Open-Ball 피처 추출 방법에 의한 3차원 물체 인식

김 성 수†

요 약

3차원 물체 인식 중 오목과 볼록을 갖고 있는 물체의 인식은 대단히 어려운 문제이다. 본 논문에서는 물체의 인식을 위한 피처(Feature)의 추출 방법으로 오픈-볼(Open-Ball)을 제안한다. 이 새로운 방법은 물체의 크기, 이동과 회전에 불변성을 갖는 피처(Feature)를 생성하는 것뿐만이 아니라, 비교되는 물체를 인식하는 것을 상대적인 닮음 정도 측정으로 구현한다.

3-D Object Recognition Using a Feature Extraction Scheme : Open-Ball Operator

Sung-Soo Kim†

ABSTRACT

Recognition of three-dimensional objects with convexities and concavities is a hard and challenging problem. This paper presents a feature extraction method out of three-dimensional objects for the purpose of classification. This new method not only provides invariance to scale, translation, and rotation R^3 but also distinguishes any three-dimensional model objects with concavities and convexities by measuring a relative similarity in the information space where a set of characteristics features of objects is mapped.

1. Introduction

Three-dimensional object recognition is one of the major tasks in machine and computer vision. Understanding objects with convexities and concavities contained in three-dimensional objects is a very challenging subject. Among the various methods, feature extraction is a popular paradigm which indexes or groups several different pieces of information on two or three-dimensional objects as one of different categories, respectively. In this sense, the method of mapping the information from an object into a unique feature is the key to good recognition and classification. Feature extraction from three-dimensional objects is more difficult than two dimensional patterns due to the complication and the difficulty involved in the observation of three-dimensional objects. In this paper, the open-ball operator (OBO) [1,2] is proposed with a more rigorous approach to the problem involved in concavities and convexities in three-dimensional objects.

In the real world, it is reasonable to assume that

† 정 회 원 : 한국전자통신연구원 선임연구원
논문접수 : 1998년 10월 24일, 심사완료 : 1999년 2월 1일

we can represent three dimensional objects by approximating them using polyhedra to the degree of a necessary precision required in various applications. The OBO, a three dimensional feature extraction method, has a property of invariance on scale, translation, and rotation by extracting feature selectively when objects contain concavities or convexities. The rest of the paper is organized as follows : Section 2 introduces the Open-Ball-Operator (OBO) for three-dimensional feature extraction. Section 3 introduces the structure of classification and discusses its properties. In section 4, experimental results are presented with a discussion on their implication.

2. Open-Ball-Opera.or

When we approximate three dimensional objects by polyhedra with irregular spatial sampling in R^3, the objects represented by a set of vertices which are spatially sampled points of the continuous surface of a three-dimensional object. The OBO maps the three dimensional information in a set of vertices of polyhedra onto a feature for an object. An open ball is defined as the following ;

Given a point x_0 *in a set* X *and a real number* $r > 0$, *we define a set as open ball:*

$$B(x_{0;}r) = \{x \in X | d((x, x_0) < r)\} \qquad (1)$$

where x_0 *is the center and* r *is the radius.*

The OBO is an operator that transforms the information of a three-dimensional object to a matrix of the norms from an object. Since the number of vertices of a three-dimensional object approximation is finite, there exists one sphere which contains the approximated model object inside of it. The norm in the three-dimensional space is the distance between two vertices as defined in Equation (2), where N is the number of vertices of an object. Each ith vertex is a center of an open ball as defined below :

$$norm\ s_i = d_{i,j}(vertex\ x_{i,} vertex\ x_j) \qquad (2)$$

where i and j are integers in the range of [0,N].

Then, we normalize the norms with respect to the maximum of the chosen set of norms for a vertex, such that all norms are within a unit interval [0,1], as it is defined in Equation (3).

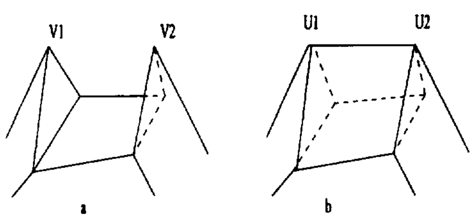
$$f(i,j) = \frac{norm\,(i,j)}{max_i} \qquad (3)$$

where i refers the ith vertex as a center of an open-ball and j refers other vertices of an object. Qualitatively, for a vertex fixed at a center of an open-ball, there are moments that the closure of the open ball meets some vertices as the radius of the open ball increases. We map each radius of the open balls which meet vertices onto a non-negative real line as a monotonic ascending sequence. After mapping the norms onto a non-negative real line as a monotonic ascending sequence, we normalize each radius of an open ball with respect to the maximum of the sequence obtained for each sequence, such that the sequence becomes a non-decreasing sequence in [0,1] on a real line. The collection of sequences constructs a matrix of norms from an object.

