論文2001-38SP-1-6

2차원영상에서 가려진 물체를 분리하기 위한 면관계 특징

Face Relation Features for Separating Overlapped Objects in a 2D Image

宋必宰*, 崔洪珠*, 車亨泰**, 韓獻洙**

(Piljae SONG, Hongjoo CHOI, Hyungtai CHA, and Hernsoo HAHN)

요 약

본 논문은 2D 영상에서 겹쳐진 물체를 분리하기 위한 새로운 기법을 제안한다. 제안된 알고리즘은 겹쳐진 물체를 분리하는 보편적인 기법으로서의 매칭 알고리즘이 가지는 계산상의 부담을 줄이고자 하였다. 영상에서의 물체의 표현은 attributed graph를 사용하며, 각 node와 arc는 물체의 면과 면간의 관계에 각각 대응시킨다. 또한, 각 arc는 그 parameter로서 관계계수를 가지며 이는 arc를 중심으로 양 끝에 존재하는 임의의 node의 상대 node에 대한 가려짐 상태에 의해 정의된다. 각 node는 이웃 node와의 관계에 의해 다양한 패턴으로 분류되며, 제안된 패턴을 이용하여 node들의 homogeneity를 검사한다. 끝으로, Homogeneity를 만족하는 node들을 하나의 집합(node set)으로 grouping함으로써, 가려진 물체와 가리는 물체를 분리하게 된다. 본 논문에서 제안한 알고리즘은 임의의 형태로 놓여있는 겹쳐진 물체를 효율적으로 분리하고 있으며, 매칭단계 이전에 물체를 분리함으로써 매칭에 필요한 시간부담을 크게 줄일 수 있음을 실험을 통해 보여주고 있다.

Abstract

This paper proposes a new algorithm that detects and separates the occluding and occluded objects in a 2D image. An input image is represented by the attributed graph where a node corresponds to a surface and an arc connecting two nodes describes the adjacency of the nodes in the image. Each end of arc is weighted by relation value which tells the number of edges connected to the surface represented by the node in the opposite side of the arc. In attributed graph, homogeneous nodes pertained to a same object always construct one of three special patterns which can be simply classified by comparison of relation values of the arcs. The experimental results have shown that the proposed algorithm efficiently separates the objects overlapped arbitrarily, and that this approach of separating objects before matching operation reduces the matching time significantly by simplifying the matching problem of overlapped objects as the one of individual single object.

I. Introduction

Occlusion among objects frequently occurs in part assembly and inspection lines. Many researchers have proposed various solutions for this problem which can be mainly categorized into two approaches; one is to use 2D information and the

* 正會員, 崇實大學校大學院電子工學科

(Dept. of Electronic Engineering, Soongsil University)

** 正會員, 崇實大學校 情報通信電子工學部

(School of Electronic Engineering, Soongsil University)

接受日字:1999年11月15日, 수정완료일:2000年12月27日

other one is to use 3D information from image. Most of those approaches are based on the model matching technique which searches for the features of the object model in the given 2D or 3D images and decides the object as the model if it contains the features being searched for.

In the case of using 3D information, the geometric and spatial features such as volumetric primitives, normal vector of a plane and distances between faces are being used for detecting the occlusion and recognizing 3D objects. For example, Dickinson[1,2] defined an object as a combination of ten volumetric primitives. Every model was represented by an aspect hierarchy which was composed of edge, face and aspect levels. To reduce matching time, the model database was constructed by observing each primitives from the different directions, 10 degrees separated in X, Y, and Z axes. Johnson and Hebert[3] proposed spin image which was constructed for all mesh vertices of the object surface. A spin image was defined by two distances, one from mesh vertex along the vertical line and the other one along the horizontal line, which were used as the criterion for matching. The spin image created for all mesh vertices of the object surface was compared with that in the model library to search the best matching vertices. The matched vertices are grouped to find the best matched model object. Those works using 3D information simplify the process of recognizing 3D objects compared to the works using 2D images, since the representation of 3D objects using 3D features is much easier than using 2D features. However, it costs too much to construct and preprocess 3D image.

On the other hand, 2D image is easier to obtain and less sensitive to noise compared to 3D images, though it loses depth information. When using a 2D image, various features of its boundary image were used to represent the object and to find the best matching object model. For example, Edwards and Murase[4] assumed the possible types of occlusions between objects and made 13 binary masks according to those types to select the object area in an image. In the first step, they used the model primitives with lower resolution to obtain the correlation with the objects in the image and thus generated more than one hypotheses on the object's models and poses. The resolution of the primitives was increased and the matching process was iterated until the number of hypotheses converged to one. Pikaz and Dinstein[5] proposed the matching algorithm between two curves, which have the relation of similarity transformation with each other. Each curve was represented by a total curvature graph in the first step, which is a relation graph between the distance and the change of gradient from any point on the contour, and then the gradient axis of both graphs was sampled periodically. The matching process between the model and the object was performed using the rate of two parameters. One is the length of distance axis that corresponds to the sampled value on the gradient axis and the other is the cumulative length on the distance axis at that point. Performance of these methods using 2D information degrades when the amount of occlusion between objects in an image increases. Also, the computation time increases as the number of object models increases.

This paper proposes a new approach to the problem of separating occluding and occluded objects. The basic idea was originated from the observation that a human being detects and recognizes the occluding object on the top first and then assumes the model of the occluded object based on the shape of the contour of edges, when recognizing the overlapped objects. Thus, the proposed algorithm classifies the relationship between surfaces based on the shape of shared edges into three types: 'connected', 'separated' and 'overlapped'. To represent these types of relationships between surfaces, an attributed graph is used where a surface is represented by a node and their adjacency in 2D image is by an arc. At each end of an arc, a relation value is given which is determined by the number of edges of the surface connected to the surface in the opposite side. Using this classification of arc types, the homogeneous surfaces pertained to the same object, called a 'node set', can be detected by testing the patterns that they construct in the attributed graph. The patterns to be classified as the homogeneous nodes are 'basic', 'self-occluded', and 'equal-position'. The proposed algorithm has been evaluated by the experiment, and its results are explained.

The rest of this paper is organized as follows: section 2 defines the attributed graph to be constructed from a boundary image and illustrates the classification of relationships between two surfaces and among three surfaces. Section 3 illustrates how to detect and separate the occluding and occluded objects from an attributed graph defined in section 2. In section 4, the performance of the algorithm is evaluated by the experiments.

