Recognizing Primitives Using Aspect-Interpretation Model Matching in A Both CAD and LP-based Measurement System

Zhou Guoqing
Dept. of Computer Science and Technology, Tsinghua University,
Beijing, 100084, P. R. China

ABSTRACT

In order to realize measuring automatically CAD-based object using line photogrammtery[3,4,6], the author in this paper presents recognizing primitive using model matching based on aspect-interpretation. The idea is composed of aspect-interpretation, aspect-graph(also called attribute relational graph) and model matching. The aspect-interpretation contains face-aspect code, face-aspect mergence code and face-aspect split code. The aspect-graph includes attribute relational graph, primitive attribute graph and attribute hypergraph. The model matching is to indicate matching model aspect-graph with image aspect-graph. Finally, the measurement system, which is an integration of CAD, Photogrammetry, Robot and CV, is briefly described. A group of experiments is tested and the results show the recognition scheme is feasible.

KEY WORDS: Recognition, Aspect-interpretation, Model matching, CAD, Line photogrammetry

1. INTRODUCTION

After the author presented the idea that CAD-based industrial object can be measured using line photogrammetry(LP) so as to built up a bridge between CAD and CV system[3,4,6], a set of soft-package of LP for measuring various kind of primitives is developed. For the sake of automatic measurement, it is necessary to recognize automatically these primitives. In this paper, the author describe how to recognize the primitives in measurement system.

Since the model(data structure, strictly speaking, abstract model) of object has been stored in CAD system, recognizing primitives is in fact CAD-based model recognition. Many CAD-based recognition schemes have been proposed in papers published. Woo[18] used convex hull technique to describe the object. Kyprianou[13] applied syntactic pattern recognition methods to classify depression. Henderson[8] and Kung[12] used logic programming and expert system to extract shape features. Joshi[9] developed attribute relational graphs for a part. Lee and Fu[14] proposed an algorithm for the exaction and unification of some features from a CSG tree. Marefty[15] used a geometric reasoning for recognition object. Most of these schemes adopted model matching, however, previous methods have limited success due to complex state of projection in intersection, difference and emergence among primitives. Furthermore, these schemes only recognize an object from relatively complex scene. So, in all of the approaches above, they differ from recognizing single primitives from an object. Comparatively speaking, our recognition exists some characters as follows.

- The primitives to be recognized are numbered.
- Prior CAD information can be provided.

In this paper, the author presents a novelty approach for recognizing different kind of primitives. Briefly, the contributions of this work can be summarized as follows.

- Aspect-interpretation was presented to recognize some simple and regular primitives that lost less information in projection, as well as to assistantly construct aspect-graph.
- The constructing regulation of aspect-graph based on aspect-interpretation was presented.
- The data structure of model matching was studied
- A measurement system was developed.

2. ASPECT-INTERPRETATION PRINCIPLE

A 2-D image is perspective projection of 3-D object, which consists of a number of primitives by Boolean operator(unit, intersection, different) in CAD system. For this 2-D image, a number of edges constructs a face in reasonable allocation way, and a number of faces forms a volume-primitive in reasonable allocation way, which corresponds to a 2-D projection of 3-D primitive. So a volume-primitive corresponds to a 3-D primitive. A number of volume-primitives constructs an object in reasonable disposition. The edge, face, volume-primitive and object form a tree construction, we call *interpretation-tree*. Every layer in this tree describes a kind of state of projection of object. We call *aspect*. i.e. the aspect is used as representing the projection of a 3-D object. So the decomposed state of projection corresponds to *edge-aspect, face-aspect, primitive-aspect* and *object-aspect* respectively. This decomposed process imitates the human recognizing process[1]. If the aspects are connected by acre, we call *aspect-graph*, which is in fact the attribute relational graph (ARG) [2,18,19]. It shows the property of connection among aspect. On the basis of analysis above, the author presents recognizing the primitives by model matching based on aspect-interpretation.

In aspect-graph, in order to recognize primitives, the node may be defined as standing for the primitive-aspect, and the branch may be defined as standing for connected relation between neighbor primitive-aspect. The advantages in this way lie in a) the volume-primitive consisting object is numbered; b) the matching process can directly research due to the independence of primitive in CAD system, c) mainly, the LP soft-package can directly be useful for measurement.

Aspect-graph constructed in this way above is only describing the volume-primitive and their connected relation. During the intersection of between primitives, a face of primitive feature may be divided into several disconnected components thus creating new faces. Furthermore, two intersecting faces of a primitive may not be adjacent. In merging between primitives, there are still analogous case above. So in order to describe faces and their connected relations, we take faces as node (element), their connected relations as branch to construct face-aspect-graph (called also *attribute graph* here). With the help of face-aspect-interpretation, we further construct primitive graph, which describes volume-primitives, and then construct attribute hypergraphy through describing connected relations between volume-primitives.