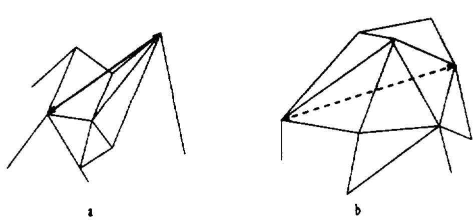
3. Concavity and Convex Problem

The procedures of OBO described above generate a matrix of norms for a three dimensional object. However, there exist cases that the same matrix of norm is extracted from two different objects when two objects are different by concavities and convexities. Figure 1 is a part of a three dimensional object, which has a concavity where an edge connecting vertex V_1 and vertex V_2 does not exist. When a set of norms is extracted from the case of Figure 1, the same norms are extracted for the case of Figure 1 where the edge connecting vertex U_1 and vertex U_2 is contained on the surface of an approximated object.

This occurs because we used only information contained in vertices without considering fully the information contained on the surface. The task is to find a way to eliminate this interruption. The main subject here is how to make a difference between these two feature matrices from two different objects. As a solution to this problem, we obtain the difference of these two objects by eliminating the norm from vertices V_1 and V_2 such that the norm from concave be different from the norms from concave. In order to remove the norms from the concaves on the surface of an object, we first examine whether a vector from one vertex to another is also an edge of a model object. If this is true that the vector is an edge of the model object, we keep it for feature matrix of the object. If not, we test whether this vector passes through any polygon to reach another vertex. If it passes through any polygons of an object, we use this vector for feature extraction. If not, we do not use it for feature extraction. Through the process mentioned above, we keep all the edges and the norms which connect two vertices passing through any polygons of the model object. Pictorial description in Figure 2 shows cases of norms that are not used for feature extraction.



(Fig. 1) Concavity and convex on the surface



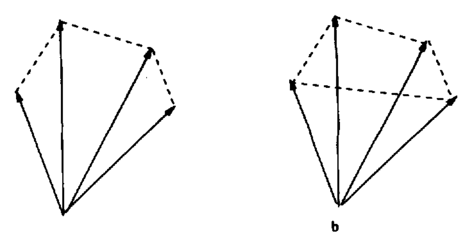
(Fig. 2) Norms not used for the feature extraction

Figure 2 describes a case of external vector, which does not pass through any surface of the model object, and Figure 2 describes a case of internal vector, which also does not pass through any point of the surface of approximated object in its path from one vertex to another. Both of the norms in the example do not pass through any polygons of a model object.

4. Method of Undesired Norms

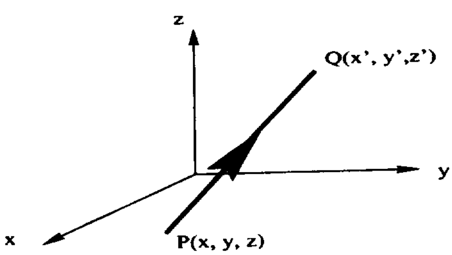
As it is discussed in the previous section, the chords, which connect two vertices, from a three dimensional model object are divided in two categories. One is a set of chords from a model object which is useful for feature extraction, and the other is a set of chords which is not useful for feature extraction.

The chords used for feature extraction consist of two kinds of chords. The first kind is the group of chords which are the edges of polygons, and the second is the group of chords which are not edges but connect two vertices of a model object. The second always passes through at least one of the polygons of an approximated 3-D model object. If we use only the edges of a model object for feature matrix extraction, there is a case that we may not be able to distinguish a feature of the 3-D object from the feature for two dimensional objects where both 3-D and 2-D objects are approximated by patches of polygons. As an example, Figure 3.a illustrates how the interconnected chord which is not an edge of model object can provide 3-D information. Figure 3 illustrates an example of 2-D object and the second shows an example of 3-D object.