Ⅱ. Attributed Graph

A 3D polyhedral object is observed as a set of polygons in a 2D image which shows only the part of the object located on the viewing direction of the camera. This image is described by the attribute graph, where a node represents the face (polygon) and an arc between two nodes represents their adjacency in the image. Each arc has two relation values on both ends. A relation value on one end of an arc enumerates the shape of adjacency of the surface represented by the node with the surface on the same end. Thus, the relation values at both ends of an arc may not be the same as can be conjectured.

1. Types of the relations between two faces

In the image of a polyhedron, the possible relations between two faces are categorized into three types: 'connected', 'separated' and 'overlapped' as shown in Fig 1. 'Connected' type is defined as the case that two faces are sharing one same edge, and 'separated' type is defined when two faces are sharing no edge. When two faces are not 'separated' and the number of shared edge is not the same as one, then 'overlapped' type is defined. In the case of 'overlapped' type, the edge of the occluding face is intersected by that of the occluded face. As shown in Fig 1(c), the edge of occluding face A is intersected by the edge of occluded face B. All three relations are described by two nodes and one arc as shown in Fig. 1(d), and their types are represented by the relation values on the arcs. Relation value m which shows the shape of border between two faces is defined as follows:

$$m = E + S \quad (1)$$

where E is the number of edges whose type is 'connected', and S is 1/(the number of intersected points between two faces + 1). For example, if the arc in Fig. 1(d) is that of Fig. 1(a), then all relation values a and b become 1. If the arc in Fig. 1(d) is

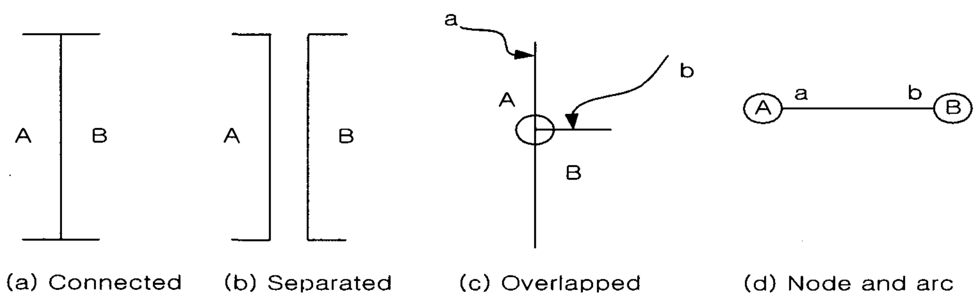


Fig. 1. Types of relations between two faces, and a node and arc representation.
그림 1. 노드와 아크를 이용한 두 면 사이의 관계 표현

that of Fig. 1(c), then the relation values a and b become 0.5 and 1, respectively.

Based on the relation values given on an arc, the relation type between two faces can be inferred and Table 1 shows the possible classifications. Typical cases in Table 1 are depicted in Fig 2.

In Table 1, case 1 occurs when face B occludes face A generating T or Π junctions in the border. This is one of the most frequently occurring cases and the typical example is depicted in Fig 2(a). Case 2 occurs when face A is partially occluding face B. Its typical example is depicted in Fig 2(b). In case 3, two faces, A and B, are adjacent to each other and sharing more than one 'connected' edges. If two faces are sharing more than one 'connected' edges and also their border forms T or Π junction as explained in case 1, then they are classified as case 4 and an example is depicted in Fig 2(d). In case 6, two faces are 'overlapped' without sharing any 'connected' edges in the image while they are located in different planes placed perpendicular to the image plane. Case 7 shows that two faces are

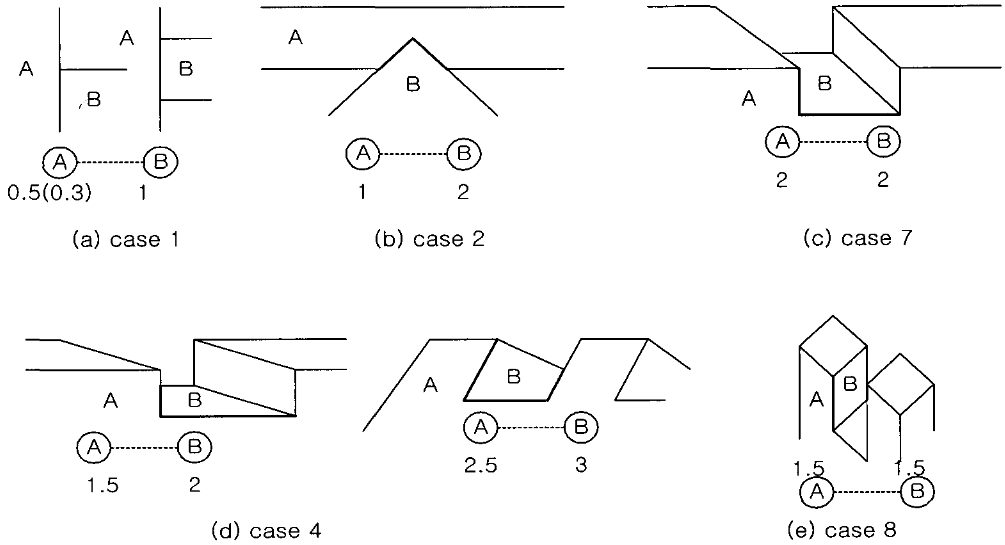


Fig. 2. Examples of some cases given in Table 1.
그림 3. 표 1에 주어진 관계의 예

Table 1. Classification of relation types between two surfaces based on the relation values.
표 1. 관계수의 크기 및 형태에 따른 가려짐 구분.

Case	Relation		Relation Value	Relation Type
1	b > a	a<1, b=1	a: decimal fraction	overlapped
2		b>a≥1	a, b: integer	overlapped
3			a, b: decimal fraction	overlapped
4			a: decimal fraction, b: integer	connected and overlapped
5	a = b	a = b =1		connected
6		a, b<1	a, b: decimal fraction	overlapped
7		a, b>1	a, b: decimal integer	included
8			a, b: decimal fraction	connected

sharing more than one 'connected' edges as depicted in Fig 2(c), and case 8 represents the case where two faces are sharing more than one 'connected' edges and also adjacent each other at the same time as shown in Fig 2(e).

2. Types of the relations among three faces

This paper deals with the polyhedra having vertices where three edges and three faces intersect. Therefore, three faces sharing the same vertex should be able to observed in the occluding object. In the attributed graph, this case is represented by three nodes connected by the arcs of 'connected' type. The proposed algorithm searches those nodes satisfying the above case in the attributed graph at first and calls this node shape a 'node set'. If all node sets are found in the attributed graph, those nodes not included in the node sets are called 'free nodes', each of which is later tested to find the node set that it can be pertained to.

