An Efficient Indexing Scheme For Image Storage and Recognition

Mayez Al-Mouhamed *

Abstract

This paper presents a model-based vision system to achieve robust recognition of planar contours that are scale invariant of known models. Planar contours are partitioned into segments by using constant curvature criterion. A set of descriptors that are invariant with respect to scale, rotation, and translation are extracted from the geometric features of the segments. The descriptors are used to carry out efficient indexed search over the models so that to reduce the search space. Fragments of contours extracted from partially occluded scenes can be individually matched by using the local shape descriptors. Pruning of large portions of the models is carried out by keeping only some matched classes which received the highest vote. This significantly reduces the search and enables the use of finer matching operators such as comparing the positioning of segments in scene to positioning of matched segments in the model. More sophisticated matching is applied in later stages over much restricted number of hypotheses. Therefore, the dependency of the recognition time over the size of the models is significantly reduced. Evaluation shows the ability of our approach to recognize scenes with real partially occluded objects. Entirely visible objects are recognized with a reasonably high efficiency (80%) even with a change in view-point of up to 25°. The efficiency smoothly decreases but remains above 60% when the percentage of visible segments drops to 50% and the change in view-point is as above.

Keywords: Database, heuristic-search, pattern recognition, vision, segmentation

1 Introduction

An effective model-based recognition system [1] must be capable of retrieving the best matched objects as well as carrying out massive pruning of inconsistent models. Modeling objects by their local geometric features [2] takes advantage of the coarse shape and enables quick indexing of object features into the models in an attempt to reduce the complexity of the search space before carrying out finer pattern matching. Hierarchical object modeling partitions the object contour into a set of fragments so that each fragment is a set of features which are selected as invariant under translation and rotation [3, 4].

The efficiency of the matching depends to a large extent on the scalability [1, 5, 6] of the recognition operator which is the ability to recognize whole contours as well as fragments of

*The author is with the Department of Computer Engineering, College of Computer Science and Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia (mayez@ccse.kfupm.edu.sa).

contours while reducing the combinatorics of the search. For this, the extracted features [7] must be local and small enough to match wherever they are present but must also be stable and discriminative. Features are used as searching keys in some quick indexing/hashing schemes. Model organization was studied by Califano and Mohan [6] which proposed the use of multidimensional indexing to keep a relatively coarse bucket quantization without scarifying selectivity. The synergy of the indexing scheme must be small enough because all the models are potentially involved in the initial search [2, 8]. Grimson [9, 10] equally treats all the available features in generating hypotheses. This results in tree-matching structure that is scanned by using depth-first search. The search over the current sub-tree is abandoned when enough inconsistent evidences are accumulated and the next sub-tree is started.

Our objective is to optimize the library model and the search so that the recognition time would mainly depend on the scene complexity without explicit dependence on the size of the models. The model is designed so that the recognition algorithm spends a small fraction of time in global model processing while keeping the rate of correct classification as high as possible. To keep on pruning of inconsistent hypotheses we propose an efficient shape matching for comparing whole contours as well as fragments. To accelerate the search, indexed tables are created by using off-line sorting of the classes with respect to each feature. A combined algorithm is then used to recognize entire contours and partially occluded contours among a reasonably large dictionary.

This paper is organized as follows. Section 2 presents object modeling including edge detection, segmentation, model generation, extraction of shape descriptors, and model organization. Section 3 presents the recognition system including recognition of closed contours and recognition of partially occluded scenes. Section 4 presents the performance evaluation which includes a discussion of the proposed method, discussion of recognition time versus increase in database size, and comparison to other approaches. Section 5 concludes about this work.

2 Modeling object contours

This section presents the design of a library model including edge detection, segmentation, fine contour model, and coarse contour model. We also present a set of contour features that are useful for structural indexing of the model.

2.1 Edge detection

Edge detection is applied on gray images for extracting the shape of objects. The first derivative can be used to detect the presence of an edge at pixel $f(x, y)$ by evaluating the *Gradient* $G(f(x, y)) = (f'_x(x, y), f'_y(x, y))$, where $f'_x(x, y)$ and $f'_y(x, y)$ are the derivatives in the X's and Y's directions, respectively. The gradient magnitude can be approximated as $|f'_x(x, y)| + |f'_y(x, y))|$. To make the gradient less sensitive to noise we use the *sobel operator* that average the gradient over larger pixel neighborhood.

Segmentation is carried out by scanning the contour in a uni-directional manner and each border pixel p_i is associated the value of its *direction* ($d(p_i)$) with respect to previous pixel p_{i-1}. A starting pixel is one terminal pixel of the contour. The 3×3 convolution mask that is used by us for *direction coding* is shown in Figure 1-(a). A coded chain is shown in Figure 1-(b).

In partially occluded scenes each intersection of three chains is examined by its edge direction (Figure 1-(c)) in the neighborhood of the intersecting pixel. To resolve the ambiguity, intersecting chains having similar values of the gradient at the intersecting edges are merged together. This allows merging chains 1 and 3 which determines the connectivity of the three chains as shown in Figure 1-(d).

2.2 Segmentation

The objective is to associate to each scene object a model that represents a polygonal approximation of the contour. The algorithm we used for polygonal approximation consists of detecting break-points by comparing the average direction of a set of pixels to the gradient of the current reference segment. In other words, our method is based on successive conditional merging operations that are performed along the chain of directions until the least-squares error line fit of the merged directions thus exceeds a preset threshold. In this case, a break-point is detected, recorded, and a new segment is started. The process is continued until all chains have been visited.

Each segment is associated its length and its exterior angle with respect to the previous segment as shown in Figure 1-(e). The kth segment D_k is formed by a pair of break-points $b_k = (x_k, y_k)$ and $b_{k+1} = (x_{k+1}, y_{k+1})$. The length of D_k is $s_k = (\Delta x_k^2 + \Delta y_k^2)^{1/2}$, where $\Delta x_k = x_{k+1} - x_k$ and $\Delta y_k = y_{k+1} - y_k$. The angle $\theta(s_k)$ between segments D_{k-1} and D_k is evaluated as the exterior angle which is defined by $\theta(s_k) = \cos^{-1}((\Delta x_{k-1}.\Delta x_k + \Delta y_{k-1}.\Delta y_k)/(s_{k-1}.s_k))$. The segment length and exterior angle are shown in Figure 1-(e). The correct sign of $\theta(s_k)$ can be found by examining the coordinates of b_{k-1}, b_k, and b_{k+1}.

2.3 Modeling the contour

A fine *angle-length* model of contour $F = \{(\theta(\rho_i), \rho_i)\}$ is an ordered set of straight segments with lengths ρ_i and exterior angles $\theta(s_i)$. Figure 2-(a) shows a contour having many small segments in the fine model which approximate the contour curving by small straight segments. These straight segments appear as horizontal segments in the angle-length graph as shown in Figure 2-(b). The correspondence between the contour (Figure 2-(a)) region and that of the fine model (Figure 2-(b)) is marked with letters. Variation in scale directly affects the segment lengths when the distance to object changes. The exterior angles between the segments do not depend on the initial planar orientation of the object. This model is invariant versus changes in position and orientation of the original object.