(Fig. 3) Norms for 2D and 3D objects

In order to obtain two kinds of norms, we can obtain the norms which are edges of polygons of an object by examining whether each set of two vertices belong to at least one of the polygons. However, extracting the norms which pass through any polygons of an object is not that simple. From a 3-D object with N number of vertices, we obtain N by N matrix which contains N^2 number of norms. Secondly, we construct the first category norms which are the edges of a polyhedra. In the third step, we examine those norms from the first step which are not grouped at the second step whether each chord passes through any other polygons. If it does, then we group them as useful information. Through these steps we eliminate the unnecessary norms and group a set of norms for feature extraction. Mathematically, it is quite a simple matter as described below. Suppose that there are two vertices $P(x, y, z)$ and $Q(x', y', z')$ in the three dimensional space as shown in Figure 4.



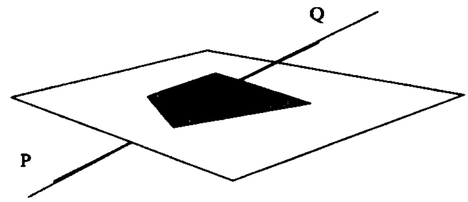
(Fig. 4) Vector between two vertices

The chord connecting two vertices can be defined as a vector with variable t which t varies between 0 and 1 as described in Equation (4).

$$\overrightarrow{pq} = (x+t[x'-x], y+t[y'-y], z+t[z'-z]) \quad (4)$$

When we expand the range of the variable t to $(-\infty, \infty)$, the chord becomes a line.

We are interested in only the interval where t varies from 0.0 to 1.0 of the line as shown in Figure 5.

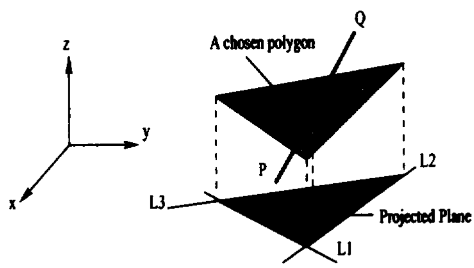


(Fig. 5) A chord passing through the plane that contains a polygon of an object

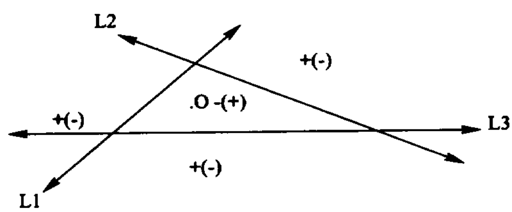
In the process of checking if a chord which contains the vertices P and Q passes through any polygons, we first check whether a line which contains the chord is parallel to the planes where the polygons are contained within. There are two cases of a chord being parallel to a plane. The first case of Figure 5 is that a line is on a plane, such that the distance between a line and a plane is zero. The second is the case that a line is parallel to a plane with some distance. In both cases a line does not pass through a plane.

Another important subject is that the intersecting point must be on a chord. In other words, the value t at the intersection point must be between 0.0 and 1.0, such that we check whether a chord intersects a plane where a polygon lies on. If not, the intersection point of the line is not in on the chord between P and Q.

When the variable t is in the range [0,1], then the chord intersects the plane which is located between two vertices P and Q as shown in Figure 6.b. Even so, we need to examine whether the chord passes through the polygon contained in the plane which the chord intersects. There is a case that the intersecting point locates between the two vertices P and Q, but the line does pass through the plane which contains the sample polygon but not the polygon of object itself. If the intersecting point is not inside the polygon of an object and just passes through the plane which contains the polygon, the line is not passing through the chosen polygon, which is not the case we want. Thus it is necessary to check whether the intersecting point contained inside of the specified polygon. For this purpose, a more or less simple linear segmentation method is used as described below. In order to check whether the intersecting point is contained on the polygon, we use the property that a polygon is constructed by several chords as described in Figure 7.



(Fig. 6) Projection of a polygon onto the x-y plane

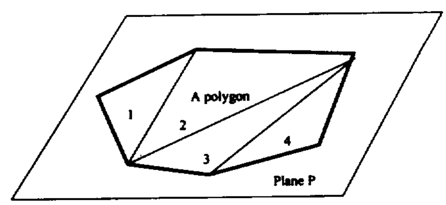


(Fig. 7) Projection of an intersection point on the x-y plane

As the first step, we project the polygon as well as the intersection point onto the x-y plane, y-z plane, and z-x plane. Since only one projection cannot determine whether the intersection point is on the polygon, a plane can be vertical to x-y plane, generally three projections are needed in this procedure. However, it is clear that there is no need to do three projections if the plane which contains the interested polygon is not perpendicular to any of three two-dimensional planes. With three directional projections, a quite simple procedure is employed to decide whether the intersection point is on the projected polygons. Let's choose the x-y plane which the projection of polygon lies on. Then there are two cases that the projection of the polygon on the x-y plane is a line.

