Chapter 6

Occlusion Detection Using a Multiple Evidence-based Approach and Object Representation

6.1 Introduction

One of the major problems in computer vision involves dealing with uncertain information. Occlusion, dissimilar views, insufficient illumination, insufficient resolution, and degradation give rise to imprecise data. At the same time, incomplete or local knowledge of a scene gives rise to imprecise interpretation rules. Automatically solving the occlusion problem is a challenge, especially when little is known about the world.

It is natural to build a computer vision system that incorporates uncertainty reasoning, which is a collection of inference techniques for reasoning with uncertain information. To cope with real world problems, one must deal with uncertainty, which may appear in the following forms [69,85]:

1. The information is partial.
2. The information is not fully reliable.
3. The information from different resources may conflict.

The major methods of statistical reasoning approaches are Bayesian reasoning, MYCIN-style reasoning, Dempster-Shafer (D-S) reasoning, and fuzzy logic. The main disadvantage of the Bayesian reasoning is that prior probabilities must be assigned, which is difficult to perform in this application [69]. The certainty factor (CF) was successful in diagnosing diseases in MYCIN-style reasoning and is good for a singleton hypothesis space. One weak point in the certainty factor mode is that it is an ad hoc technique and therefore lacks mathematical foundation. Since the D-S combination rule includes the Bayesian and certainty factor, combining functions as special cases, the same disadvantages discussed above are true. In the fuzzy logic method, a major problem exists when a rule provides very low confidence and many other rules have high values. Final confidence values computed using fuzzy intersection operations are then completely determined by the lowest confidence value, even if many other rules have good credibility. This method seems to be inconsistent with human decision making.

This work describes an approach for automatically extracting representations of three-dimensional (3D) solid objects from a single range image. The objects of interest are polyhedra, for which some of the surfaces may be partially occluded. After an initial image segmentation procedure, tentative 3D vertex locations for the objects are obtained through an analysis of two-dimensional (2D) region shape and corner proximity.

A novel procedure then utilizes geometric constraints and robust density estimation to associate each vertex with appropriate faces of each object. A major contribution of this work is the analysis of occlusion, which provides the means by which vertices are associated with (and disassociated from) particular object faces. Finally, vertex locations are refined through a global optimization step. Because the system described here goes beyond image segmentation to perform object extraction, this research differs considerably from previous work that involves the analysis of range images. The approach is model-driven only in the sense that objects are assumed to be polyhedral; there is no recognition process, and no library of known shapes to guide the extraction process.

6.2 Vertex Substitution Algorithm

The 2D corner locations detected in Section 4.4 are mapped to 3D locations that represent the initial estimates of object vertices. To map from 2D to 3D, the system uses calibration information and range values that are present in the range image.

Unfortunately, because of sensor noise and inaccuracies in the segmentation and corner-detection processes, some of these initial estimates of vertex locations do not lie near the true object vertices. Accuracy suffers especially near occluding boundaries. For this reason, additional analysis is needed to improve the initial estimates. A first improvement comes from a process that termed *vertex substitution*. Because each vertex has already been associated with one or more object faces, it is relatively simple to map initial vertex estimates onto the planes of best fit through the respective object faces. Vertex substitution is performed only for initial vertex estimates that are initially associated with two or more object faces.

Using the results of vertex substitution, it is possible to apply a geometry-driven analysis to determine whether occlusion is present for faces that correspond to adjacent regions in the original range image. Knowledge of the presence or absence of occlusion is crucial to the object extraction and vertex refinement processes. To this end, a multiple evidence-based approach is used to consider pairs of object faces in turn, testing each pair for evidence of occlusion.

This approach is illustrated in Figure 6.1, which contains labeled regions and a line drawing that represents a box and a pyramid as shown in Figures 6.1(a) and (b). Following the approach described here, five regions are detected in the range image, represented as s_1-s_5. The corner-detection and -merging procedure locates 12 corners designated as v_1-$v_{12,}$ as shown in Figure 6.1(c). The merging procedure retains knowledge of which vertices are associated with which regions. For example, vertex v_9 is associated with the three regions (s_3, s_4, s_5) and vertex v_3 is associated with the three regions (s_1, s_3, s_5). However, vertex v_2 in this example is associated only with region s_1, because no corner is detected in close proximity for neighboring region s_3.

For the reasons given above, initial vertex estimates often lie far from the true vertex locations. This is illustrated in Figure 6.1(d), where "*" represent some of the initial 3D estimates that map directly from 2D corner detection. The purpose of "vertex substitution" is to identify 3D *interior points* that lie on each face. These are illustrated using "+" in the figure.

These interior points are needed because the subsequent geometric analysis assumes that representative points are available from actual object faces. To see the importance of this, consider vertex v_3, which has been associated with the faces s_1, s_3, and s_5.

Assigning the interior points is done according to perpendicular distance and surface normal agreement criteria for each point of the window surrounding the initial vertex location and its associated face. The vertex v_3 is associated with the faces s_1, s_3, and s_5, as described previously. Clearly, the association with all three faces is not correct. By

deriving three separate interior points to replace v_3, as shown in Figure 6.1(d), the occlusion that is present can be detected more readily.