The starting segment depends on initial orientation of contour and on the starting point in the contour-following. Changing the initial orientation produces fine models that differ in their starting segments. Each of these models can be obtained from any other through a number of rotate-shift operations over the ordered sequence of segments.

For given starting segments of fine models F_A and F_B of objects A and B, a distance denoted by $d(F_A, F_B)$ can be defined as the *sum of the area difference* along the common length of F_A and F_B in the angle-length graph. The distance is defined by $d(F_A, F_B) = \sum_{\Delta\rho} |\theta_A(\Delta\rho) - \theta_B(\Delta\rho)|\Delta\rho$, where $\Delta\rho$ is the largest interval of contiguous lengths over which both $\theta_A(\Delta\rho)$ and $\theta_B(\Delta\rho)$ are constant. In other terms, models F_A and F_B generally have different number of segments with different lengths, thus the area difference $d(F_A, F_B)$ must be evaluated over the union of length intervals of F_A and F_B.

The distance is meant to be the least value of all possible sum of area difference that is the minimum value of $d(F_A, F_B)$ versus all possible horizontal shifts of F_A with respect

to F_B. The minimum area difference is an excellent metric to quantify the degree of similarity between two shapes. It provides a metric that is useful for *shape matching*. To find the minimum value [11] of $d(F_A, F_B)$ we find a value of initial angle that minimizes a quadratic objective function on the distance.

The use of this metric in a recognition system faces the problem of linearly searching all the models that any recognition system is to avoid. Therefore, the distance matching $d(F_A, F_B)$ becomes useful only when enough evidences have been accumulated on potential matching of the scene object F_A and a small set of library models. In summary, brute force evaluation of $d(F_A, F_B)$ is completely inefficient but the use of segment matching information alleviate the need for the shift operations and makes this distance matching very useful in later matching decisions.

2.4 Coarse model

Shape matching is based on the use of local geometric features which are initially too fragmented in the the fine model. Too simple features may occur in many models. The features should contain enough discriminatory information in order to provide efficient and accurate indexing of candidate library models. Too complex features have two drawbacks: 1) cannot be observed in partial contours, and 2) lead to linear search across the database. We therefore need local features that contain enough discriminatory information.

We need to exploit the benefit of library model in decomposing contours into constantly curved segments. A sequence of segments that corresponds to a contour having constant curvature can be represented by one single segment. Successive straight segments, having similar ratios of angular change over length, are merged to form one super segment. For example, the coarse segments shown on the angle-length graph of Figure 2-(c). Formally, the coarse model C results from clustering the segments of the fine polar model F. Though the fine model has 55 straight segments (Figure 2-(b)) the clustering produces a stable coarse model with only 16 segments (Figure 2-(c)).

In general, a coarse segment S_k is characterized by three parameters that are: 1) the initial angle $\theta_{init}(S_k)$, 2) the total angular change $\Delta\theta_e(S_k)$, and 3) the segment length ρ_k. The initial angle $\theta_{init}(S_k)$ is the exterior angle between segments S_{k-1} and S_k that is the turning angle from S_{k-1}, or its tangent if S_{k-1} is curved, and S_k or its tangent if S_k is curved. The total angular change $\Delta\theta_e(S_k)$ is the turning angle from the tangent to S_k at its start point to the tangent to S_k at its end point. Formally, the coarse model C is an approximation of the original contour by means of an ordered set of constantly curved segments, i.e. $C = \{S_k = (\theta_{init}(S_k), \Delta\theta_e(S_k), \rho_k)\}$.

The coarse model provides gross modeling of the contour shape (sketch) regardless of the object scale or initial position and orientation. Our approach is to extract structural geometric features out of ordered set of coarse segments which will prove to be essential to reduce the complexity of the recognition process.

2.5 Designing shape descriptors

The efficiency of the matching depends to a large extent on the scalability [1, 5, 6] of the recognition operator which is the ability to recognize whole contours as well as fragments of contours while reducing the combinatorics of the search. For this, the extracted features [7] must be local and small enough to match wherever they are present but must also be stable and discriminative. Global features are inadequate when contours are partially

observed. To avoid the linear search problem, different approaches have been proposed in the literature which use some level of abstraction in modeling objects as a collection of local features which are used in pruning unreasonable matches prior to attempting the accurate matching over a small number of potential objects. This strategy greatly reduces the matching complexity.

The features must be: 1) stable enough with respect to digitization, 2) invariant versus changes in the position and orientation, and 3) scale invariant to some degree. By combining the features we can define a library of *descriptors*. There are two objectives behind the design of descriptors which are: 1) clustering of the library information in a discriminative manner, and 2) enabling the search of library objects using partial information. We present two experiments for setting of descriptors: (1) *descriptor set A* (DS-A), and (2) *descriptor set B* (DS-B).

In DS-A, we present a set of eight local and global descriptors which are shown in Table 1. The descriptors can be classified as *straight segment descriptors* and *curved segment descriptors*.

In DS-B, a set of four descriptors (see Table 2) are defined on the basis of referring the current coarse segment S_i with respect to its previous segment S_{i-1}. To increase discriminability, four categories of segment connections are used to distinguish between different possible combination of straight and curved segments. For example, in type straight-curved (s-c) the current segment S_i is straight and the previous segment S_{i-1} is curved. The four possible types of the two-segment descriptors are: 1) straight-straight (s-s), 2) straight-curved (s-c), 3) curved-straight (c-s), and curved-curved (c-c). The geometric representation of each of these types is shown on Figure 3. The length ρ_i of S_i is used with reference to the sum of segment lengths $\rho_i + \rho_{i-1}$, the relative length is $\rho_i/(\rho_i + \rho_{i-1})$. When the current segment S_i is curved the total angular change $\Delta\alpha_i$ that S_i undergoes from its start to its end is also used. Another important point is that in DS-B the descriptors have *multi-dimensional index* that include the type, the relative length, exterior angle, and eventually the total angular change.

Note that in both experiments the selected descriptors are invariant with respect to change in object position and orientation. Some descriptors from DS-A like the number of segments (D_1) and length of longest segment (D_2) are global descriptors. These can be used only for entirely observed objects. On the other hand, the length used in DS-A is scale variant because its values change with changes in the scale of the shape. This is the case of D_2, D_4, and D_8.

2.6 Building the library model

To maximize sharing of the library model, supervised learning is applied for defining classes so that each class consists of a set of objects whose descriptors are nearly identical.