When the plane which contains the polygon is vertical to either the variable x is fixed or the variable y is fixed. We use the x-z plane If the variable x is constant and the y-z plane if the variable y is constant. Using this approach, we can project a polygon onto a plane for examining the positions of points of polygon. For the case that we choose, the points inside of the polygon is projected inside of the projected polygon if the plane which contains polygon is not vertical to the x-y plane. If so, then we use either x-z or y-z plane.

Suppose that the lines L_1, L_2, and L_3 are the projected lines of the edges of a polygon on the surface of an object onto the x-y plane as shown in Figure 8.



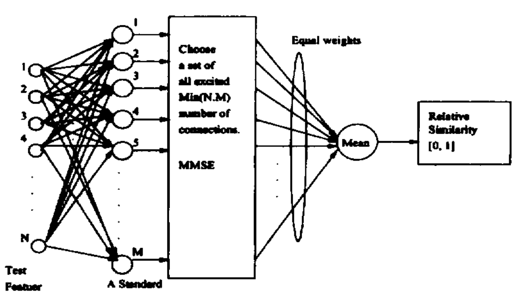
(Fig. 8) Checking intersection points on the non-triangle polygons by triangulation

The point that a chord $\overrightarrow{(P, Q)}$ meets the plane which contains the polygon we are interested in is projected as a point O' on the x-y plane also. Each line can be expressed as $f(x,y,z)=0$ such that the value of the value of $f(x,y,z)$ at the projected intersection point O' is either $f(O'(x,y,z)) < 0$ or $f(O'(x,y,z)) > 0$. Three different lines out of projection are illustrated in Figure 8. If the function for all points inside of the triangle constructed of three lines $f(x,y,z)$ is positive (negative) while the function for all points outside of the triangle is negative (positive) as shown in Figure 8, we can determine whether the projected point lies inside of the triangle by checking the sign of the function $f(x',y',z')$ for a projected point $O'(x',y',z')$.

The goal of this is to let the three line equation have the same sign for all points inside of a triangle as in Figure 8. The method employed here is that we control the signs of the coefficients of a line equation such that the functions $f(x, y, z)$ for L_1, L_2, and L_3 of a chosen point yield the same sign. In order to check the sign of three equations at a fixed point inside of a triangle, let's choose the centroid of a triangle as shown below equation for three vertices $V_1 = (x_1, y_1, z_1)$, $V_2 = (x_2, y_2, z_2)$, and $V_3 = (z_1, z_2, z_3)$,

$$O_c = \left(\frac{x_1 + x_2 + x_3}{3.0}, \frac{y_1 + y_2 + y_3}{3.0}, \frac{z_1 + z_2 + z_3}{3.0} \right) \quad (5)$$

With checking the signs of three line functions $f_{L1}(O_c)$, $f_{L2}(O_c)$, and $f_{L3}(O_c)$, we change the signs of coefficients such that all three line equations have a positive number for each point inside of the triangle. Then, we check the signs of $f_{L1}(O')$, $f_{L2}(O')$, and $f_{L3}(O')$ whether all of them have positive values. If so, then the projected point O' is inside of the projected triangle, which implies that the intersection point is on the polygon. This method can be applied to any polygons since we can represent any polygons as several triangles attached by edges as shown in Figure 9 above. The same method can be applied onto polygons as shown in Figure 9 since any polygon can be a collection of triangles that cover, without overlapping, the whole plane of arbitrary polygon which has more than three vertices.



(Fig. 9) Connections for Similarity Measure

5. C[0,1] PARTITION

Suppose that we obtain two matrices X and Y of norms for two different objects A and B that are to be compared. The two norm matrices X and Y have a finite number of elements since only the finite number of vertices are used for approximating three-dimensional object. All of the elements of the norm matrices X and Y are between 0.0 and 1.0 as the results of using the Equation (3). Analytically, the information on the geometrical structure of objects is mapped into a continuous interval C[0,1] as a set of finite number of points in C[0,1]. In this research, we employed the metric distance defined as the least mean square (LMS) for measuring the distance between three-dimensional model objects.