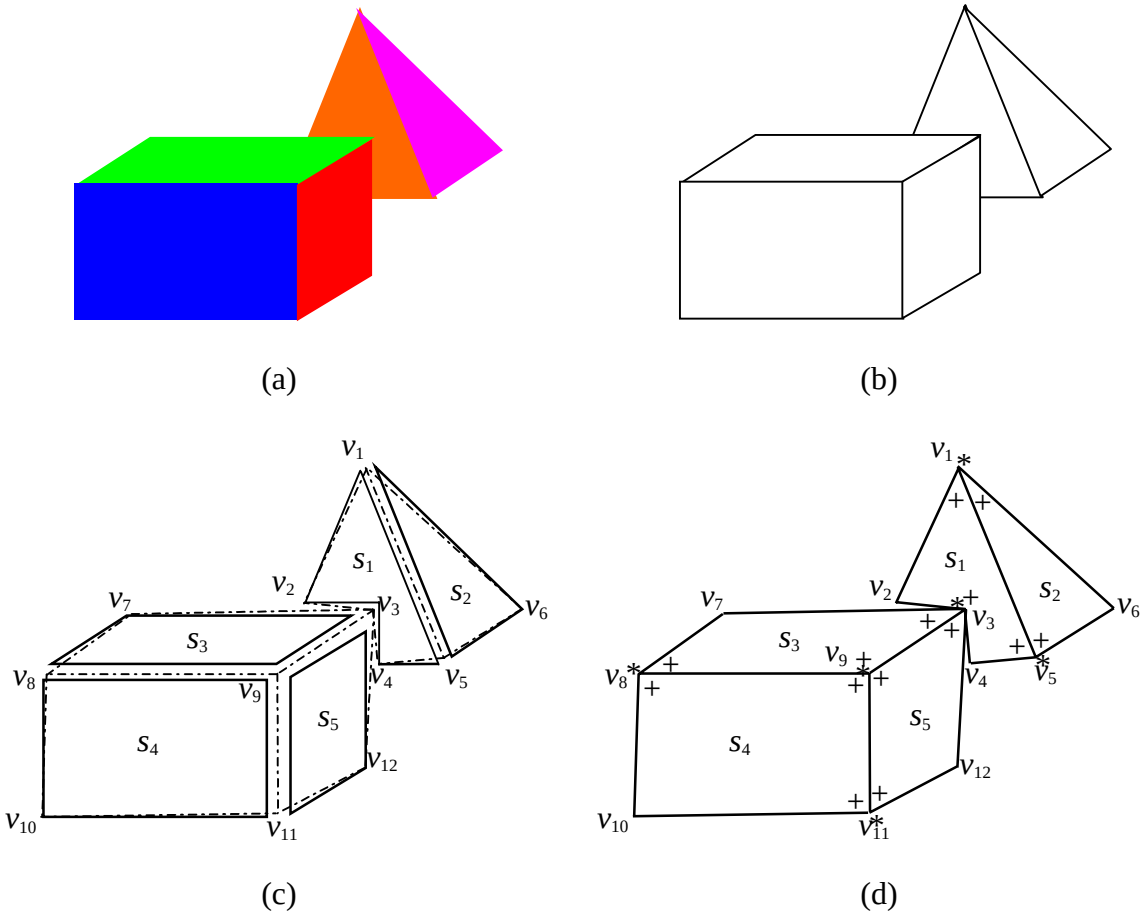


Figure 6.1 Illustration of vertex merging, assignment to faces, and vertex substitution. Solid and dashed lines represent the face borders in isolation from all others and the object borders after corner merging respectively. (a) Labeled faces. (b) Line drawing of these regions. (c) Vertex merging and assignment to faces. (d) Vertex substitution with interior points.

This algorithm is applied for each initial estimate vertex that is associated with more than one face and is driven as follows:

1. Set window size to 3×3 around the initial estimate 2D vertex location.
2. Repeat the following steps until all adjacent faces associated to the current vertex are assigned interior points:
   i. For each point in the window surrounding the current vertex location, compute:
      - The perpendicular distances from that point to the planes that contain the pair of adjacent faces.
      - The angle between surface normal vectors of these planes and of that point under consideration, as computed from a neighborhood in the range image.
      - If the two values computed above are within threshold values for more than one point in each face of the adjacent pair then,

   ii. Use a robust statistical method to select the two points belonging to this current pair of adjacent faces, considering one point for each face.

   iii. Assign these two points as interior points for this pair of adjacent faces.

   iv. If no interior point for any of the faces can be found, increase window size by 1 and repeat steps i through iii.

The accuracy of computing interior points is an important issue because much of the decision of occlusion is based on these points. A robust statistical method is used to select the two most optimal points for each pair of adjacent faces according to the distances between these points. This statistical method is called α-trim algorithm [45].

6.3 Geometric Rules Used for Occlusion Detection

6.3.1 Overview

This section describes five rules that are used in occlusion detection between two adjacent faces. These rules are in the form of "if-then" statements, with MYCIN-style certainty factors associated with each rule [69]. The rules are implemented as a set of functions that return occlusion confidence values based upon their analyses of the data, much in the same way as Selfridge's demons [47].

The system uses geometry and topology primitives to detect occlusion. Examples of these primitives are 3D vertex locations, edge and surface parameters, mean and variance for region's depth values, adjacency relationship between regions, jump edge magnitude, and a histogram of depth values between two adjacent faces. In addition to these primitives, the system uses the Euclidean and perpendicular distance from point to point or from point to line.

In this work, multiple rules were needed because no single rule was adequate for detecting occlusion alone. This is because of real-world problems such as noise and quantization effects. Also, in general, multiple sources of evidence can provide more reliable information than a single source. Each rule has been tested individually to detect occlusion between adjacent faces, and it has been found experimentally that each rule not only fails at least one time to detect an occlusion, but also is guilty of false detection. This is shown in the experimental results of Chapter 7.

Occlusion can be defined mathematically as a discontinuity in a smoothed function of a surface. The goal of occlusion detection is to determine whether a significant discontinuity (within the discrete domain) is present for adjacent regions. The rules given in the next section are assumed to cover all possible cases of two planes of faces in 3D that have the following geometrical relationships:

1. The planes supporting these faces are coplanar.
2. The planes supporting these faces are parallel.
3. The planes supporting these faces are intersecting.

The rules interpret local information at each vertex for the geometrical topology for any pair of adjacent faces associated with a certain vertex. The rules have been design to be independent as possible by using different information sources and heuristics. Based on the results given in the next chapter, it is concluded that no pair of rules is mutually correlated.