Formally, let $D(O_i)$ and $D(O_j)$ be the set of descriptors associated to objects O_i and O_j, respectively. These objects will be clustered into the same class if for every descriptor D_k we have $|D_k(O_i) - D_k(O_j)| \leq \Delta D_k$, where ΔD_k is an accepted tolerance for D_k. Due to digitization noise some tolerance must be established on the value of descriptors to ensure reliable indexing. The range of values for D_k allows finding a reasonable upper bound on the tolerance ΔD_k. The bound should be tight enough to preserve the descriptor *discriminating information* while allowing some *degree of sharing* among similar shapes. The classes result from partitioning the objects based on their coarse description.

Given the classes and their corresponding descriptors, we define a set of indexed tables $\{Inx_k\}$ each is associated to a distinct descriptor D_k. Table Inx_k results from sorting all the classes according to the decreasing order of the kth descriptor values D_k of all the classes. Inx_k is used as an indexed table which means that the table is built so that its entry key is the value of D_k and its output is a set of classes for which the value of the kth descriptor is the key. In other terms, $\{Inx_k\}$ is the set of all classes sorted so that the first class is one with the highest value of D_k, the second class has next highest value of D_k, etc.

A heuristic search function H consists of finding all the library classes which share a given value of one specific descriptor. Denote by $D_k(O)$ the value of the kth descriptor for some object O, then a class C will be selected by H if the kth descriptor $D_k(C)$ matches the value of $D_k(O)$ within some allowed tolerance. In other terms, we must have $|D_k(O) - D_k(C)| \leq \Delta D_k$. Therefore, the heuristic decision $H(D_k(O), \Delta D_k)$ allows finding a cluster (CL_k) of classes so that each class in this cluster shares with O the value of the kth descriptor, i.e. $C \in CL_k$ implies $|D_k(O) - D_k(C)| \leq \Delta D_k$.

The storage of library model is as follows: 1) the set of library classes, 2) the descriptors of each class, 3) the fine model of each library object, and 4) the set of indexed tables for the library. More specifically, each model contour must be associated a set of descriptors that results from pre-processing of the scene image as shown in Figure 4-(a). It is shown that the operators used are: 1) edge detection, 2) direction coding, 3) segmentation, 4) clustering, and 5) feature extraction which produces a set of descriptors associated to the scene contour. The set of descriptors are used to build the indexed tables which together with the fine and coarse models represent the library models. This process is summarized in the flowchart shown in Figure 4-(a).

3 The recognition system

We present a recognition method based on the use of structural indexing as the main strategy to avoid linear search of the models. The detail of the recognition system will be presented in a gradual manner starting with recognition of closed contours to complex recognition of partially occluded scenes.

3.1 Recognizing closed contours

The heuristic search for closed contours is based on local and global descriptor values which enables efficient indexing of the library. This allows pruning large portions of the models in early steps of recognition. The strategy used consists of three steps: 1) pre-processing, 2) pre-recognition, and 3) recognition. In the following we explain these steps.

Pre-processing of the scene image consists of applying low level vision operators to obtain the model of scene contours. This is shown in the the dashed frame of Figure 4-(a) in which a solid box represent a processing function and input or output are indicated inside simple boxes. Pre-processing consists of obtaining the fine and coarse models as well as the descriptor values as shown in Figure 4-(b). Each descriptor values (D) allows finding a matching cluster (CL) of classes so that each class in CL admits the value of D as part of its descriptor set. Note that descriptors from DS-A and DS-B can be used in this phase. The next step consists of selecting a small percentage of candidate classes

which received the highest vote among all matched clusters. The matched classes are sorted in decreasing order of frequency.

Finally, we evaluate the distance matching between the scene object and each of the matched classes. Starting with the class having the highest matching frequency (C), the distance matching ($D(O_x, C)$) between scene object O_x and model C is evaluated. The first model for which the value of distance matching is below some threshold is hypothesized. The algorithm fails when none of the hypothesized models can be matched with the scene object.

The use of local and global features considerably reduces the combinatorics of the search. Unfortunately, this strategy does not apply for partially occluded scenes in which only local features are observed. In the next section we present recognition of partially occluded scenes.

3.2 Recognition of partially occluded scenes

Figure 5 shows one possible partial occlusion among two *cutters* and one *allen wrench* from the library shown in Figure 6. Figure 6 shows two classes of objects. The scene contains six fragments of contours (A_i), of which one contour is recognized as an entirely observed contour (A_1) and the remaining five fragments (A_2, A_3, A_4, A_5, and A_6) are partially observed contours.

Some fragments like A_4 and A_5 carry poor information, while other fragments (A_2) carry rich geometric information. A recognition algorithm is to exploit the amount of available information on each fragment starting with rich fragments that contains more discriminative information. Poor fragments can help in consolidating and validating overall interpretation. The ambiguity due to three intersecting contours is examined by the gradient direction and resolved by assuming continuity of the contour in the neighborhood of the intersecting pixel. This connects two contours and the third is left as open.

The objective is to find a global interpretation of the scene in which each fragment is mapped (matched) to known library object. The mapping must be coherent that is the inter-relationships among the fragments in the scene must be similar to the inter-relationships among the matched fragments in the models. Our heuristic search consists of indexing local features to find a set of classes so that each fragment is mapped by its local geometric shape to a small set of library fragments.

Denote by $A = \{A_i : i = 1, \ldots, t\}$ a set of t open fragment of contours. The problem is to relate the descriptors $D(A_i)$ associated to fragment A_i to those of a candid class C_k.

Descriptors D_1, D_2, and D_5 from DS-A cannot be used in the search because these are *global descriptors*. The heuristic search requires the use of *local descriptors* such as D_3, D_4, D_6, D_7, and D_8. We also note that descriptors D_4 and D_8 are scale variant. Therefore, recognizing partially occluded scenes with the descriptors defined in DS-A require the descriptors be *local* and *scale invariant*. This means that the descriptors that can be used from DS-A must be limited to D_3, D_6, and D_7.

On the other hand, the descriptors defined in DS-B (Table 2) do not require any global contour information. The reason is that the descriptor used refers to the following information: 1) the type of the two joining segments, 2) the ratio of successive segments length, 3) the exterior angle between successive segments, and 3) the total angular change for curved segments. All these features are *local* with respect to the contour as well as *scale invariant* (relative length). Therefore, recognizing partially occluded scenes can be

done by using all the descriptors defined in DS-B.

The heuristic search is applied to the descriptors of each fragment A_i for finding its cluster $CL_i = \{(C, F) : D(A_i) \in C\}$, where C is a class that shares F values of the descriptors $D(A_i)$. Finding the clusters of matched classes is the first step of recognizing partially occluded scenes as depicted in the *pre-recognition* part of the flowchart shown in Figure 7. At this point, only some classes of CL_i contains the geometric order of the segments of A_i. Since each cluster initially contains many classes, therefore, the need to reduce each matched cluster to those classes that contains the entire shape of the fragment. This is the objective of the next section.