First we partition uniformly C[0,1] by a proper number we can obtain by increasing the number of partitions until we separate the objects which we want to separates. Choosing a proper number of partition is another subject to be mentioned. Second, let the percentile of the number of points in a partition, which is the frequency of points in each interval, be the pulse magnitude at the interval as shown in Equation (6). This is a row of a feature matrix of an object.

$$F_i(j) = \frac{Num(x_i, j)}{Num(X(i))} \quad (6)$$

where Num is a function that counts the number of elements of its domain and $x_i(j) = [\frac{j}{N}, \frac{j+1}{N}]$ and i is the ith row of norm matrix and j is the jth partition. Equal weight is applied to each partition, such that each partitioned interval weights $W_j = 1/N$ $W_j = \frac{1}{N}$ and $\sum_{j=1}^{N} W_j = 1.0$. Choosing a proper number of partition is important in generating a feature matrix. Intuitively, we can see that too small number of partition needs to be avoided. For example, if we use a partition $P = 2$, all column vectors with symmetric distribution are to be 0.0 distance.

Suppose that there are P number of elements in one row of a feature matrix, such that each partitioned interv . has at most one point. When we increase the number of partitions by k to P partitions, we increase only null intervals that do not have any point. Thus as $\lim_{k \to \infty} \frac{P+k}{k} = 1.0$ shows, the similarity between two sequences becomes 1.0 due to the zero Least Square Error(LSE) distance between two partitions of none elements. This implies that we need to have a proper number of partitions and that partitioning C[0,1] into a very large number does not help for object classification. When the number of partitions P is small compared to the number of sample vertices, separation of compared feature vectors is not stable either because the information separation is not successful, especially when the compared objects are similar. The proper number of partitions for each different group of objects is different. If the objects to be compared have similar feature matrix, we need to choose a larger number of partitions and the system becomes sensitive to noise.

6. Classification Method

Classification between 3-D objects that are approximated by polyhedra is performed by comparing of their features obtained. As it is simply described, we first measure the similarity between two sequences of the same length, which are obtained by the partitioning process, using $\sum_{j=1}^{N} \frac{1}{N} e^{-|x_{i,j} - y_{i,j}|}$. Then we choose M number of vertices that is the number of vertices of the smaller between two objects to be compared. Then we choose M connections which yield to have the largest similarity. The similarity measured between objects is calculated by Equation (7):

$$Similarity = \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{1}{N} e^{-|x_{i,j} - y_{i,j}|} \quad (7)$$

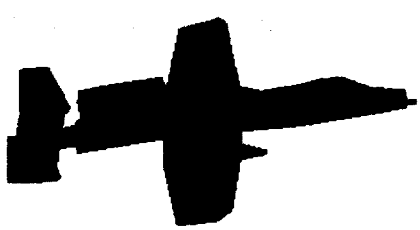
where $x_{i,j}$ is an element of a problem object feature matrix and $y_{i,j}$ is an element of a standard object feature matrix, N is the number of partitions, and M is the number of vertices that is the smaller than the other. In this process, each partition for a vertex weights equally $\frac{1}{N}$. Also each vertex weights uniformly $\frac{1}{M}$. Through the procedures described above, we measure the metric distance between two objects using their feature matrices, the least mean square(LMS).

7. Experimental Results

The purpose of experiments is focused not only on classification of the elements of a set of patterns (objects) but also on the relative similarity measurement among them.

The noise tolerance of the algorithm developed is also measured for different number of partitions of C[0,1]. In experiments, A10, F15, F16, and F18, four models, the three-dimensional approximations of the actual objects, are used as shown in Figures 11, 12, 13, and 14 as described in <Table 1>.

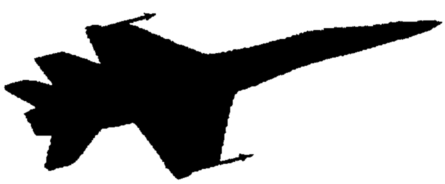
With increasing the number of partitions, the relative similarities between the four objects are measured. The Figures 11, 12, 13, and 14 show the relative similarities of objects with respect to A10, F15, F16, and F18 plane models respectively.