Each rule has associated with it a certainty factor, which is a measure of the extent to which the evidence that is described by the antecedent of the rule supports the conclusion that is given in the rule's consequence [69]. A certainty factor (*CF* [*h*, *e*]) is defined in terms of two components:

- *MB* [*h*, *e*] – a measure (between 0 and 1) of belief in hypothesis *h* (occlusion) given the evidence *e* (rule). *MB* measures the extent to which the evidence supports the hypothesis. It is zero if the evidence fails to support the hypothesis.
- *MD* [*h*, *e*] – a measure (between 0 and 1) of disbelief in hypothesis *h* (occlusion) given the evidence *e* (rule). *MB* measures the extent to which the evidence supports the negation of the hypothesis. It is zero if the evidence support s the hypothesis.

From these two measures, the certainty factor can be defined as

$$CF[h,e] = MB[h,e] - MD[h,e] \tag{6.1}$$

In this work, it is assumed that all rules support only measure of belief (*MB)* and not measure of disbelief (*MD = 0*). All rules provide evidence that relates to a single hypothesis. This hypothesis *h* is the existence of occlusion between two adjacent faces. The next step is to combine the measures of belief for the rules. If the combined confidence is high, an occlusion is detected at this pair of adjacent faces.

6.3.2 Geometric Rules for Occlusion Analysis

This section describes five rules that are used in occlusion detection between two adjacent faces.

Rule #1 (Jump edge between two adjacent regions)

IF

(1) Two adjacent faces (s_i, s_j) are tentatively associated with a vertex, AND

(2) The magnitude of the edge $\hat{E}_{ij}$ that exists between the two adjacent faces is large indicating jump edge,

THEN

There is suggestive evidence that occlusion exists between these two adjacent faces.

Rule #2 (Euclidean distance between two interior points of the two adjacent faces)

IF

(1) Two adjacent faces (s_i, s_j) are tentatively associated with a vertex, AND

(2) The Euclidean (3D) distance $d_{p_i p_j}$ between the interior points p_i and p_j of the two adjacent faces is large,

THEN

There is suggestive evidence that occlusion exists between these two adjacent faces.

Rule #3 (3D distance between each interior point and line of intersection of the faces)

IF

(1) Two adjacent faces (s_i, s_j) are tentatively associated with a vertex, AND

(2) The perpendicular (3D) distances ($d_{p_i l_{s_i s_j}}$ and $d_{p_j l_{s_i s_j}}$) between either interior point p_i or p_j of the two adjacent faces, and line of intersection $\vec{l}_{s_i s_j}$ of the planes of the adjacent faces is large,

THEN

There is suggestive evidence that occlusion exists between these two adjacent faces.

Rule #4 (Check modality of the histogram for two adjacent regions)

IF

(1) Two adjacent faces (s_i, s_j) are tentatively associated with a vertex, AND

(2) The histogram of the depth values of the two adjacent faces has two significant modes and a valley v between these two modes has a small value,

THEN

There is suggestive evidence that occlusion exists between these two adjacent faces.

Rule #5 (Perform the T-test for difference of depth values of the two adjacent faces)

IF

(1) Two adjacent faces (s_i, s_j) are tentatively associated with a vertex, AND

(2) The differences in variance and mean of the depth values of the two adjacent faces t_{obs} computed using the T-test is large.

THEN

There is suggestive evidence that occlusion exists between these two adjacent faces.

To illustrate the importance of these rules, Figure 6.2 shows an example that involves occlusion. Figure 6.2(b) presents the segmented image, with labels 1-11 representing separate regions. Using the multiple evidence-based analysis, the measures of belief are obtained for each rule and for each pair of adjacent faces. Occlusion analysis determines that occlusion is present at three of the vertices, labeled *A*, *B*, and *C*. Table 6.1 illustrates the fact that occluded pairs of faces result in large values for the certainty factor (described in greater detail in Section 6.3.5). For example, occluding faces (s_1, s_5) have a certainty factor of CF=0.81, whereas the non-occluded pair (s_3, s_5) have a certainty factor of CF=0.15.

Rule #1 presents the existence of jump edge between two adjacent faces associated with a certain vertex. The magnitude of the estimator $\hat{E}_{ij}$ for the edge E_{ij} that exists between the two adjacent faces is large, indicating jump edge. Examples of the measures of

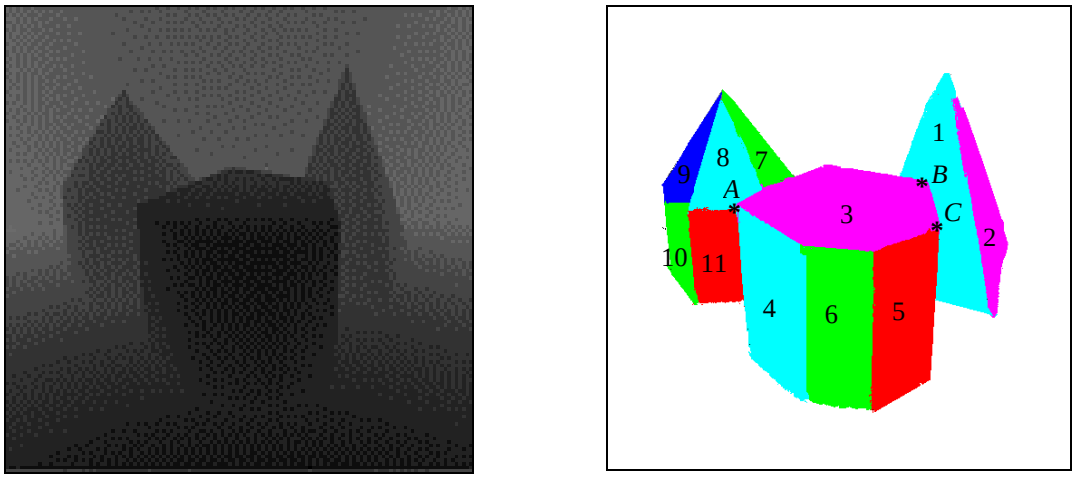
belief generated by this rule for the adjacent pairs of faces {(1,3), (1,5), (3,7), (3,8), (4,11)} are the large values {0.83, 0.97, 0.84, 0.72, 0.85} as shown in Table 6.1.