3.2.1 Cluster reduction

The reduction process consists of selecting a subset of classes from each matched cluster such that each selected class contains the geometric shape of at least one fragment A_i. To test the matching between A_i and some $C_k \in CL_i$, the information on the identity of the matched descriptors between A_i and C_k is used together with the distance measure that has been defined in Section 2.3. A class $C_k \in CL_i$ whose distance $d(A_i, C_k)$ to A_i is small enough is added to a reduced cluster CL_i^* to state that the geometric shape of A_i is present in the description C_k.

Finding the reduced clusters of matched classes is the second step as depicted in the *cluster reduction* part of the flowchart shown in Figure 7. It evaluates the distance $D(A_i, C_k)$ along the segments of A_i. If the distance exceeds some normalized threshold ϵ_f, then evaluation of $D(A_i, c_k)$ is abandoned and evaluation of the distance is started for the next class C_{k+1}. This allows finding unique or multiple matching solutions because a fragment of contour can be matched more than once in a given matched class. The output of the cluster reduction step is a set of reduced clusters that contains the matched classes sorted in the decreasing order of the descriptor matching frequencies which is the output of *cluster reduction* shown in Figure 7.

3.2.2 Partitioning the fragments into classes

Using the reduced clusters, grouping all the fragments that belong to the same class allows generation of hypotheses on the potential mapping of fragments to classes. A group $g_k\{A_i\}$ consists of all the scene fragments whose geometric shape is present in class C_k. Assume $A_i, A_j \in g_k$. One can find out whether the relative position and orientation of A_j with respect to A_i in the scene is identical or similar to that of the matched segments within class C_k. If the above geometric relationships are similar, then the pairing of (A_i, A_j) to class C_k is valid because C_k may contains fragments A_i and A_j as they are positioned in the scene.

Assume that A_i and A_j have previously been matched to model C_k. Let x_1, x_2, and x_3 be any non co-linear vertices of some three segments which are denoted by (s_1, s_2, s_3) of A_i. Denote by A_i^* and A_j^* the contours of C_k that are matched to A_i and A_j, respectively. As each segment of A_i is matched to some segment ${A_i}^*$, then consider the points x_1^*, x_2^*, and x_3^* that are the vertices of the segments of ${A_i}^*$ which are matched to s_1, s_2, and s_3, respectively. Similarly, let y_1, y_2, and y_3 be any non co-linear points that are the vertices of some segments denoted by c_1, c_2, and c_3 of A_j. Now, let y_1^*, y_2^*, and y_3^* be the vertices of the segments of A_j^* that are matched to c_1, c_2, and c_3, respectively.

We use a relative distance error $d(A_j\text{-}A_i/scene, A_j^*\text{-}A_i^*/C_k)$ between to relative position and orientation of A_j^*/A_i^* in scene and those of their matched segments. The distance is intuitively defined by:

$$d(A_j - A_i/scene, A_j^* - A_i^*/C_k) = \sum_k^3 \sum_l^3 \frac{|d(x_k, y_l) - d(x_k^*, y_l^*)|}{Min\{d(x_k, y_l), d(x_k^*, y_l^*)\}}$$

where $Min(.,.)$ is used to normalize the distance error. The pairing (A_i, A_j) is valid with respect to class C_k whenever $d(A_j - A_i/scene, A_j^* - A_i^*/C_k) \leq \epsilon_d$, where ϵ_d is some normalized threshold. In this case, the relation $R_k(A_i, A_j)$ is established. The output of *geometric matching* shown in in Figure 7 is a set of $\{g_k^*\{A_i\}\}$ of matched groups.

3.2.3 Interpretation

Generally, a fragment can be member of more than one group which indicates the need for some selection criterion to find valid partitions such as favoring the largest group of fragments. In this case, the group with highest cardinality is selected first. Among all the remaining groups, it only keeps the groups to which it belongs at least one fragment that is not covered by the previously selected groups. Note that the selected set of fragments are not necessarily disjoint due to potential overlapping.

Assume five fragments (A_1, A_2, A_3, A_4, and A_5) have been matched into four groups $g_1^*(A_1, A_3, A_4)$, $g_2^*(A_1, A_2, A_3)$, $g_3^*(A_2, A_3, A_5)$, and $g_4^*(A_1, A_3, A_4, A_5)$. Selection of the largest group g_4^*, leads to select either groups g_2^* or g_3^* because the uncovered fragment A_2 may belong to g_2^*, g_3^*, or both. Therefore, two initial solutions are present: $sol_1 = \{g_2^*, g_4^*\}$ and $sol_2 = \{g_3^*, g_4^*\}$. Note that fragments (A_1, A_3) and (A_3, A_5) overlap in sol_1 and sol_2. By discarding g_4^*, the fragments are still all covered by the remaining groups, other solutions can then be found. Fragment A_4, is now only covered in ${g_1}^*$. One solution is $sol_3 = \{g_1^*, g_3^*\}$ because A_5 is only covered in g_3^* and the union of g_1^* and g_3^* covers all fragments.

For the example shown in Figure 5, A_1 is matched to *cutter 3* which is shown on Figure 6-g. Fragment A_2 is matched to the *cutter class* and based on geometric matching A_2 was matched to *cutter 4*. Fragments A_4 and A_5 were matched to a large number of classes. Geometric matching of pairs (A_2, A_4) and (A_2, A_5) in the scene and in model (such as in *cutter 4*) failed. A_2 has been hypothesized for a much smaller set of models compared to those of fragments A_4 and A_5. Our algorithm processes first the fragments that have less number of matched models to avoid excessive processing overhead.

Early in the recognition, the class of *allen wrench* has been hypothesized for fragments A_3 and A_6 among few other classes. The geometric matching of (A_2, A_3) or (A_2, A_6) was not evaluated because the *cutter class* was not hypothesized to fragments A_3 and A_6. The geometric matching was evaluated only for the pairs (A_3, A_4), (A_3, A_5), (A_6, A_4), (A_6, A_5), and (A_3, A_6). The relative positioning of the above pairs were found to be quite similar to that of their matched contours in *allen wrench 2* and *3* (Figures 6-(b) and -(c)) with slight advantage to *allen wrench 2* that is the correct solution. Interpretation of the scene gives the solution: $g^*_{cutter-3}(A_1)$, $g^*_{cutter-4}(A_2)$, and $g^*_{allen-wrench-2}(A_3, A_6, A_4, A_5)$.

4 Performance evaluation

Evaluation of the proposed indexing scheme for image modeling and recognition is carried out by: 1) discussing the features of the proposed system, 2) evaluating the search method and its time, and 3) comparing to others.

4.1 Features of the method

In this sub-section we present some important features [12] of the proposed modeling and recognition system. The features refers to the *generality*, *stability*, *robustness*, and *discriminative power* of this approach.