(Fig. 10) A10 model plane without noise

(Fig. 11) F15 model plane without noise

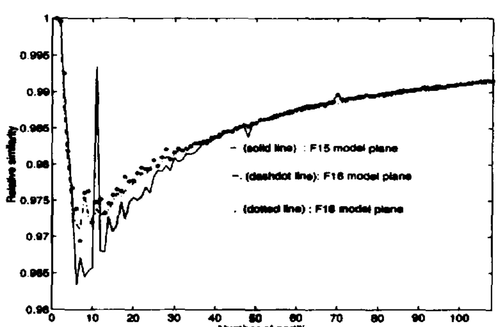
(Fig. 12) F16 model plane without noise



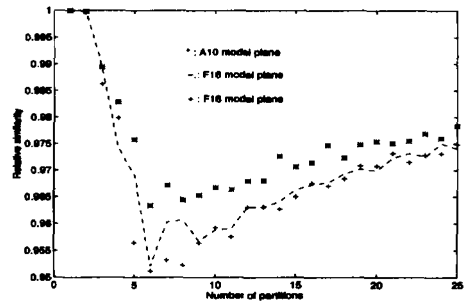
(Fig. 13) F18 model plane without noise

<Table 1> 3D model object approximation

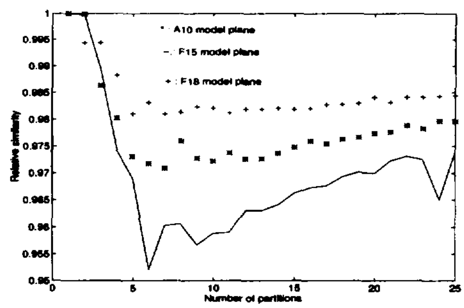
Specification	A10	F15	F16	F18
Number of Vertices	164	702	263	377
Number of triangles	255	638	440	783



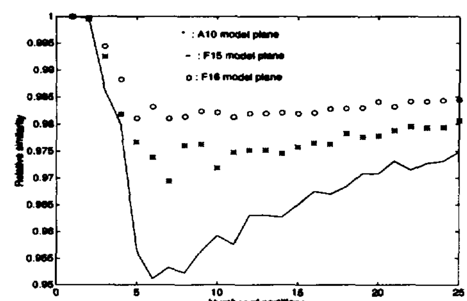
(Fig. 14) Relative similarity w.r.t. A10



(Fig. 15) Relative similarity w.r.t. F15



(Fig. 16) Relative similarity w.r.t. F16



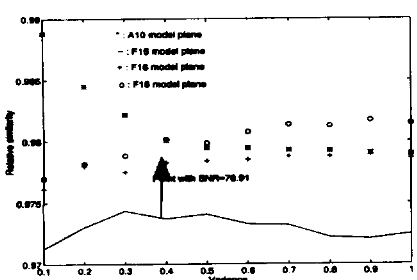
(Fig. 17) Relative similarity w.r.t. F18

The relative similarity measurement in Figure 15 is properly established when the number of partitions is between 12 and 24 approximately as shown in <Table 2>. The upper bounds of the number of partitions for F15, F16, and F18 is greater than 24, such that the least upper bound of the intervals where classification is possible is 24. The greatest lower bound of the intervals in <Table 2> is 12. From the results, we choose the mean value of the interval [12,24] that is common to all of the cases. In this case, the mean is 18. The process described above builds a unique information relationship of four objects. We define this process of finding a proper number for partition. From this experiment we observe that the information on the four objects constructs an information space.

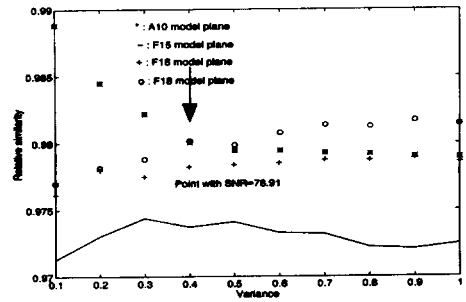
<Table 2> The lower and upper bounds of partition

Specification	A10	F15	F16	F18
Lowe Bound	12	5	4	3
Upper Bound	24	greater than 24	greater than 24	greater than 24