The types of the relationships that any three faces can make can be categorized on the basis of the relation types between two faces. Since three faces (A, B, and C) may be grouped by two in three different ways (AB, BC, AC) and each group can have any of three relation types, the total number of possible cases that three surface can make becomes 3×3×3=27. By ignoring the sequence of nodes, we can reduce the number of possible cases from 27 to 10. Ten possible cases are summarized in Table 2.

Table 3 shows the node patterns of the cases summarized in Table 2. In Table 2, the surface in the starting side of an arrow occludes the surface in the ending side of the arrow. The case where both ends of an arc is arrowed occurs when one surface shares more than one 'connected' edges with the surface on the opposite side of the arc as depicted in Fig 2(c). In Table 2, the last case where any two among three surfaces is 'separated' can be discarded from the list, since the nodes are entirely isolated from each other.

3. Patterns of the homogeneous nodes

To separate the occluding and occluded objects, each node in the attributed graph is tested first whether it can be grouped with other nodes as those of the same object. If a set of nodes in the attributed graph are pertained to the same object, then they are called the 'homogeneous' nodes. In Table 3, each pattern of relation among three nodes can be further classified by considering relation values of an arc. The conditions for the nodes in each pattern in Table 3 to be homogeneous is defined in this section. For convenience, the relation patterns are categorized into three groups: 'basic', 'self-occluded', and 'equal-position'.

1) 'Basic' Pattern

As mentioned above, the homogeneous nodes have a pattern where three nodes are 'connected' by the arcs whose relation values are all 1. For example, the same object is observed from different views and depicted in Fig. 3(a) and (b). They generate the same node pattern, although the relation values are different. In the figures, the following sets of nodes

Table 2. Ten possible cases of relations among three faces, based on relation types between two surfaces.

표 2. 세 개의 node 사이에서 발생할 수 있는 10가지의 관계

cases	AB	BC	CA	cases	AB	BC	CA
1	connected	connected	connected	6	connected	separated	separated
2	connected	connected	overalpped	7	overalpped	overalpped	overalpped
3	connected	connected	separated	8	overalpped	overalpped	separated
4	connected	overalpped	overalpped	9	overalpped	separated	separated
5	connected	overalpped	separated	10	separated	separated	separated

are forming the basic patterns: {A, B, C}, {C, D, E}, {G, H, I}, {J, K, I}.

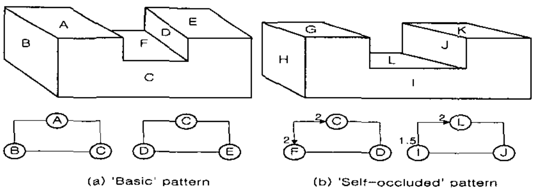


Fig. 3. 'Basic' and 'self-occluded' patterns.
그림 3. 기본패턴과 자체가려짐패턴

2) 'Self-occluded' pattern

This pattern can be found in the image of the objects including concave polygons, one of which occludes another one. For this pattern to be homogeneous, occlusion should be 'self-occluded'. As can be found in Table 2, the arc connecting two nodes with an 'overlapped' relation type may have various relation values. However, there is the common feature in the 'self-occluded' patterns that the occluding or occluded surface is sharing at least more than one 'connected' edges with its counterpart besides 'overlapped' edges. For example, surfaces F and L in Fig. 3 are occluded by surfaces C and I, respectively. Each surface set, {F, C} and {L, I}, shares more than one edges. Therefore, the relation value of the arc connecting these two nodes in the attributed graph becomes larger than or equal to 1. This feature can be formulated as the condition of deciding the self-occluded pattern as follows:

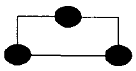
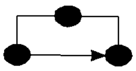
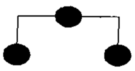
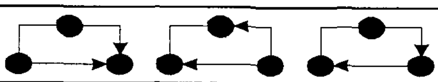
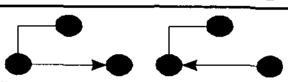
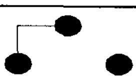
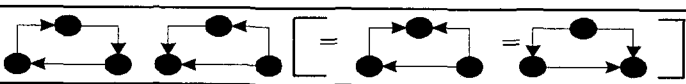
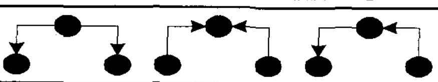
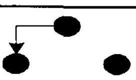
Condition 1: $0 < |n_1 - n_2| \leq 0.5,\ n_1, n_2 \geq 1$

Condition 2: $|n_1 - n_2| = 0,\ n_1, n_2 > 1$

where n1 and n2 are the relation values of the arc connecting the nodes N1 and N2 respectively.

'Condition 1' formulates the case where there is at

Table 3. The node patterns of the cases summarized in Table 2.
표 3. 표 2의 각 조합에 대한 세부패턴

#case	Node and Arc pattern
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

least one adjacent edge shared by two surfaces and the border has a shape of either T-junction or Π-junction. Surfaces L and I in Fig. 3 are pertained to this case. 'Condition 2' considers the case where two surfaces are including each other. As surfaces F and C in Fig. 3, two surfaces are sharing 'connected' edges larger than 1. In order for two nodes having 'self-occluded' pattern to be homogeneous, the third node that both nodes are commonly adjacent to is to be considered. Fig. 3(b) shows the examples of this case. That is, the third node adjacent to each of two nodes should have the 'connected' type. Those nodes included in the 'self-occluded' pattern are considered as homogeneous.

3) 'Equal-position' pattern

For two adjacent surfaces sharing a 'connected' edge to be homogeneous, they have to be located in 3D space at a close distance with reference to the third surface which is commonly adjacent to both surfaces. Thus, the nodes satisfying this condition form a 'equal-position' pattern. For example, F and G are occluded by surface C which are adjacent to both surfaces F and G, in Fig. 4(a). It can be conjectured from this scene that surfaces F and G are located close to surface C. The patterns in case 4 in Table 3 are the examples of this case.

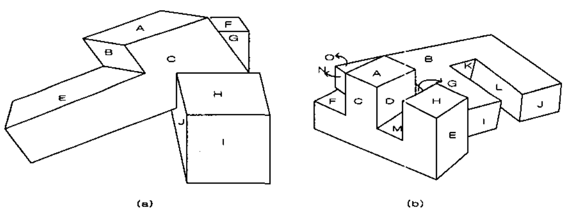


Fig. 4. Examples of occlusions.
그림 4. 겹쳐진 물체의 예

Based on this idea, the homogeneity of two adjacent nodes sharing 'connected' arc is tested by comparing the relation values of the arcs connected to the third node from each node, if there exists an 'equal-position' pattern in 2D image. That is, the homogeneity of two nodes can be conjectured from the fact that the two nodes are pertained to an object different from the one to which the third node is pertained. For example, by confirming that two surfaces, F and G, are pertaining to different object from the one to which surface C is pertained, the homogeneity of two adjacent surfaces F and G can be determined. For this purpose, the type of occlusion relation (T-junction or Π-junction) between surfaces F and C, and G and C is searched. The relation values to construct the simple occlusion type have to satisfy the following condition, where n1 and n2 are respectively the relation values of arbitrary nodes N1 and N2:

Condition 3: $|n_1 - n_2| > 0.5,\ n_1,\ n_2 \le 1$

Two nodes satisfying the above 'condition 3' are pertained to different objects.