The *generality* of modeling and recognition system is concerned with the generality of the class of contour shapes that can be successfully recognized in the majority of cases. Our approach is based on partitioning contours into set of constantly curved segments, extracting descriptors from these coarse segments, and using of the descriptors in pruning the models. All polygonal shapes can easily be used in this method as their only effect is to produce simple descriptors. Objects with curved contours having moderate change in curving are the most adequate for our modeling because these shapes produce moderate number of coarse segments and consequently require reasonable processing time during modeling and recognition. Shapes having large number of inflection points are likely to cause significant increase of the processing time for this approach as well as for many other proposed modeling and recognition. In our case, the advantage of partitioning contours into set of constantly curved segments is clear when compared to simple contour segmentation [12, 13, 3]. Our modeling approach has better contour fitting, produces much less number of coarse segments, and makes our recognition applicable to a larger class of real objects.

The *stability* of contour modeling provides information on how invariant the resulting model is in the presence of variations in scale, noise, and quantization. We have proposed a two-level segmentation approach for modeling curved contours by means of constantly curved segments as a strategy to obtain stable model of the sketch of the original object. In the first segmentation level, a noisy contour is partitioned into many small straight segments (fine model). In the second segmentation level, successive straight segments are merged into super coarse segments which preserves the contour shape because the secondary effects of noisy break-point are discarded. This process reduces the effects of noise and digitization on the output model because of its two-level filtering.

The *robustness* of a recognition system measures its ability to handle real object contours under variations in the rotation, translation, scale, and view-point. Our approach is based on partitioning contours into set of constantly curved coarse-segments and use of their descriptors in the recognition process. The descriptors in DS-A are invariant under rotation and translation but also scale variant. The geometric parameters used as descriptors in DS-B have coarser granularity and are invariant with respect to rotation, translation, and scale. It was observed that the recognition can still give a reasonable (80%) rate of successful classification even when with up to 25° change in view-point than the model. Modeling contours by using constantly curved segments is more robust than simple polygonal approximation for representing contours of real objects. However, complex industrial objects that have arbitrary shapes with arbitrary number of inflection points may cause degradation in the processing time. Therefore, the main effect of in-

creasing the complexity of object shapes is an increase of the model size and the implied increase in the recognition time.

The *discriminative power* [12] of the proposed modeling and recognition refers to the cost of generating the matched classes (hypotheses) in the pre-recognition phase and the cost of carrying out the verification through cluster reduction, geometric matching, and interpretation. We use similar notation to that of [12]. We assume the descriptors are uniformly distributed over the entries of the indexed tables. The number of descriptors for each table entry is d, the number of descriptors for the scene is n, and the number of scene fragments is n_f.

When there is significant discrimination power the number of descriptor values that fall into each table entry is small and possibly equal to 1 ($d = 1$). In this case, each cluster corresponds to one single model. There are n/n_f descriptors per fragments and each descriptor is matched to the correct model in the best case. In this condition there is no need for cluster reduction, geometric matching, and interpretation because of the one-to-one correspondence between the descriptor values and models. The cost of obtaining the matching cluster for each fragment is $O(n/n_f)$. The complexity of the best case is $O_{best}(n) = n_f \times O(n/n_f) = O(n)$. We note that the complexity of the best case in our approach is identical to that of [12].

In the worst case, when the discrimination power is small the number of descriptor values that fall into each table entry is large. In this case each descriptor cluster (out of n clusters) may contains large portion of the library models which means that each table entry has m models in the worst case, where m is the number of library models. At the early stages of recognition, the number of correct matches represent a very small fraction of total number of matches. Our approach keeps only few matched models (fixed number). If we assume the descriptor values are uniformly distributed over the n_f scene fragments, then each fragment has n/n_f matched models following pre-recognition. Each fragment has n/n_f clusters and each contains m models in the worst case. For each fragment, the complexity of sorting the descriptor clusters and selecting a fraction of models with the highest frequencies is $O((mn/n_f)^2) = O(m^2n^2/n_f^2)$ which is needed to obtain the single fragment cluster. For all the fragment clusters the complexity is $n_f \times O(m^2n^2/n_f^2) = O(m^2n^2/n_f)$. The complexity of cluster reduction for each fragments is $O(mn/n_f)$ because it requires evaluation of the distance matching for each matched model. There are n_f scene fragments, the complexity of cluster reduction for the scene is $O(mn)$. The geometric matching requires comparing the positioning of pair of fragments in scene to positioning of matched fragments in the model. The complexity of geometric matching is $O(mn_f^2)$ because in the worst case each scene fragment must be geometrically matched to each other fragment. Finally, the interpretation phase has complexity $O(mn_f^2)$. Overall worst case complexity is $O_{worst}(n) = O(n^2m^2/n_f + mn + mn_f^2)$. The real algorithm complexity ($O_{alg}(n)$) is somehow bounded as $O_{best}(n) \leq O_{alg} \leq O_{worst}(n)$ that is $O(n) \leq O_{alg} \leq O(n^2m^2/n_f + mn + mn_f^2)$.

4.2 Sub-linear recognition time

We study the effects of increasing the library size on the recognition and classification of scenes of three objects under partial occluding. Each of the studied objects has between 10 to 30 coarse segments. All thresholds used were experimentally evaluated. Four settings of the model are used: 1) 10-objects (LB_{10}), 2) 30-objects (LB_{30}), 3) 60-objects (LB_{60}),

and 4) 100-objects (LB_{100}). We started by setting up LB_{10} and randomly selecting 3 scene objects. To build LB_{30} we randomly selected 20 more objects and added them to LB_{10}, and so on. All studied objects are *lab mechanical tools* with different sizes and different shapes having between 20 to 140 fine segments or between 10 to 50 coarse segments.

The recognition algorithm is run under each of the database settings for recognizing the scenes like that shown on Figure 5. Table 4 shows the number of fine ($F\text{-}Seg$) and coarse ($C\text{-}Seg$) segments and number of hypotheses generated for each fragment (Figure 5) versus the size of the models. The reason for the large number of hypotheses is that each segment is likely to be matched to many objects due to cluster tolerance.

The descriptors of DS-A are fine grain compared to those of DS-B which explains why DS-A received more hypotheses. As a result the selectivity of descriptors from DS-B is greater than that of the descriptors from DS-A. Too small segments like A_4 and A_5 were omitted from hypothesis generation. The number of hypotheses grows linearly with the number of segments and the size of the models for both experiments. The recognition process would be completely inefficient if all of these hypotheses must be verified further. Fortunately, the voting technique enables pruning most inconsistent hypotheses. The fraction of the recognition time spent in the pruning process is small, thus the recognition time is likely to be independent from the number of hypotheses.

The ranking of hypotheses represents the upper percentage of hypotheses that includes all correct hypotheses. For example, DS-A with LB_{10} the algorithm retained 11.8% (see Table 3) of the hypotheses. This was sufficient to ensure that all correct interpretations are retained by the pruning technique. This percentage becomes smaller and smaller as the library size grows. This is shown in Table 3. We may keep only 4% or 5% of total number of hypotheses for DS-A and LB_{100}. These percentages depend on shape complexity, degree of similarity among the models, and amount of variations in spatial layout. By experimentally choosing these percentages the number of *retained hypotheses* becomes constant regardless of library size. Thus the recognition algorithm mainly depend on the scene complexity without explicit dependence on the library size.