In Rule #2, the Euclidean distance is measured between two interior points of the pair of adjacent faces as in equation (6.3). Table 6.1 shows an example of the measures of belief generated by Rule #2 for the adjacent pairs of faces {(1,3), (1,5), (3,7), (3,8), (4,11)} that are {0.73, 0.13, 0.59, 0.51, 0.56}. This rule returned a small value of the measure of belief for the pair (1,5), where $MB2 = 0.13$.

Rule #3 measures the 3D distance between each interior point of the two adjacent faces to the line of intersection of the planes containing these faces. Examples of the measures of belief generated by this rule for the adjacent pairs of faces {(1,3), (1,5), (3,7), (3,8), (4,11)} are the large values {0.76, 0.79, 0.19, 0.58, 0.61} as shown in Table 6.1. Tthis rule returned a small value for pair (3,7), where $MB3 = 0.19$.

In Rule #4, the histogram of the depth values for the two adjacent faces in the range image is constructed and checked for bimodality. The multiple modes in the histogram indicate occlusion between the two adjacent faces. In the case of a bimodal histogram, the vallyness is measured; the measure of belief is basically dependent on the vallyness value. Table 6.1 shows an example of the measures of belief generated by Rule #4 for the adjacent pairs of faces {(1,3), (1,5), (3,7), (3,8), (4,11)} that are {0.71, 0.99, 0.94, 0.74, 0.00}. This rule returned a small value of the measure of belief for the pair (4,11), where $MB4 = 0.00$.

In Rule #5, a T-test is made for the difference of the local average of the range values of the adjacent faces. Examples of the measures of belief generated by this rule for the adjacent pairs of faces {(1,3), (1,5), (3,7), (3,8), (4,11)} are the large values {0.77, 0.98, 0.53, 0.62, 0.87}.



(a) (b)

Figure 6.2 Inter-object occlusion example. (a) Range image. (b) The extracted objects without special consideration of partially occluded faces. Vertices *A*, *B*, and *C* are considered the critical cases for occlusion detection in this example.

Table 6.1 A representative sample of the measures of belief (*MBs*) for some cases of the adjacent faces for the objects in Figure 6.3(b). These *MBs* are combined using the average weighted method to obtain the certainty factor *CF*. The highlighted rows indicate the cases of occlusion.

Face s_i	face s_j	*MB*1	*MB*2	*MB*3	*MB*4	*MB*5	*CF*
3	7	0.84	0.59	0.19	0.94	0.53	0.67
3	8	0.72	0.51	0.58	0.74	0.62	0.61
4	11	0.85	0.56	0.61	0.00	0.87	0.60
3	5	0.16	0.08	0.12	0.01	0.31	0.15
3	6	0.11	0.03	0.04	0.01	0.42	0.14
5	6	0.09	0.06	0.12	0.01	0.20	0.10
4	6	0.14	0.15	0.12	0.01	0.47	0.19
1	3	0.83	0.73	0.76	0.71	0.77	0.76
1	5	0.97	0.13	0.79	0.99	0.98	0.81
1	2	0.46	0.08	0.14	0.01	0.25	0.21

6.3.3 Mathematical Basis for the Rules

This section presents the mathematical background for the rules. Figure 6.3 shows the geometry of two adjacent faces s_i and s_j with surface normals $\vec{n}_i = (a_i, b_i, c_i)$ and $\vec{n}_j = (a_j, b_j, c_j)$. The adjacent faces have two calculated interior points, $p_i = (x_i, y_i, z_i)$ and $p_j = (x_j, y_j, z_j)$, near the initial estimate $p_0' = (x_0', y_0', z_0')$ of the vertex that shares the two faces.

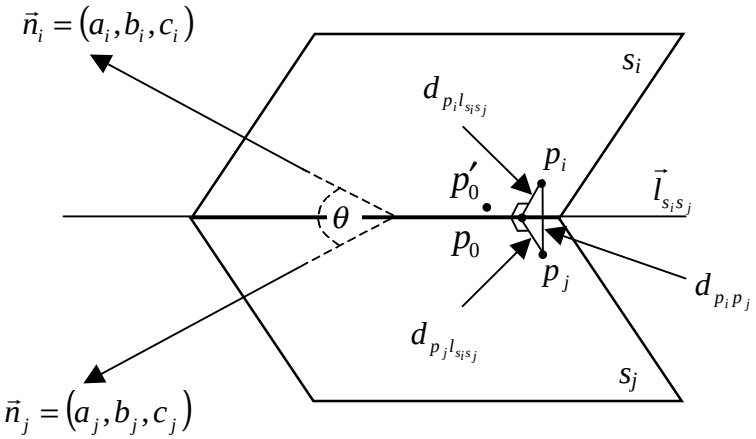


Figure 6.3 The geometric arrangement for two adjacent faces s_i and s_j and the interior points in each face. The rules depend on the features (distances) that are illustrated.

For Rule #1, depth differences for points on or near the boundaries of adjacent regions are computed within window *w* as follows:

$$E_{ij}(x, y) = \max_{w} \{ |z(x, y) - z(x + k, y + l)| \} \tag{6.2}$$

The statistical α-trim algorithm is used to evaluate the estimator $\hat{E}_{ij}$ of all points that lie on jump edges E_{ij}.