4) Isolation Condition

When the types of arcs connecting three nodes are all 'overlapped' or more than one 'separated', the homogeneity of the nodes cannot be determined directly from the node pattern and extra conditions for nodes which can be used as the criteria in the node merging process. For example, in Fig. 4(b), it is possible to classify node D, H and G as homogeneous since two nodes D and H satisfy 'simple occlusion' condition about the third node G. However, nodes D and H do not form 'equal-position' pattern since they are occluding each other. In this case, it is impossible to determine from the node pattern whether the nodes are homogeneous or

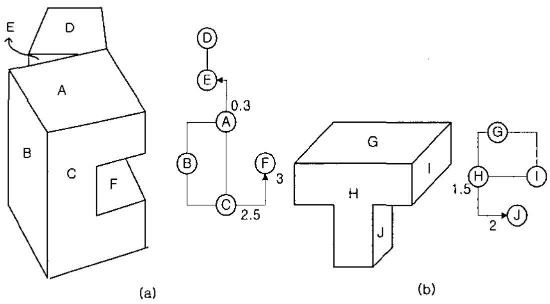


Fig. 5. Isolation condition.
그림 5. 고립조건

not. Other examples are given in Fig. 5 where there exists at least one arc of 'separated' type and thus extra conditions should be applied to.

A node is defined as 'isolated' if only one node is adjacent to. In Fig. 5, nodes D, F, and J are 'isolated' ones with reference to nodes E, C, and H. These nodes correspond to the surfaces forming the contour of the image. 'Isolated' node can be merged as one node set with its adjacent node if the relation values of the arc satisfy 'condition 1' or 'condition 2', although the type of the arc is not 'connected'. For example, in Fig. 5, since there is no adjacent surface except surface C for surface F, surface F satisfies the isolation condition about surface C. In this case, if the relation values of the arc connecting nodes C and F satisfy 'condition 1', nodes C and F can be merged as one node set.

Ⅲ. Separation of Occluding and Occluded Objects

Fig. 6 shows the flow of the process of separating occluding and occluded objects using an attributed graph. The algorithm extracts the list of the node sets, each of which is a set of homogeneous

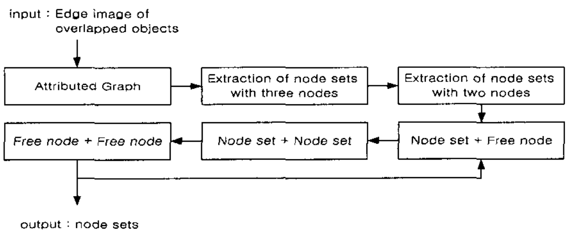


Fig. 6. Flow of the algorithm.
그림 6. 알고리즘의 개략도

surfaces, from the attributed graph. The remained nodes are called 'free nodes' and tested in the next step to see whether they can be merged with any node set or with other 'free nodes' to construct a new node set.

1. Extraction of node sets

The procedure of extracting node sets from attributed graph consists of two steps, as explained in the following:

Step 1: Extraction of 'basic' pattern (Refer to Fig. 7(a))

1) Select a node as Nref connected to N1 and N2.

2) If all arcs are of 'connected' type, three nodes Nref, N1, and N2 are determined as a node set.

3) Repeat the above procedure for all nodes in the attributed graph.

Step 2: Extraction of 'equal-position' pattern (Refer to Fig. 7(b))

1) Select a node as Nref having one arc of 'connected' type, connected to N1.

2) Search the third node N2 which is commonly adjacent to Nref and N1 with a same relation type.

3) If the relations between Nref and N2, and between N1 and N2 are 'overlapped' type and the relation values satisfy 'condition 3', then construct a node set with two nodes Nref and N1.

4) Repeat above procedure for all nodes in the attributed graph.

2. Merging of nodes

The algorithm extracts node sets by considering patterns of any three nodes in attributed graph. Node

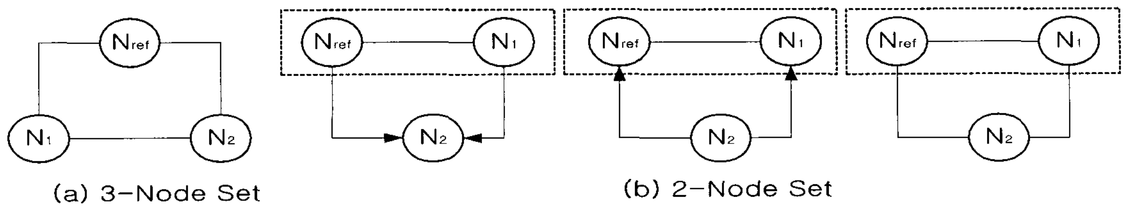


Fig. 7. Extraction of node sets from attributed graphs.
그림 7. Attributed Graph에서 node set의 추출

sets are different or some node are 'free', it is also possible that some node sets or 'free' nodes are of the same object. This section explains the procedure of finding and merging 'free' node and 'free' node, 'free' node and node set, and node set and node set. The merging procedure selects a 'free' node as a reference Nref and tests whether it can be merged into other 'free' nodes or node sets.

Merging of a free node and a node set

Step 1: Among nodes connected to Nref, eliminate those nodes connected by the arcs generating T- or Π-junction since they can be considered to be pertained to a different object. Those nodes are occluded by Nref and the relation values of the arc connecting these nodes with Nref satisfy 'condition 3'.

Step 2: Find the number of arcs connected to each of the extracted node sets.

Step 3: Test the isolation condition of the node set NST with reference to Nref. If satisfies, then merge Nref to NST.

Step 4: If the number of arcs connected to NST is equal to or greater than 2, test whether any three nodes including Nref and two from NST make a new node set. If it constructs a 'basic' or 'self-occluded' pattern, then it forms a new node set. (See Fig. 8(a).)