The recognition time (Table 3) slightly increases with the library size. However, the descriptors used in DS-B have better selectivity and discrimination power than those of DS-A, thus the saving on the recognition time. The recognition time is sub-linear versus library size which is an indicator of efficiency for the proposed approach.

In Figure 8 we show the efficiency of the recognition system when using the *descriptor set B* versus the percentage of visible segments of known models. Here the efficiency refers to the *percentage of cases for which the correct model was among the set of hypothesized models*. These results are obtained for LB_{100} and the models used in the recognition are *lab mechanical tools* having between 30 and 60 fine segments. Each plotted point results from averaging at least 30 recognition cases. We also repeat the recognition for 4 view-point angles. Here, a 0° view-point angle corresponds to vertical direction. For entirely visible models (all segments are visible), a reasonably high efficiency (80%) is obtained (Figure 8) when the change of view-point is below 25°. However, excessive change in the view-point angle significantly affects the efficiency. The efficiency smoothly decreases but remains above 60% when the percentage of visible segments decreases to 50% and the view-point does not exceed 25°.

4.3 Comparison to other approaches

In [12] contours are partitioned into straight segments and aggregation of few segments is used for defining a "super-segment" that is used for indexing the library. Similar modeling were also used in [3, 13]. In our case we have proposed fine grained descriptors in *descriptor set A* and showed the benefit of increasing the granularity of the descriptors, making them structural (assembly of neighboring geometric features) and scale invariant. This was proposed in *descriptor set B*.

Grimson [9, 10] used tree-searching schemes that equally treats all the available features in generating hypotheses on possible matches. The search over the current sub-tree is abandoned when enough inconsistent evidences are accumulated and the next sub-tree is started. Though this organization allows pruning many inconsistent sub-tree interpretations, the number of visited sub-trees can be large even for simple scenes. In [3] all scene features participate in the generation of hypotheses that are ranked by mutual support. This consists of reporting the matched models into the scene and collecting supporting evidences whenever they map to similar locations. Due to the large number of initially generated hypotheses, a massively parallel machine (Connection Machine CM5) is used to parallelize the complexity of scene and each processor carry out verification of one hypothesis. The recognition time mainly depends on scene complexity. As a result of huge parallelism the recognition time of few objects is about that of recognizing one object.

Our approach avoids handling contours with poor information. It consists of selecting contours with the largest number of features among all scene contours which allows pruning large portions of the models and provides robust generation of hypotheses. In feature hypothesizing, we select only the most probable hypotheses which greatly reduces the recognition time and improves its independency on the size of the models. In fragment hypothesizing, the ordering of the matched scene features of each fragment of contour is searched in the hypothesized models which allows finding a small number of consistent hypotheses.

The novelty of this approach resides in the search operator hierarchy in which the first operators are timely inexpensive but have large number of operands. Each operator contributes in reducing the search space, thus leaving less number of verifications which necessarily become more timely expensive because of their higher conceptual level.

5 Conclusion

In this paper we presented the design and implementation of a model-based pattern recognition system that is invariant with respect to rotation, translation, and scale. The proposed recognition strategy extracts some geometric descriptors from the sketch of the object and use this information for pruning large portions of the models. It progressively applies finer matching to refine the recognition. Our scheme is based on four steps. A fine and coarse models of the scene object are used. The coarse model is used to extract descriptor attributes. Indexed search over the library is carried out for pruning inconsistent matching out of an initially large set of hypotheses. Geometric relationships and distance matching are used to consolidate the filtered decisions. The result of our library organization and our pruning strategy is that we can produce correct classification with reasonable probability. The recognition time mainly depends on scene complexity with only marginal dependence on library size. This proves the effectiveness of our proposed approach in modeling and recognizing mechanical tools under partially occluded scenes.

6 Acknowledgment

Thanks to Mr. Limalia Ismail for implementing the low level vision processing system as part of his M.S. Thesis at the Computer Engineering Department, College of Computer Science and Engineering, King Fahd University of Petroleum and Minerals. The author acknowledges support from the Research Committee and the College of Computer Science and Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia.

References

[1] G. J. Ettinger. Large hierarchical object recognition using libraries of parametrized model sub-parts. *Proc of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 32–41, 1988.

[2] X. Chin and C. R. Dyer. Model-based recognition in robot vision. *ACM Computing Surveys*, 18, No. 1:67–108, 1986.

[3] L. W. Tucker, C. R. Feynman, and D. M. Fritzsche. Object recognition using the Connection Machine. *Proc of the IEEE Conf. on Computer Vision and Pattern Recognitiopn*, pages 871–878, Jun 1988.

[4] N. R. Corby. Machine vision for robotics. *IEEE Trans. on Industrial Electronics*, 30, No 3:282–291, Aug 1983.

[5] W. E. L. Grimson and D. P. Huttenlocher. On the verification of hypothesized matches in model-based recognition. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 13, No 12:1201–1213, Dec 1991.

[6] A. Califano and R. Mohan. Multidimensional indexing for recognizing visual shapes. *Proc of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 28–34, Jun 1991.

[7] N. Ayache and O. D. Faugeras. HYPER: A new approach for recognition and positioning of two dimensional objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 1:44–54, Jan 1986.

[8] X. Bolles and R. A. Cain. Recognizing and locating partially visible objects: the local-features-focus method. *Inter. J. of Robotics Research*, 1, No. 3:57–82, 1982.

[9] W. E. L. Grimson. On the recognition of curved objects. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 11, No 6:632–642, Jun 1989.

[10] W. E. L. Grimson. The combinatorics of heuristic search termination for object recognition in cluttered environments. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 13, No 9:920–935, Sep 1991.

[11] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An efficient computable metric for comparing polygonal shapes. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 13, No 3:209–216, Mar 1991.

[12] F. Stein and G. Medioni. Structural indexing: efficient 2-D object recognition. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 14, No 12:1198–1204, Dec 1992.

[13] P. W. M. Tsang and P. C. Yuen. Recognizing of partially occluded objects. *IEEE Trans. on Systems, Man, and Cybernatics*, 23, No 1:228–236, Jan-Feb 1993.