In Rule #2, the Euclidean distance between the two interior points p_i and p_j is computed using

$$d_{P_i P_j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{6.3}$$

The symbol $\vec{l}_{s_i s_j}$ represents the line of intersection for the planes of the two adjacent faces s_i and s_j. The parametric form of this line is $\vec{l}_{s_i s_j} = [(x_0 + at), (y_0 + bt), (z_0 + ct)]^{\mathrm{T}}$, where $p_0 = (x_0, y_0, z_0)$ is a point on the line that can be obtained by projecting initial estimate $p_0' = (x_0', y_0', z_0')$. The line lies in the direction (a, b, c). The parameters (a, b, c, d) of the line $\vec{l}_{s_i s_j}$ are calculated using

$$\vec{l}_{s_i s_j} = \vec{n}_i \times \vec{n}_j = \begin{vmatrix} u_1 & u_2 & u_3 \\ a_i & b_i & c_i \\ a_j & b_j & c_j \end{vmatrix} = (b_i c_j - c_i b_j) u_1 - (a_i c_j - c_i a_j) u_2 + (a_i b_j - b_i a_j) u_3 \tag{6.4}$$

where u_1, u_2, and u_3 are the direction components of the line, and $d = -ax_0' - by_0' - cz_0'$. Also, the point $p_0 = (x_0, y_0, z_0)$ can be computed as follows:

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} a_i & b_i & c_i \\ a_j & b_j & c_j \\ a & b & c \end{bmatrix}^{-1} \begin{bmatrix} -d_i \\ -d_j \\ -d \end{bmatrix} \tag{6.5}$$

The parameter t of the line $\vec{l}_{s_i s_j}$ is computed at the interior point $p_j = (x_j, y_j, z_j)$ as

$$t = \frac{a(x_j - x_0) + b(y_j - y_0) + c(z_j - z_0)}{a^2 + b^2 + c^2} \tag{6.6}$$

Then, the perpendicular distance $d_{P_j l_{s_i s_j}}$ between the point p_j and the line $\vec{l}_{s_i s_j}$ is determined using

$$d_{p_j l_{s_i s_j}} = \sqrt{(x_j - x_0 - a\,t)^2 + (y_j - y_0 - b\,t)^2 + (z_j - z_0 - c\,t)^2} \tag{6.7}$$

For Rule #5, the sample mean, $\overline{R}_{s_i}$, and the sample variance, $\sigma_{s_i}^2$, of the range values of face s_i are computed, as are the sample mean, $\overline{R}_{s_j}$, and the sample variance, $\sigma_{s_j}^2$, of the

range values of face s_j. Let m_i and m_j be the number of pixels in s_i and s_j, respectively. Applying the T-test, the statistic, t_{obs} is

$$t_{obs} = \frac{\overline{R}_{s_i} - \overline{R}_{s_j}}{T_C \sqrt{\frac{1}{m_i} + \frac{1}{m_j}}} \tag{6.8}$$

where

$$T_C^2 = \frac{(m_i - 1)\sigma_{s_i}^2 + (m_j - 1)\sigma_{s_j}^2}{m_i + m_j - 2} \tag{6.9}$$

A large value of t_{obs} suggests the presence of occlusion.

6.3.4 Measures of Belief Generation

The measure of belief (*MB*) for each rule is a real number between 0.0 and 1.0, and its value represents the significance of this rule in detecting the occlusion. The measure of belief follows the sigmoid (or S-shaped) function to produce this real number [69]:

$$f(x) = \frac{1}{1 + e^{-ax}} \tag{6.10}$$

The sigmoid function is continuous and differentiable. The output of this function approaches to 1.0 if *x* is getting large (approaches to +∞) and approaches to 0.0 if *x* is very small (approaches to -∞). While the geometric primitive as distance are taken values from 0 to +∞, so the function in (6.10) is modified to return *MB*s that lie in the range [0, 1] as follows

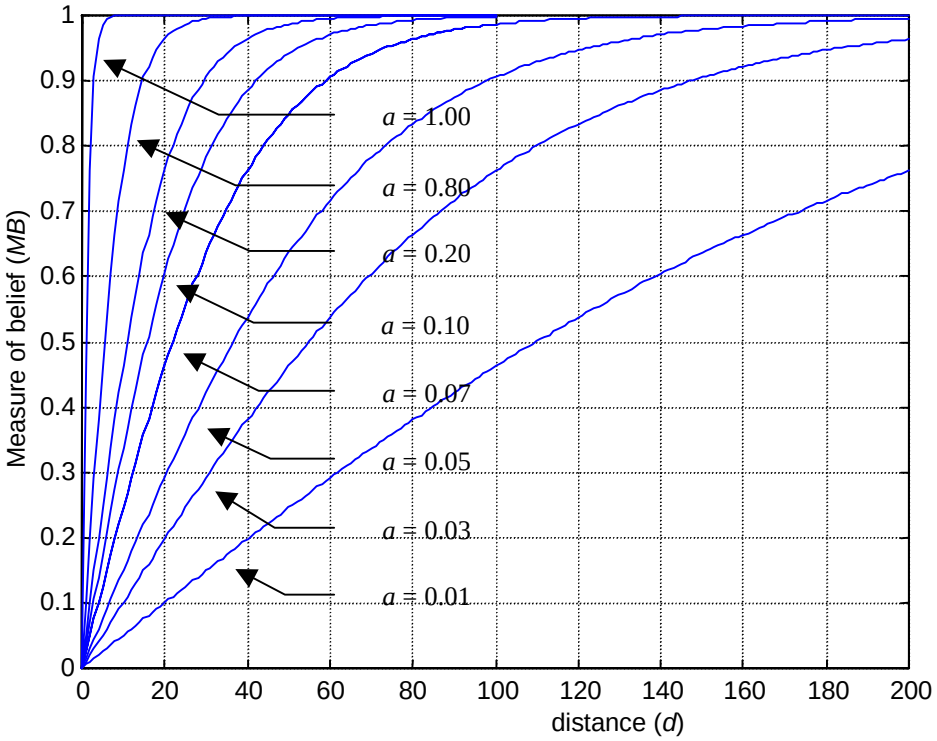
$$MB(d) = \frac{1 - e^{-ad}}{1 + e^{-ad}} \tag{6.11}$$

Figure 6.4 presents an example of the function (6.11) at different constant *a*. This function is applied for Rules #1, #2, #3, and #5.