Step 5: If the number of arcs connected to NST is equal to 1, make a group of three nodes including Nref, one node N1 from NST, and the third node N2 which is connected to both of Nref and N1. If these three nodes construct a 'basic' or 'self-occluded' pattern, then they form a new node set. (See Fig. 8(c) and (e).) If the relation types between Nref and N1, and between N1 and N2 are all 'occluded', then test if they satisfy 'equal-position' pattern. If they do, then they form a new node set. (See Fig. 8(b) and (d).)

If there lefts no 'free' node or 'node sets' to be merged, test whether different node sets are pertained to the same object. Here each node set becomes a reference node set NSref in turn.

Merging two node sets (Refer to Fig. 9)

Step 1: Count the number of arcs connecting two

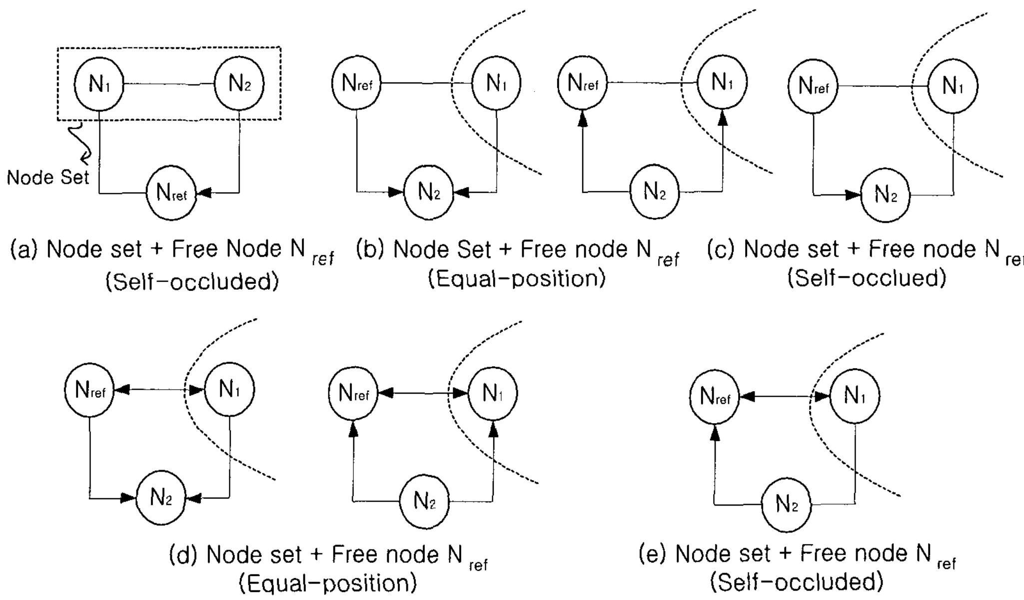


Fig. 8. Merging of a node set and a 'free' node.
그림 8. Node set과 독립 node의 결합

node sets, NSref and NS1. Any node of NSref is connected to any node of NS1, then the connecting arc is counted. However, if the relation values of this arc satisfy condition 3, then it cannot be counted in.

Step 2: If the number of arcs connecting NSref and NS1 is equal to or greater than 2, test whether any homogeneous pattern can be constructed among three connected nodes. (See Fig. 9(a).) If found, then merge those two node sets.

Step 3: If the number of arcs connecting NSref and NS1 is equal to 1, then make a group of three nodes, one node N1 from NSref, one node N2 from NS1, and the third one from other node set which is adjacent to both of N1 and N2. Test whether any of three homogeneous patterns can be constructed

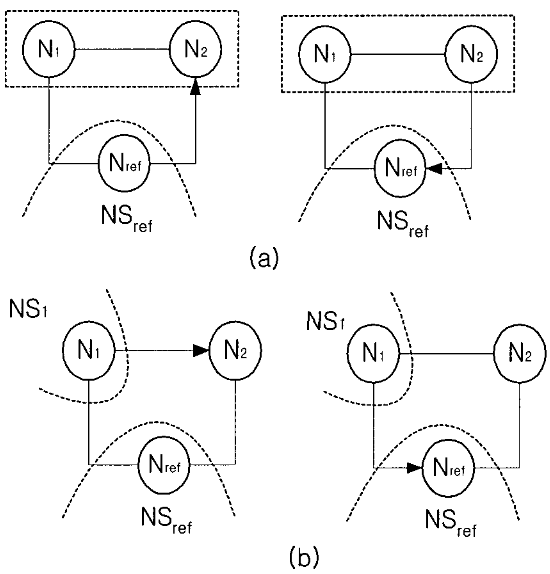


Fig. 9. Merging two node sets.
그림 9. Node set과 node set의 결합

among these three connected nodes. (See Fig. 9(b).) If found, then merge those two node sets.

Finally, test merging conditions among 'free' nodes. If two 'free' nodes satisfy the merging condition to be explained, then they construct a new node set.

Merging two 'free' nodes

Step 1: Count the number of arcs connecting a 'free' node Nref to other 'free' nodes.

Step 2: If Nref satisfies the isolation condition with another 'free' node N1, then construct a new node set with these two nodes.

Step 3: If the number of arcs connected to Nref is equal to or greater than 2, then make a group of three nodes including Nref and two connected nodes. If these three nodes construct any homogeneous patterns, then merge those nodes to make a new node set. (See Fig. 10(a) and (b).) If the relation types among these nodes are all 'overlapped', then test if they satisfy 'equal-position' pattern. If they do, then they form a new node set. (See Fig. 10(a) and (b))

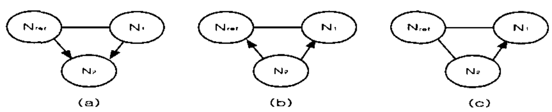


Fig. 10. Merging two 'free' nodes.
그림 10. 두 독립 node의 결합

Ⅳ. Experiment and Discussion

To evaluate the proposed algorithm, the vision system shown in Fig. 11 has been implemented. It is composed of a CCD camera, a frame grabber and a Pentium PC. Fig. 12 shows the polyhedral objects used in the experiment. As mentioned in the context,

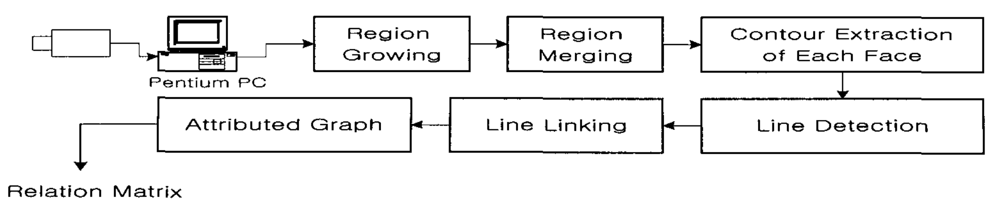


Fig. 11. The hardware system and the block diagram of the algoritm.
그림 11. 실험에서 사용한 시스템 및 영상처리 흐름도

the algorithm is effective when there exists no empty space between the overlapped objects in a test image. When the complete boundary image is not acquired from the given input image, algorithm that recovers the lost edges such as given in [12] is applied before following the procedure.