Type	Descriptor	interpretation
straight	D_1	Number of segments
	D_2	Length of the longest segment
	D_3	Collection of exterior angles $\{\theta_i : \|\theta\| \geq \epsilon_\theta\}$
	D_4	Collection of segment lengths $\{\rho_i \geq \epsilon_\rho\}$
curved	D_5	Number of segments
	D_6	Collection of curvature factors $\{H_i\}$
	D_7	Collection of total angular changes $\{\Delta\alpha_i\}$
	D_8	Collection of segment lengths $\{\rho_i\}$

Table 1: Topological and geometric features of *descriptor set A*

Type	Feature 1	Feature 2	Feature 3	Descriptor
straight/ straight	Length $\frac{\rho_i}{\rho_i+\rho_{i-1}}$	exterior angle θ_i	NA	$< s-s, \frac{\rho_i}{\rho_i+\rho_{i-1}}, \theta_i >$
straight/ curved	Length $\frac{\rho_i}{\rho_i+\rho_{i-1}}$	exterior angle θ_i	NA	$< s-c, \frac{\rho_i}{\rho_i+\rho_{i-1}}, \theta_i >$
curved/ straight	Length $\frac{\rho_i}{\rho_i+\rho_{i-1}}$	Exterior angle θ_i	Total angular change $\Delta\alpha_i$	$< c-s, \frac{\rho_i}{\rho_i+\rho_{i-1}}, \theta_i, \Delta\alpha_i >$
curved/ curved	Length $\frac{\rho_i}{\rho_i+\rho_{i-1}}$	Exterior angle θ_i	Total angular change $\Delta\alpha_i$	$< c-c, \frac{\rho_i}{\rho_i+\rho_{i-1}}, \theta_i, \Delta\alpha_i >$

Table 2: Structured and scale invariant features of *descriptor set B*

	DESCRIPTOR SET A				DESCRIPTOR SET B			
	LB_{10}	LB_{30}	LB_{60}	LB_{100}	LB_{10}	LB_{30}	LB_{60}	LB_{100}
Percentage of correct matches	11.8	7.3	5.4	3.9	8.2	4.3	3.1	2.1
Recognition time (seconds)	27.4	29.7	32.4	35.1	21.4	23.7	25.6	26.8

Table 3: Ranking of correct hypotheses and recognition time for descriptor sets A and B

SCENE			DESCRIPTOR SET A				DESCRIPTOR SET B			
Fragment	F-Seg	C-Seg	LB_{10}	LB_{30}	LB_{60}	LB_{100}	LB_{10}	LB_{30}	LB_{60}	LB_{100}
A_1	118	14	72	241	344	538	38	133	161	189
A_2	96	12	55	163	233	417	27	87	117	135
A_3	33	6	36	114	171	312	15	41	62	68
A_6	10	5	27	76	104	193	11	28	38	52
Total	257	37	190	353	852	1460	91	289	378	444

Table 4: Hypothesis generation for each fragment for descriptor sets A and B

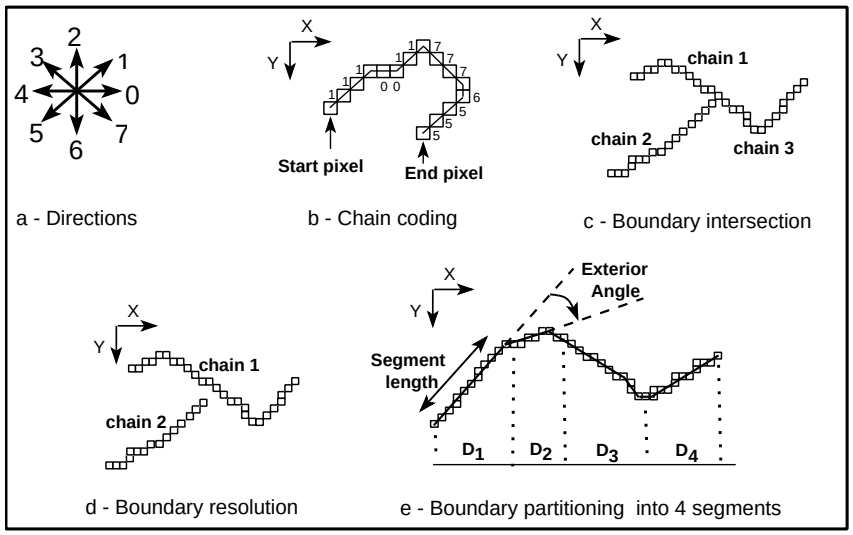

Figure 1: (a) Directions, (b) chain coding, (c) intersection, (d) resolution, and (e) segmentation

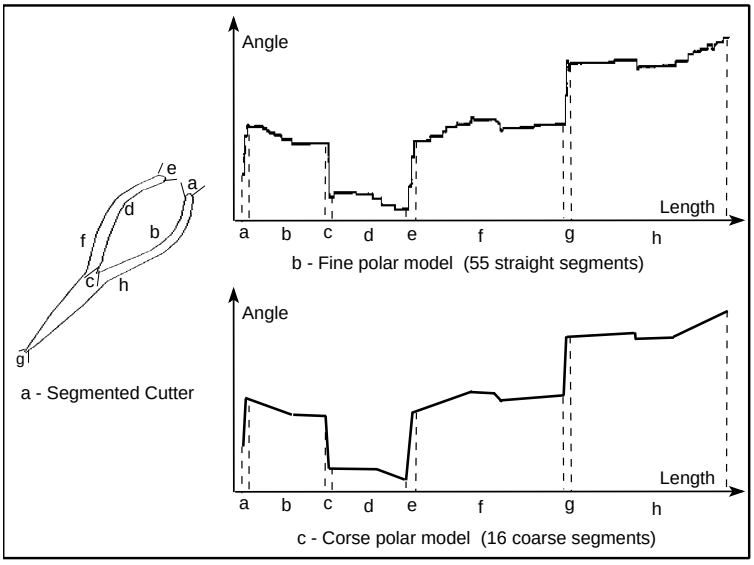

Figure 2: Segmented cutter (a) with its fine model (b) and its coarse model (c)