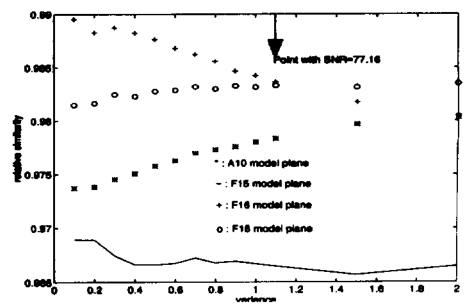
In this section, we discuss about the tolerance to noise of the classification. In order to measure the tolerance to noise a proper number 18 which is obtained from the process of finding a proper number of partitions, is used. Experiments show that classification has some tolerance to noise as shown in Figures 19, 20, 21, and 22. Noise $N(0,\sigma)$ is added to each the standard vertices of four objects. Figures from 25 to 36 display the three dimensional models with noise for A10, F15, F16, and F18 where we can see how much the objects have been changed by noise in vertices. The energy contained in noise is defined as $\sum_{i=1}^{N}(0,\sigma)$ and the energy contained in a feature matrix is defined as $\sum_{i=1}^{N}\sum_{j=1}^{N} Norm^2_{i,j}$ where N is the number of vertices for a model object. Experiments show that the system has noise tolerance in a certain amount as shown in <Table 3>.



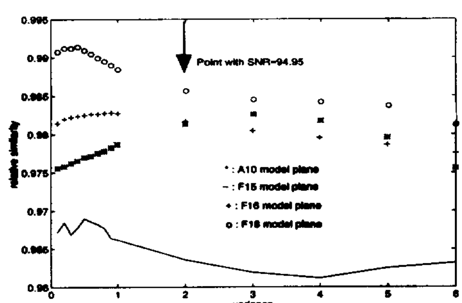
(Fig. 18) Relative distance of objects with noise to the A10 without noise



(Fig. 19) Relative distance of objects with noise to the F15 without noise



(Fig. 20) Relative distance of objects with noise to the F16 without noise

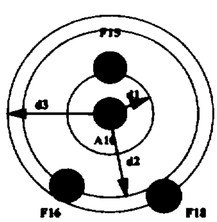


(Fig. 21) Relative distance of objects with noise to the F18 without noise

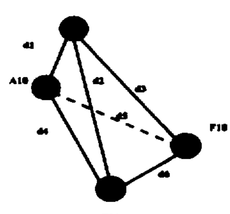
<Table 3> The maximum variance and the least SNR for recognition

Noise Tolerance	A10	F15	F16	F18
Variance	0.4	100	1.1	2
SNR	76.91	36.08	77.16	94.95

However, it is true that the tolerance varies depending on what objects are in the group to be classified. Figure 23 illustrates the spatial locations of feature matrices of objects in the information space constructed by A10, F15, F16, and F18. In Figure 23, the locations of feature matrices of four objects with respect to A10 are described. F15 can be laid any place on the circle of radius d1 centered at A10. Similarly, F18 and F16 can be on the circle with radius d2 and d3 respectively. However, the locations of the feature matrices are put in three-dimensional structure as shown in Figure 24, the interrelation between feature matrices becomes more clear. This interpretation can describe for the possible cases that cannot be explained in two-dimensional information space. One possible example is that both the feature matrices of an object that is needed to be identified and A10 have the same distance to both F16 and F18 but the object has also some distance to A10. In this case, two-dimensional information space cannot be used. Therefore, the interpretation of the experiment should be done by replacing sphere to circle.



(Fig. 22) Four objects in the information space w.r.t. A10



(Fig. 23) Feature location in 3-D the information Space

8. Conclusion

This paper presents a new method of recognizing three-dimensional objects (classification) with feature extraction. The Open-Ball Operator (OBO) extracts the norm matrices from the 3-D model objects and the feature matrices are extracted from the norm matrices by normalizing the distribution in a properly partitioned interval. The distances between the extracted features are measured using the least mean square (LMS). The quantity of distances between features is relative, which means that the measured distance should be interpreted as relatively not in linear scale. The advantage of this method is that the distances between one object that we are interested in and the objects that we have learned through extracting features from objects, can be measured not by just choosing the one that has the minimum distance out of the measured distances but by locating the feature matrix of object into the three-dimensional structure of information space. Thus we can have the information on how much the model objects are similar to each other relatively by means of the metric distance between features obtained from the model objects. In addition to this advantage, the OBO provides invariance not only on rotation, translation, and scale but also on tolerance to noise onto three-dimensional object recognition.