1. Preprocessing of input image

The proposed algorithm uses boundary images as input. To obtain boundary images from an input image, the faces are segmented by a region growing algorithm in the first step. The segmented regions are merged based on the Mahalanobis distance, proximity between them, and the existence of an edge. Mahalanobis distance was defined in [10] as follows:

$$d_M(X_A, X_B)=\frac{d_E(C_R(X_A), C_R(X_B))}{\sqrt{\sigma_R(X_A)+\sigma_R(X_B)}}$$

where XA and XB represent segmented regions A and B, respectively, and CR(X) is the averaged grey level of region X. dE and σR(X) represent Euclidean distance and the standard deviation of region X, respectively.

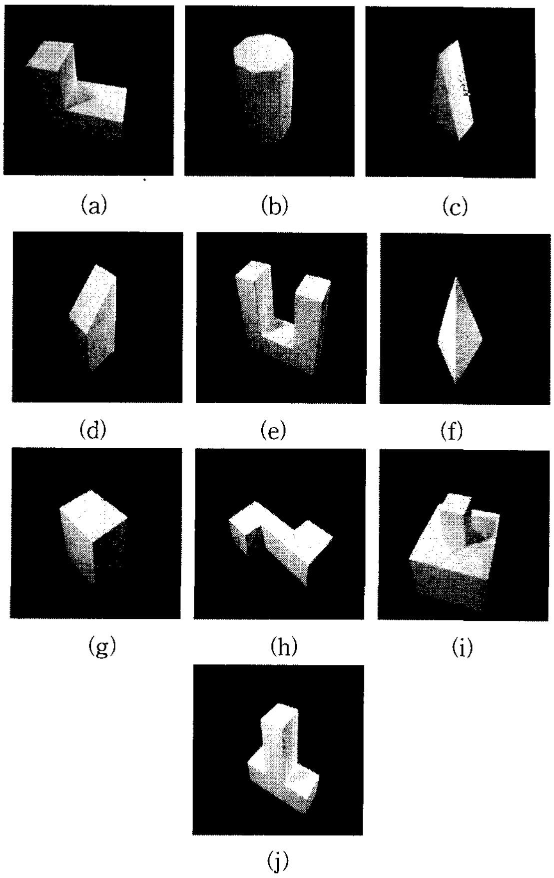


Fig. 12. Object models used in the experiment.
그림 12. 실험에 사용한 다양한 종류의 다면체 모델

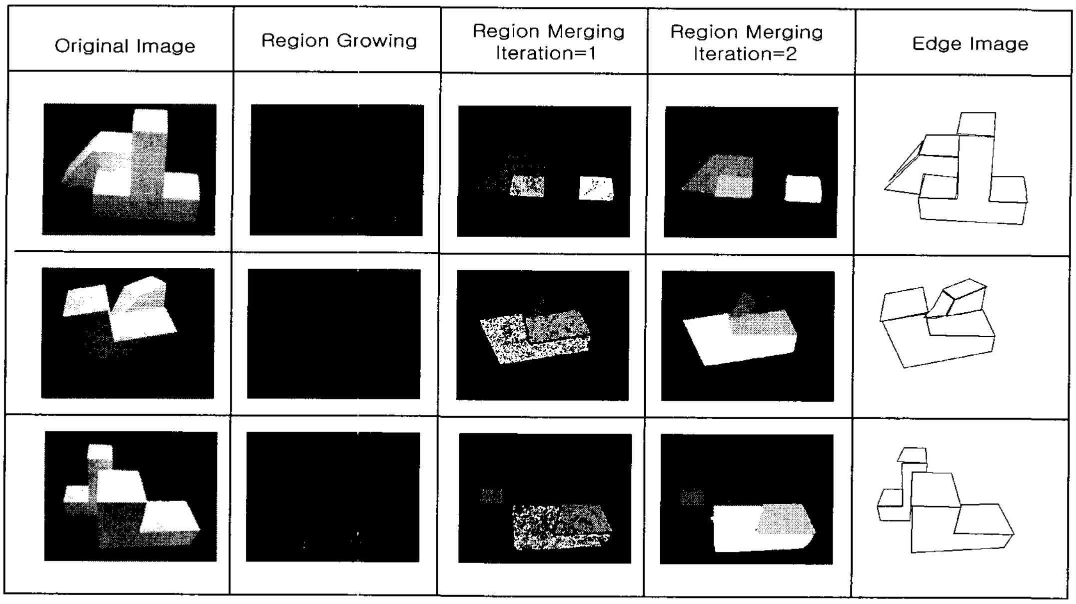


Fig. 13. Original input images and their segmented boundary images.
그림 13. 원영상 및 면이 분리된 경계선 영상

Fig. 13 shows three images of overlapped objects and the process of extracting their edge images. In general, the input image is over segmented by the region growing algorithm and the coherent regions are merged in a couple of iterations of a merging procedure. From the segmented image, the boundaries are extracted by Hough Transform. Each boundary segment is linked and vectorized in such a way that the surface bounded by the boundary segment should be located on the right side of the boundary segment.

2. Separation of occluding and occluded objects

From the segmented boundary image, its attributed graph is constructed and stored in the database array. The relation values of the arcs are determined with reference to the relative positions between nodes, using Eq. (1). Fig. 14 shows the attributed graphs of the images given in Fig. 13 as examples. The intermediate results of each step of the algorithm applied to the attributed graph given in Fig. 14(a) are summarized in Table 4 to show how it works. It extracts the node sets which have

