1537–1552, December 2004.

[5] P. David, D. DeMenthon, R. Duraiswami, and H. Samet, Simultaneous Pose and Correspondence Determination using Line Features, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. II-424–II-431, June 2003.

[6] P. David, D. DeMenthon, R. Duraiswami, and H. Samet, SoftPOSIT: Simultaneous Pose and Correspondence Determination, *Int. Journal of Computer Vision*, 49(3):259–284, 2004.

[7] P. David and D. DeMenthon, Object Recognition by Deterministic Annealing of Ranked Affine Pose Hypotheses, University of Maryland, College Park, MD, Dept. of Computer Science Technical Report CS-TR-4731, July 2005.

[8] J. Denton and J.R. Beveridge, Two Dimensional Projective Point Matching, *Proc. 5th IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 77–81, April 2002.

[9] M.A. Fischler and R.C. Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Comm. Association for Computing Machinery*, 24(6):381–395, June 1981.

[10] S. Gold and A. Rangarajan, A Graduated Assignment Algorithm for Graph Matching, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(4):377–388, 1996.

[11] S. Gold, A. Rangarajan, C.P. Lu, S. Pappu, E. Mjolsness. "New Algorithms for 2D and 3D Point Matching: Pose Estimation and Correspondence," *Pattern Recognition*, 31(8):1019–1031, August 1998.

[12] C.G. Harris and M.J. Stephens, "A Combined Corner and Edge Detector," *Proc. Fourth Alvey Vision Conference*, Manchester, pp. 147–151, 1988.

[13] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. Journal of Computer Vision*, 60(2):91–110, 2004.

[14] C. Merkwirth, U. Parlitz, I. Wedekind, and W. Lauterborn, "TSTOOL User Manual," University of Göttingen, http://www.physik3.gwdg.de/tstool/index.html.

[15] K. Mikolajczyk, A. Zisserman, and C. Schmid, "Shape Recognition with Edge-Based Features," *Proc. British Machine Vision Conference*, September 2003.

[16] L. Roberts, "Machine Perception of Three-Dimensional Solids," *Optical and Electrooptical Information Processing*, J. T. Tipett, Ed., M.I.T. Press, Cambridge, MA, pp. 159-197, 1965.

[17] C. Schmid and R. Mohr, "Local Grayvalue Invariants for Image Retrieval," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5):530–535, May 1997.

[18] S. Ullman, "Aligning Pictorial Descriptions: An Approach to Object Recognition," *Cognition*, 32(3):193–254, 1989.