References

[1] S. S. Kim, T. Kasparis, and G. A. Schiavone, ''Three-dimensional object recognition using wavelet de-noising," SPIE AeroSense, 1996.

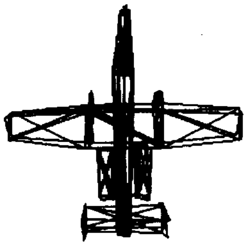
[2] S. S. Kim, T. Kasparis, G. A. Schiavone, and C. R. Madhudram, "A similarity measure for non-uniformly sampled multi-resolution terrain data using open-ball operator," The 30th IEEE Conf. on Signals, Systems and Computers, Asilomar, Nov. 1996.

[3] D. H. Ballard and C. M. Brown, Computer Vision, Prentice Hall, Englewood, NJ., 1982.

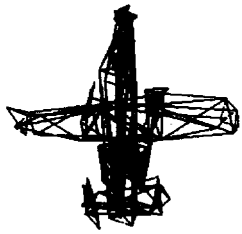
[4] G. B. Thomas, JR. R. L. Finney, Calculous and Analytic Geometry, Addisin-Wesly Publishing Company, 1988.

[5] J. T. Tou, R. C. Gonzalez, Pattern Recognition Principles, Addisin-Wesly Publishing Company, 1974.

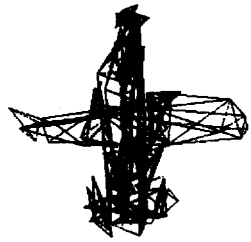
[6] {bow} Sing-Yze Bow, Pattern Recognition, Marcel Dekker, Inc. 1984.



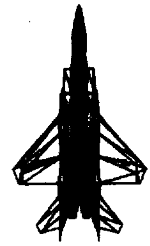
(Fig. 24) A10 with noise N(0, 0.2)



(Fig. 25) A10 with noise N(0, 0.6)

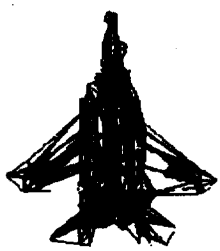


(Fig. 26) A10 with noise N(0, 1)

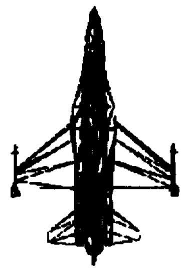


(Fig. 27) F15 with noise N(0, 20)

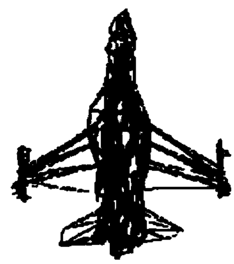
(Fig. 28) F15 with noise N(0, 60)



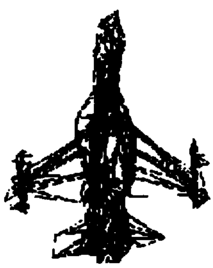
(Fig. 29) F15 with noise N(0, 90)



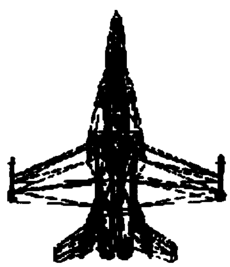
(Fig. 30) F16 with noise N(0, 0.2)



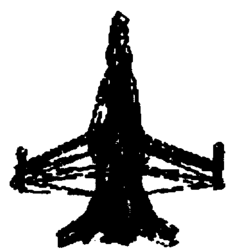
(Fig 31) F16 with noise N(0, 0.6)



(Fig. 32) F16 with Noise N(0, 1.0)

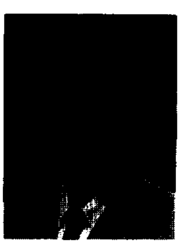


(Fig. 33) F18 with noise N(0, 0.2)



(Fig. 34) F18 with noise N(0, 0.6)

(Fig. 35) F18 with noise N(0, 1)



김 성 수

e-mail : sskim@cvweb.etri.re.kr

1983년 충북대학교 전기 공학과 졸업(학사)

1989년 아칸소 주립대학 전기공학과 졸업(공학석사)

1997년 중앙플로리다 주립대학 전기공학과 졸업(공학박사)

1998년~현재 한국전자통신연구원 선임연구원

관심분야 : 디지탈 영상 및 신호처리, 디지탈 통신, 최적화, 웨이브렛, 함수해석학