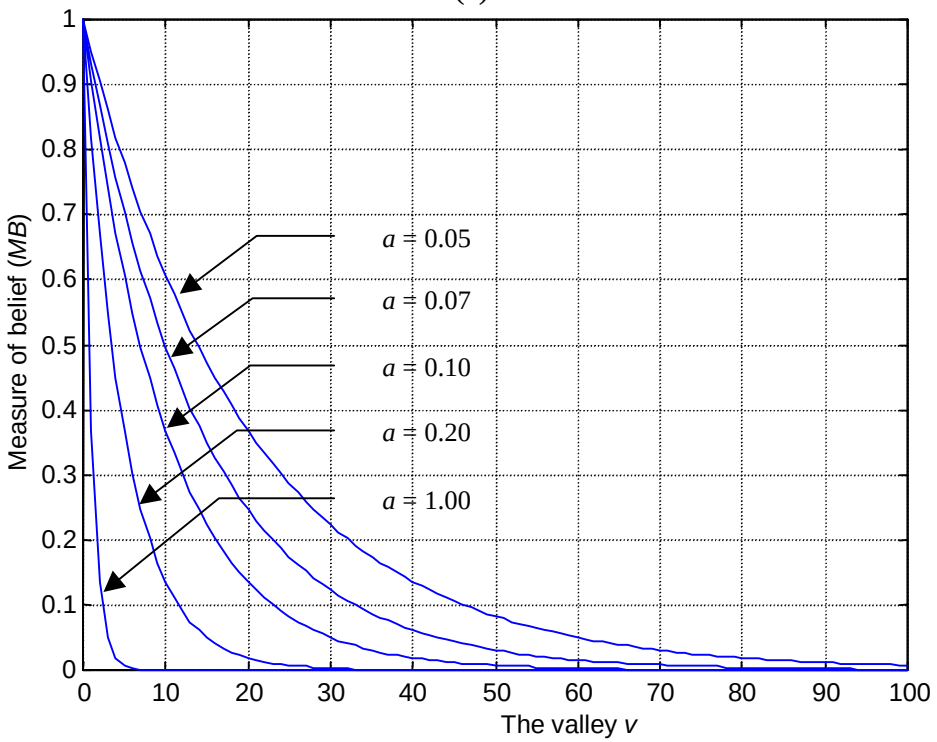
The function in (6.10) is also modified to work with the concept of small valley value *v* of the histogram. The measure of belief is getting large (approaches to 1.0) if the valley value of the histogram is getting small (approaches to 0.0), as shown in Figure 6.4(b). This function is applied for Rule #4.

$$MB(v) = \frac{1}{e^{av}} \tag{6.12}$$

Selecting the parameter *a* is based on the application and the dimension of units as shown in Figure 6.4(a) and (b).



(a)



(b)

Figure 6.4 The sigmoid functions used to produce the measures of belief (*MBs*) for the rules from the geometrical features at different *a*. (a) The sigmoid function used for Rules #1, #2, #3, and #5. (b) The sigmoid function used for Rule #4.

6.3.5 Combining Approaches for the Measures of Belief of the Rules

The measure of beliefs generated for each rule should be combined into a single certainty factor (*CF*) for each pair of faces that are associated with a single vertex as shown in Figure 6.5. The reliability of the combination methods is very important in detecting occlusion. The combining function has the following properties:

- Since the order in which evidence is collected arbitrary, the combining functions should be commutative and associative.
- If uncertain inferences are chained together, then the result should be less certain than either of the inferences alone.
- Until certainty is reached, additional confirming evidence should increase *MB.*

Three methods for the purpose of combining the measures of belief have been considered. The first method is the MYCIN method [69], in which the measure of belief of a hypothesis *h*, given two evidences e_1 and e_2, is computed using

$$MB[h, e_1 \wedge e_2] == \begin{cases} 0 & \text{if } MD[h, e_1 \wedge e_2] = 1 \\ MB[h, e_1] + MB[h, e_2] \cdot (1 - MB[h, e_1]) & \text{otherwise} \end{cases} \qquad (6.13)$$

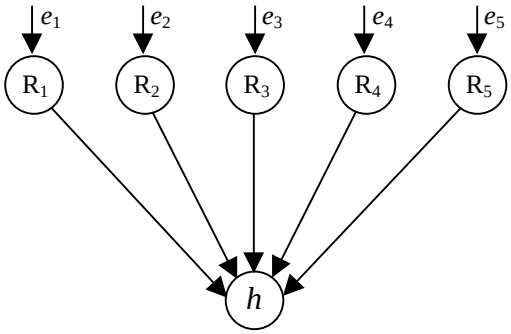


Figure 6.5 Combination of rules evidences.

A problem with the first method is that a single large *MB* can incorrectly cause a large *CF*, even if the remaining *MB*s are very low. To address this problem, a second method was considered in which voting is used to select the *CF*. If a majority of the *MB*s is greater than a threshold value, then the *CF* is computed in the same way as first method; otherwise, it is computed by combining all *MB*s having values less than the threshold value.