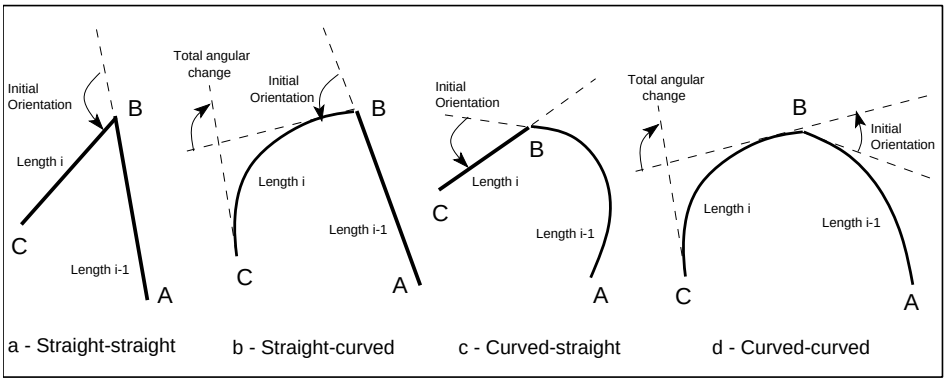

Figure 3: Types of geometric descriptors: (a) s-s, (b) s-c, (c) c-s, and (d) c-c

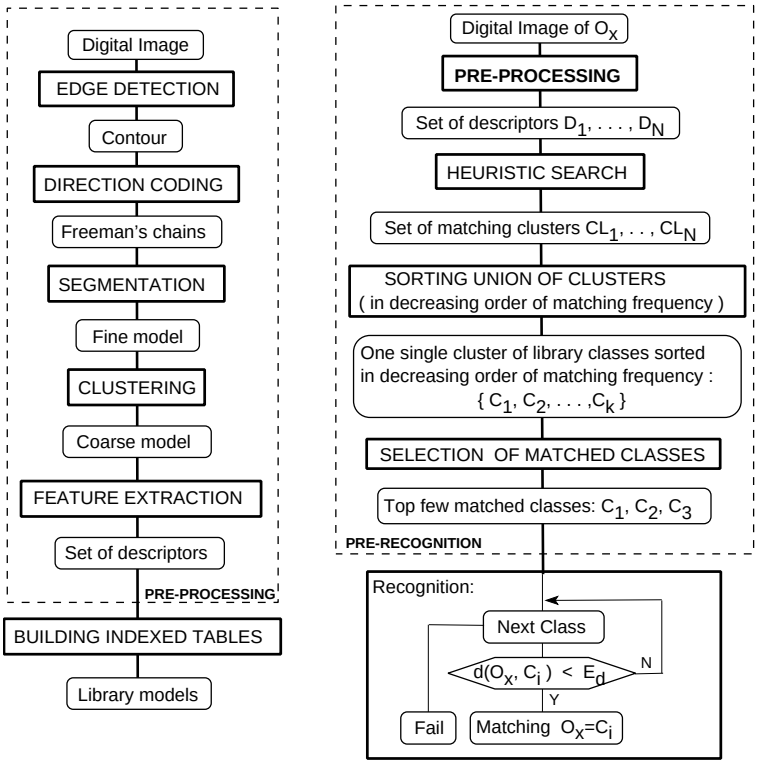

Figure 4: Flowchart of library model and recognition of closed contours

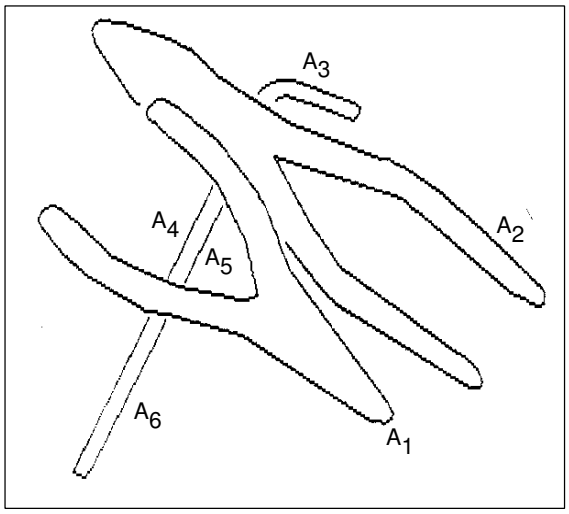

Figure 5: A scene with partial occlusion among three objects

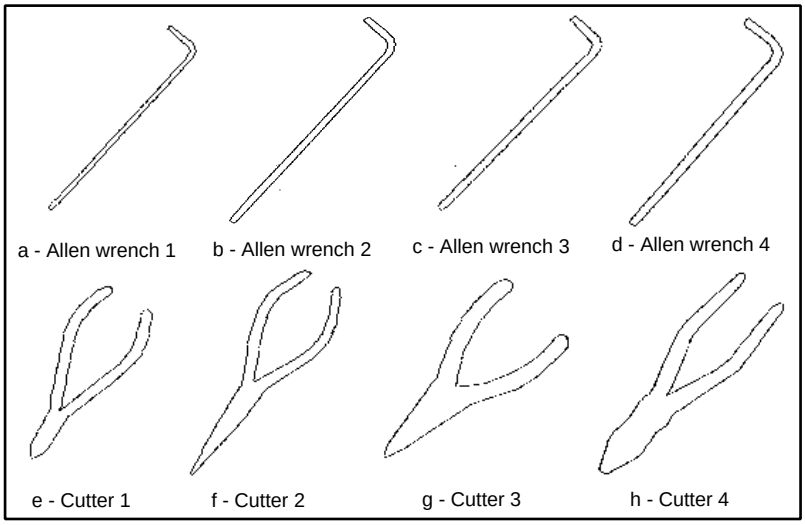

Figure 6: Two classes of library objects (Allen Wrench and Cutter)

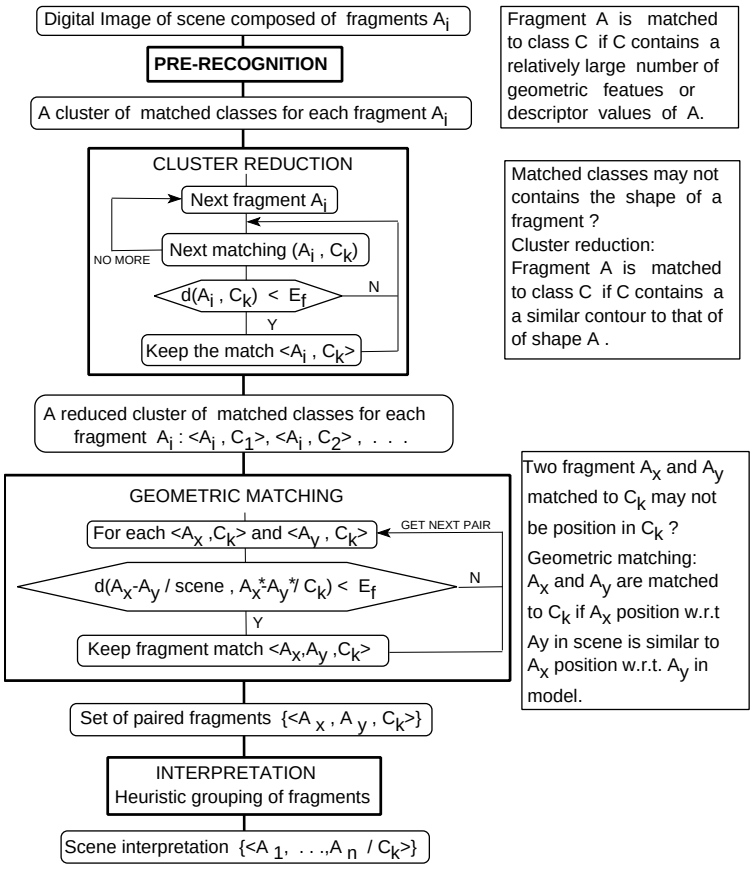

Figure 7: Flowchart of recognition of partially occluded scenes

Figure 8: Efficiency of recognition system (descriptor set B) versus the percentage of visible segments for few view-angles