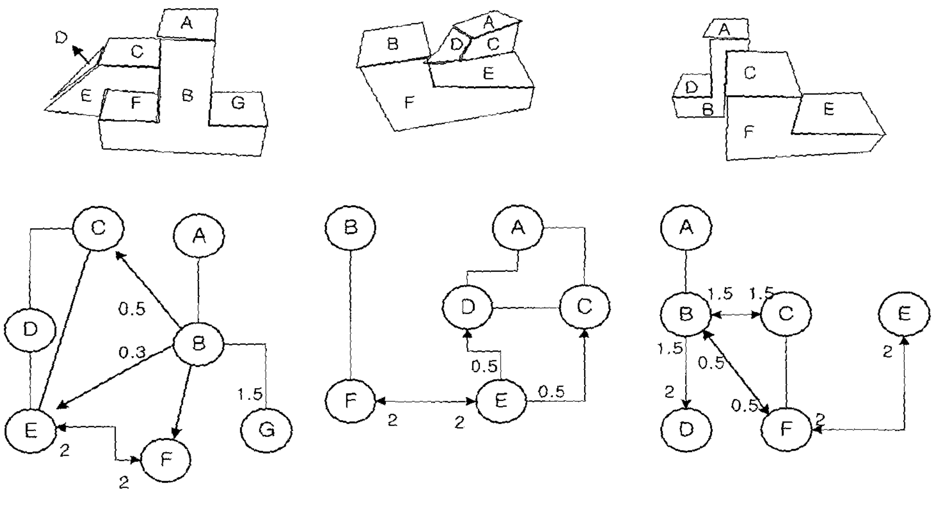


Fig. 14. Attributed graphs of the images given in Fig. 13.
그림 14. 그림 13에 주어진 영상에 대한 attributed graph의 생성

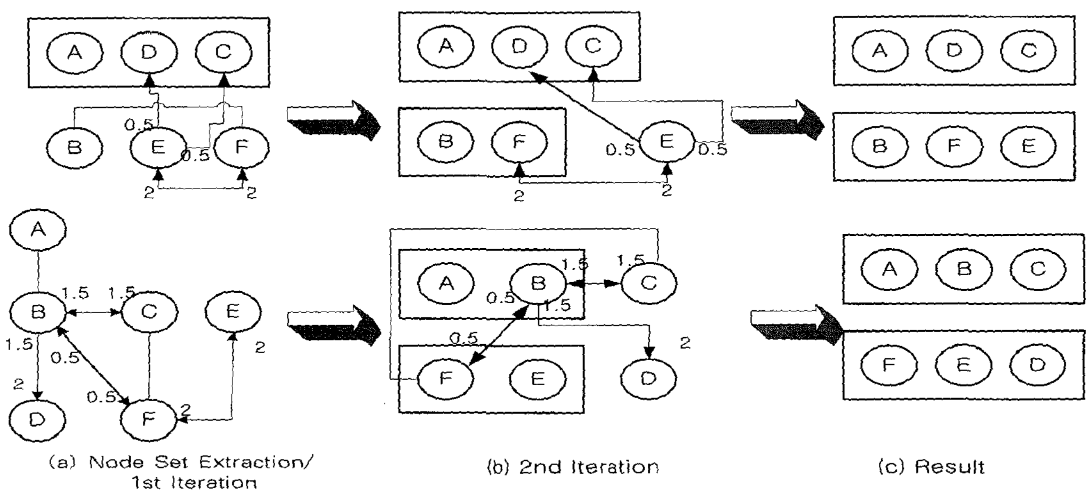
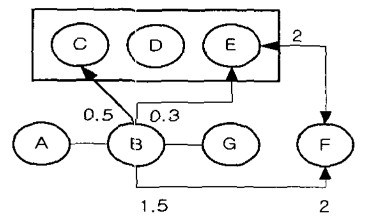
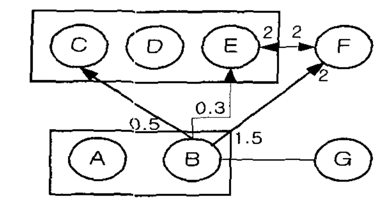
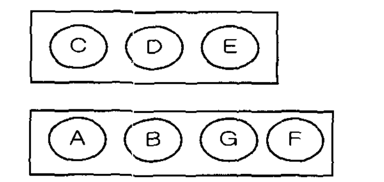


Fig. 15. The results of the algorithm applied to the input images given in Fig. 13(b) and Fig. 13(c).
그림 15. 그림 13(b)와 (c)의 영상에 대한 algorithm 적용 결과

Table 4. Intermediate results of the algorithm applied to the input image given in Fig. 14(a).
표 4. 그림 14(a)의 영상에 대해 algorithm을 적용하는 과정

step	graph	Node Pattern		
node set extraction and 1st Iteration		Basic : C,D,E Self-Occluded : Equal-Position:B,C,E Isolated node : A&B satisfy the isolation condition		
		node set + free node	node set+ node set	free node+ free node
				A+B
2nd Iteration		Basic : C,D,E Self-Occluded : F,{A,B} Equal-position :B,C,E Isolated node : G&{A,B} satisfy the isolation condition		
		node set+ free node	node set+ node set	free node+ free node
		{A,B} + G {A,B} + F		
Result				

'basic' pattern or 'equal-position' pattern. The node set {C, D, E} is an example that forms a 'basic' pattern. In the first iteration, it found that node A satisfied the isolation condition with node B and their relation values also satisfied 'condition 1' so that they could be merged as a node set. In the second iteration, it found that node G satisfied the isolation condition about the node set {A, B} and their relation values also satisfied 'condition 1' so that they could be merged as a new node set. It could also be found that three nodes, E, B, and F, formed a connected graph where the relation values of the arcs connecting two node pairs, F and E, and F and B, satisfied 'condition 1 and 2'. However, the node pair, F and E, do not have a third node which is adjacent to both nodes and thus they do not satisfy the conditions to make a node set. Therefore, the arc connecting node F and the node set {C, D, E} could be eliminated. In other words, node F was isolated from the node set {A, B} and satisfied the sufficient condition to be 'self-occluded' pattern with the node set {A, B}, these three nodes could be merged as a new node set.

m

The results of the algorithm applied to the input images given in Fig 14(b) and (c) are illustrated in Fig. 15, and Fig. 16 shows in the second row the occluding object and in the first row the occluded object, separated by the algorithm.

3. Evaluation

For evaluation of the performance of the algorithm, time complexity is calculated as a function of the number of surfaces, n. The proposed algorithm selects every surface as a reference one and tests its relationship with those surfaces adjacent to the reference surface. If the average number of adjacent surfaces to be tested is m, the time complexity of the proposed algorithm can be represented by O(mn). In practice, the time spent by the algorithm in the cases of those images in Fig. 13 did not exceed 25ms when using the Pentium II PC which is less