The third method uses a weighted average to compute the *CF*. This depends on the confidence of each rule $P(h/e_i)$ that represents the reliability of each rule with respect to occlusion detection as follows:

$$CF(h/e_1,e_2,...,e_5)=\frac{\sum_{i=1}^{5}P(h/e_i)MB[h,e_i]}{\sum_{i=1}^{5}P(h/e_i)} \tag{6.14}$$

This method achieves reliable and good results in terms of ignoring the *MB* outlier (false alarm) and the rate of occlusion detection.

6.4 Occlusion Detection Strategy

This section describes the novel occlusion-detection strategy that has been developed. The basic idea is to consider pairs of adjacent faces that are associated with a single vertex initially extracted in Section 4.4. The initial extracted vertices may be associated with faces that partially occlude each other. The case of more than one face associated to an initial vertex is considered. An approach is used to check all adjacent pairs of faces associated to this vertex to determine if there an occlusion exists between any adjacent pairs. If occlusion is found for a pair of faces, then the occluded face is disassociated from the vertex under consideration. All adjacency relations, occlusion information, and vertex associations are maintained in a lookup table. The decision is taken based on the certainty factor value resultant from combining the measure of beliefs of the rules applied to the adjacent pair. The procedure is driven as follows:

1. Repeat for all vertices of the range image that are initially associated with more than one face:
2. Apply the vertex substitution algorithm to determine interior points on each face.
3. For all adjacent pairs of faces associated with the current vertex:
4. Apply all rules to find the measures of belief for the current pair of faces.

5. Combine these measures of belief using the average-weighted method to obtain the certainty factor of this pair.
6. If the computed certainty factors exceed a threshold value, then an occlusion is assumed between the two faces.

6.5 Grouping Body Faces Into Objects

A fundamental problem in the analysis of images is object isolation. Objects are composed of faces, which are portions of a surface bounded by edges. An image region is a connected part of the projection of a face. . The idea here is to group each set of faces into objects. The advantage of grouping faces into objects is to obtain context information about the object. This information—the moment, area, etc.—could be used in the recognition post-processing and matching with a stored model for these objects.

The classical method of separating overlapped objects is based on line labels and vertex labels [24,85]. Also, the process needs to traverse from one vertex to another, knowing the relationship between each vertex and their neighbors. This technique has some problems in case of concave and complex 3D solid object.

The proposed approach of grouping faces is basically dependent on collecting the adjacent faces that do not have occlusion between them. The region adjacent graph (RAG) has an important role in detecting occlusion between adjacent faces and in grouping adjacent faces that do not have occlusion. The region grouper algorithm begins by removing the adjacent relationship between the occluded adjacent faces in the RAG lookup table. Next, beginning with the first face, it collects faces that are adjacent to this face according to the adjacency relationship in the lookup table of the RAG. After grouping the first object faces, a search of the RAG is performed to check for other faces. If other faces exist, the algorithm takes the first found face in the RAG lookup table and starts to group the next object faces. This process is repeated until all faces in the RAG lookup table are grouped into objects. Faces that are adjacent and shared only in one vertex are considered belonging to two different objects. This assumption overcomes the cases where a mistake occurs in the voting of occlusion between two adjacent faces or to isolate two RAGs that linked together at one vertex.

6.6 Model Creation

6.6.1 Overview

In classical solid modeling, a computer model is a representation that encodes the infinite point set (of a given object) using a finite amount of computer storage. The main task is to produce a smaller data set (in the form of a model) that retains the important features of the data in a manner conducive to answering geometric questions about the object. To some degree, the relative importance of each of these aspects is dependent upon the task to be performed using the model.

For instance, a robot arm moving about a desktop may require only a crude box as a model for a telephone, this being sufficient to avoid collision. In the visualization of the same telephone, however, the shape of the handset is probably vital. In the computation of answers to geometric questions, the model used for collision avoidance would need to support quick and precise responses (to some tolerances), while the model used for visualization may not need to support geometric responses at all. The modeling technique used in this work is based on the boundary representation. The vertices of the 3D object are considered a significant primitive in the model constructed by the boundary representation. A refinement technique is developed to improve the accuracy of these vertices using a global optimization method.

6.6.2 Global Optimization for the 3D Vertex Locations of Polyhedrons

Object representations can be improved through a final refinement in estimated vertex positions. This section considers a global optimization procedure for vertex coordinates, which is especially important near occluding boundaries.

The current estimates of vertex coordinates, as obtained in Section 4.4, are improved using an iterative search procedure that is formulated as a problem of energy minimization. The implementation is an unconstrained, quasi-Newton style optimization procedure. The objective function considers all vertices and visible faces for an object *O* and is based on the robust least-median-of-squares method. The median tolerates up to 50% outliers in a

data set [45]. The estimated optimal location for all visible vertices of object *O* is $\hat{\theta}_O = \left\{ \left[(\hat{x}_{Ok}, \hat{y}_{Ok}, \hat{z}_{Ok}) \right]^T \middle| k = 1, ..., n_O \right\}$, which is obtained as follows:

$$\hat{\theta}_O = \arg\min_{\hat{\theta}_O} \left\{ \underset{q}{\text{med}} \left(\frac{a_{Oj}\hat{x}_{Oq} + b_{Oj}\hat{y}_{Oq} + c_{Oj}\hat{z}_{Oq} + d_{Oj}}{\sqrt{a_{Oj}^2 + b_{Oj}^2 + c_{Oj}^2}} \right)^2 + \underset{k}{\text{med}} \left(\hat{V}_{Ok} - V_{Oik} \right)^2 \right\} \tag{6.15}$$

where $q = 1, ..., \sum_{j=1}^{m_O} u_{Oj}$ and $k = 1, ..., n_O$. The variables $(a_{Oj}, b_{Oj}, c_{Oj}, d_{Oj})$ are the plane parameters of the face *j* that belong to object *O*. Also, $\hat{V}_O$ and V_{Oi} matrices are the optimal and initial estimate of vertex location. They are represented as well as the optimal values $\hat{\theta}_O$ in the image coordinates for object *O* as follows:

$$\hat{\theta}_o = \begin{bmatrix} \hat{x}_{o1} & \hat{y}_{o1} & \hat{z}_{o1} \\ \hat{x}_{o2} & \hat{y}_{o2} & \hat{z}_{o2} \\ \vdots & \vdots & \vdots \\ \hat{x}_{on_o} & \hat{y}_{on_o} & \hat{z}_{on_o} \end{bmatrix}, \ \hat{V}_o = \begin{bmatrix} \hat{r}_{o1} & \hat{c}_{o1} \\ \hat{r}_{o2} & \hat{c}_{o2} \\ \vdots & \vdots \\ \hat{r}_{on_o} & \hat{c}_{on_o} \end{bmatrix}, \ V_{oi} = \begin{bmatrix} r_{oi1} & c_{oi1} \\ r_{oi2} & c_{oi2} \\ \vdots & \vdots \\ r_{oin_o} & c_{oin_o} \end{bmatrix} \tag{6.16}$$

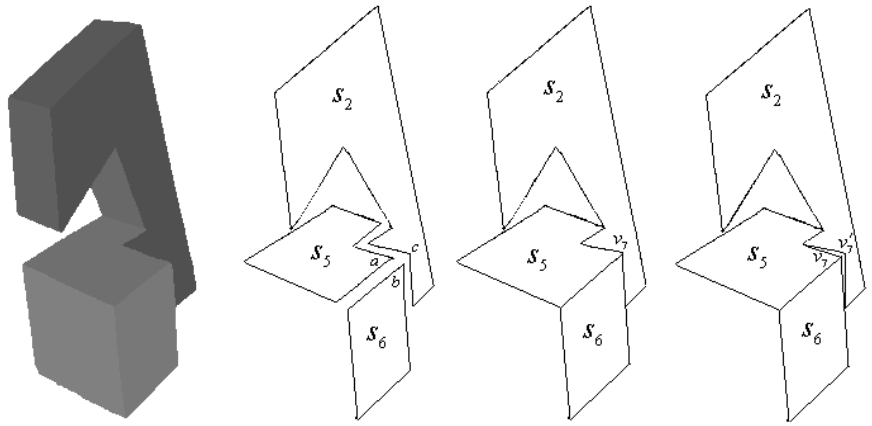
Then, the parameters m_O and n_O represent the number of faces and vertices for object *O* respectively. Finally, u_{Oj} represent the number of vertices of face *j* for object *O*. Determination of the matrix $\hat{V}_o$ is performed in each iteration for the optimization procedure by the inverse transformation of the system of in Section 3.3 using numerical method.

The first term in the objective function represents the perpendicular distances for all vertices to their faces of one object. The second term represents the deviation of the optimized vertex locations from the initial estimate in the image coordinates. The second term also prevents the optimal vertex locations from going far away from the initial estimate.

6.6.3 Example of Vertex Optimization with Self-occluded Object Faces

The vertex refinement method described in Section 6.6.2 works well for objects that do not contain occlusion-related problems. In cases of occlusion, it works well after disassociating the occluded face from the initial vertex.

Consider the object that appears in Figure 6.6(a). One of these faces, labeled s_2 in Figure 6.6(b), is partly occluded by faces s_5 and s_6. Initially, the corners $a \in s_5, b \in s_6$, and $c \in s_2$ will be merged incorrectly into a single 3D vertex v_7 as illustrated in Figure 6.6(c). If the method of the previous section is applied without detecting the occlusion between the faces s_2, s_5, and s_6, the refined vertex resulting from optimization will be incorrect. In order to prevent such a problem, a multiple evidence-based approach is applied at each pair of adjacent faces associated with v_7.



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

Figure 6.6 Examples of an object that exhibits partial self-occlusion of one face, s_2, by two other faces, s_5 and s_6. (a) Segmented object. (b) Contours of regions, s_2, s_5 and s_6, with 3 corners indicated. (c) Incorrect merging of corners to form a single 3D vertex, v_7. (d) Results of geometric reasoning to accommodate partial occlusion. A new point v_7' is introduced, and is used for further optimization.

As a result of occlusion detection, interior points are introduced for the faces s_2, s_5, and s_6. As shown in Figure 6.6(d), the interior point v_7' now lies on the face s_2, which is updated during the subsequent optimization procedure as described in the previous section. Although this point does not correspond to a physical vertex of the object, in this

analysis, it is convenient to treat it as if it does. And although this example illustrates a case of self-occlusion only, the same considerations are effective for inter-object occlusion.

6.6.4 Boundary Representations

Boundary representations represent a solid object by storing a description of its boundary [42, 85]. The boundary of an object divides space into two parts, one having finite volume and the other having infinite volume. If it is assumed that all objects have finite volume, then an object can be represented unambiguously by its boundary. The boundary is often divided into a three-level hierarchy of entities: faces, edges, and vertices.

Representation schemes, which are both unambiguous and unique for a polyhedron, are highly desirable because they are one-to-one mappings from the object space to the representation space. This implies that, in such schemes, distinct representations correspond to distinct objects. In this research, the model for each 3D object is represented as a single region adjacency graph (RAG), sets of surface parameters (a, b, c, d), and sets of vertices (x, y, z).

6.7 Summary

A major contribution of this work is the analysis of occlusion, which provides the means by which vertices are associated with (and disassociated from) particular object faces. Finally, vertex locations are refined through a global optimization step. Significant gains in accuracy are obtained especially when vertices are correctly assigned to object faces that are partially occluded.

The multiple evidence-based approach is used to identify faces that are partly occluded, and vertices are reassigned based on this geometric information. The visible faces of polyhedral objects are represented using region adjacency graphs, with object vertices associated with the appropriate faces.