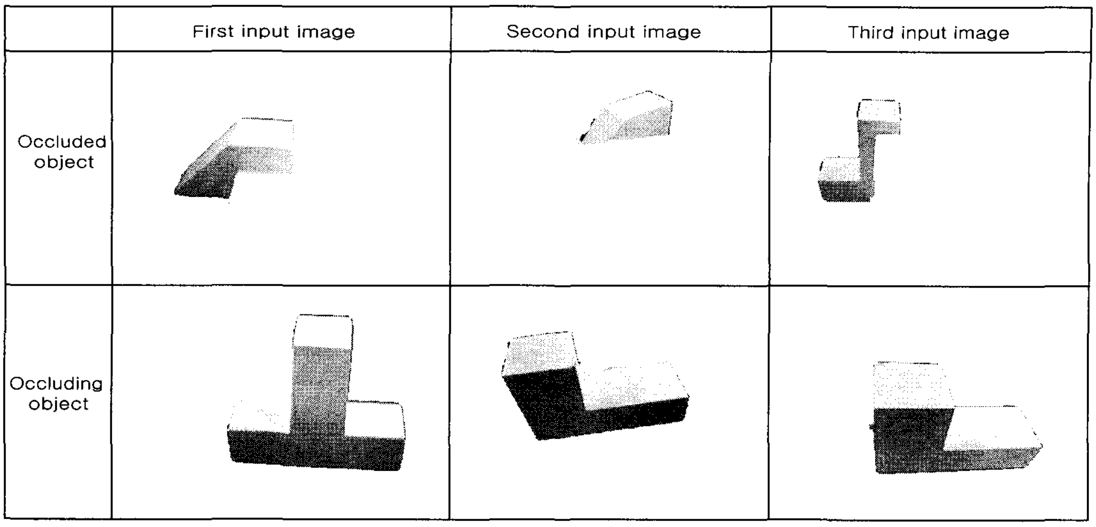


Fig. 16. Results of the separation operation.
그림 16. 물체의 분리작업의 결과

than 5% of the total preprocessing time to segment the object image.

The proposed algorithm has been applied to two algorithms summarized in Section 1 to show that it reduces the recognition time significantly. The algorithm proposed by Pikaz[5] represents an object by a total curvature graph where each curve is represented by discontinuing points and tangential angles at those points. If the number of surfaces in the object image in n and the number of surfaces of the object model to be searched is p, then this algorithm requires the searching time proportional to O(pn2). If the proposed algorithm is applied to this Pikaz' algorithm, then the time complexity is reduced to 1/4 times since the number of surfaces, n, in the object image is reduced by a half on average. The time spent by the proposed algorithm to separate the occluding object from the occluded object, proportional to O(mn), is much smaller compared to the time saved by applying the proposed algorithm, proportional to O(pn2). In the case of applying to the Tsang's algorithm[8], the proposed algorithm improved the recognition time by the same proportion as Pikaz's algorithm did. Both cases have shown that the proposed algorithm saves the recognition time significantly.

V. Conclusionb

This paper proposed a noble approach of separating an occluding object from the occluded one(s). By separating the occluding and occluded objects, the matching problem of multiple overlapped objects is simplified as a matching task of individual single object. The proposed algorithm uses surface level information not edge level one as the unit of representation and the relationship between surfaces is enumerated for ease manipulation of the relations. The experimental results have shown that the proposed algorithm efficiently separates the objects overlapped arbitrarily, and this approach of separating objects before matching operation reduces the matching time significantly.

References

[1] S. J. Dickinson, A. Pentland, and A. Rosenfeld, "From Volumes to Views: An Approach to 3-D Object Recognition", Computer Vision Graphics and Image Processing, Vol. 55, No.

2, March, pp. 130-154, 1992.

[2] S. J. Dickinson, A. Pentland, and A. Rosenfeld, "3-D Shape Recovery Using Distributed Aspect Matching", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No 2, February 1992.

[3] A. Johnson and M. Hebert, "Efficient Multiple Model Recognition in Clustered 3-D Scenes", In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 671-677, 1998.

[4] J. Edwards and H. Murase, ""Appearance Matching of Occluded Objects Using Coarse-to-fine Adaptive Masks", In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 533-539, 1997.

[5] A. Pikaz and I. Dinstein, "Matching of Partially Occluded Planar Curves", Pattern Recognition, Vol. 28, No. 2, pp. 199-209, 1995.

[6] T. Fan, G. Medioni and R. Nevatia, "Recognizing 3-D Objects Using Surface Descriptions", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, No.11, November 1989.

[7] E. Salari and S. Balaji, "Recognition of Partially Occluded Objects Using B-Spline Representation", Pattern Recognition, Vol. 24, pp. 653-660, 1991.

[8] P. TSang, P. Yuen and F. Lam, "Recognition of Occluded Objects", Pattern Recognition, Vol. 25, pp. 1107-1117, 1992.

[9] P. TSang, P. Yuen and F. Lam, "Classification of Partially Occluded Objects Using 3-Point Matching and Distance Transformation", Pattern Recognition, Vol. 27, pp. 27-40, 1994.

[10] Boohyung Lee and Hernsoo Hahn, "Representation and Recognition of Polyhedral Objects using the Signatures of Surfaces in a Single 2ED image", 「KITE Journal-S」, Vol. 34, No. 2, pp. 63-70, 1997.

[11] A. Tremeau and N. Borel, "A Region Growing and Merging Algorithm to Color Segmentation", Pattern Recognition, Vol. 30, No. 7, pp. 1191-1203, 1997.

[12] Y. Kim, H. Hahn and C. Choo, "Determination of the Initial Sensing Trajectory for Active Sensing and its Application to Stereo Vision", The Proceeding of 1993 Asian Conference on Computer Vision, 414-417, 1993.

저 자 소 개

宋 必 宰(正會員)

1993년 숭실대학교 전자공학과 졸업 (학사). 1996년 숭실대학교 산업대학원 전자공학과 졸업 (석사). 1999년 숭실대학교 대학원 전자공학과 박사과정 수료. 1999년 - 현재 경복대학 인터넷정보과 겸임교수

崔 洪 珠(正會員)

1994년 숭실대학교 전자공학과 졸업 (학사). 1999년 숭실대학교 대학원 전자공학과 졸업 (석사). 1999년 - 현재 루탑정보통신 연구원

車 亨 泰(正會員)

1985년 숭실대학교 전자공학과 졸업 (학사). 1988년 미국 Pittsburgh 대학교 대학원 전자공학과 졸업 (석사). 1993년 미국 Pittsburgh 대학교 대학원 전자공학과 졸업 (박사). 1993년 - 1996년 삼성전자 연구소 선임연구원. 1996년 - 현재 숭실대학교 정보통신전자공학부 조교수

韓 獻 洙(正會員)

1981년 숭실대학교 전자공학과 졸업 (학사). 1983년 연세대학교 대학원 전자공학과 졸업 (석사). 1986년 미국 남가주대학(USC) 대학원 전기공학과 졸업 (석사). 1991년 미국 남가주대학(USC) 대학원 전기공학과 졸업 (박사). 1992년 - 현재 숭실대학교 정보통신전자공학부 부교수