2.1 Face-aspect Coding Principle

We choose a set of usual primitives from CAD system[Fig. 1], its probable projection to be happened is shows in Fig. 2 when view point is located in general position. We take the face projected by 3-D surface as element, then analyze, study and deliberate these faces, finally refine out eleven visual faces, which contain some characters as follows:

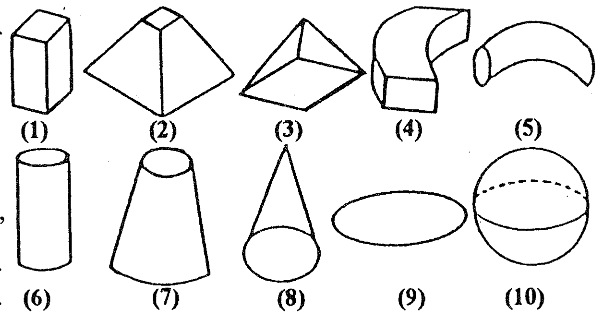


Fig. 1 Ten usual primitives

face1 It consists of four edges, all edges are straight line and each pair of opposite edge is parallel.

face2 It consists of four edges, all edges are straight line. One pair of opposite edge is parallel, another pair of opposite edge converges a same point.

face3 It consists of four edges, one pair of opposite edge is straight line, and another pair of opposite edge is curve. Both pairs of opposite edges are parallel.

face4 It consists of four edges, one pair of curve opposite edge is parallel, and another pair of straight opposite edge converge a same point.

face5 It consists of four curve edges, each pair of two curve opposite edges is parallel.

face6 It consists of three edges, the bottom is straight line, two other edges are curve.

face7 It consists of three straight edges.

face8 It consists of three edges, the bottom edge is curve, two other edges are straight line.

face9 Its shape is an ellipse.

face10 Its shape is a circle.

face11 The face is constructed by two curve edges.

The eleven visual faces are taken as the foundation of face-aspect-interpretation, we represent them with the ordinal numbers **1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11**[Fig. 3]. Each ordinal number corresponds to a fixed visual face. Each face corresponds to a fixed code. Their coding principles are suggested as follows:

1) The amount of edges in face-aspect is defined as first-figure number of coding. For example, if

a face-aspect consists of four edges, the first-figure is 4. Others are on the analogy of this.

2) If the edge in face-aspect is straight line, the code is defined as 1, or as 0.

3) If each pair of opposite edge is parallel, the code is defined as 1, or as 0.

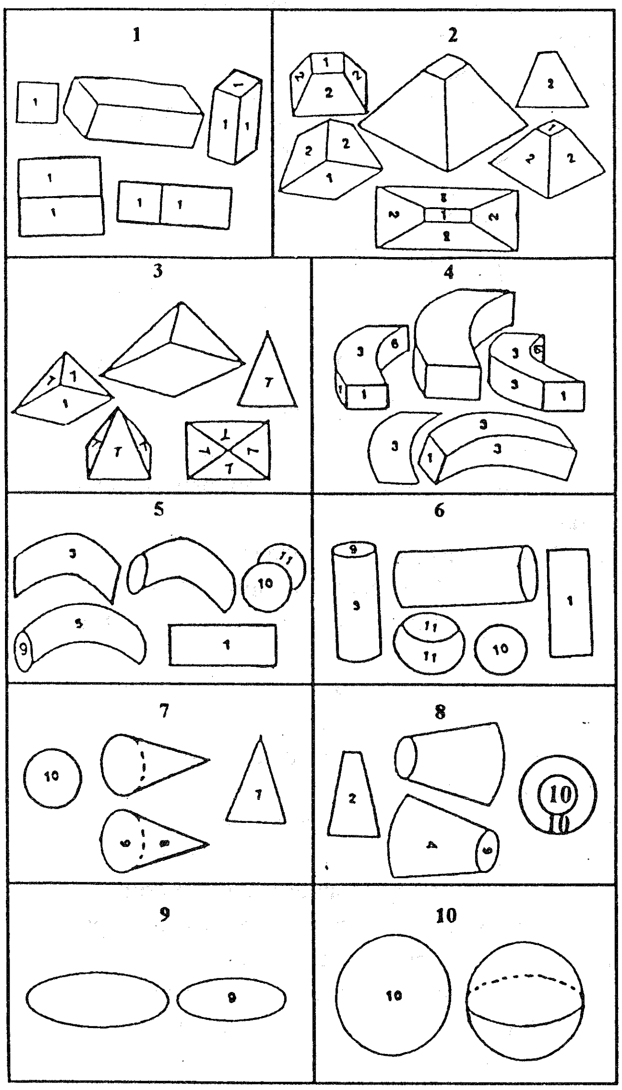


Fig. 2 Projection of primitives

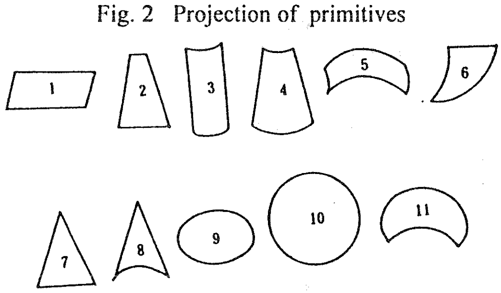


Fig. 3 Face-aspect code

4) If the code of face-aspect is 10, it shows the face possibly is a circle or an ellipse. In this case, curvature should be computed. If curvature is equal in any point, it shows the face is a circle, its code is defined as 10c, or it is an ellipse and its code is defined as 10e.

5) If a face-aspect is composed of more than four edges, it possibly is an unit or intersection of many primitives by Boolean operator. The coding principle is in the same way as above.

We build up a base of face-aspect code for eleven face-aspect, and their codes correspond to **1.** *4111111,* **2.** *4111110* or *4111101*, **3.** *4101011* or *4010111*, **4.** *4101001* or *4010110,* **5.** *4000011,* **6.** *3100* or *3001* or *3010,* **7.** *3111*, **8.** *3011*, *3110* or *3101*, **9.** *10e,* **10.** *10c,* **11.** *20.*

2.2 Coding Mergence Principle

In some case, face-aspect can correctly be interpreted into volume-primitive using face-aspect code stored in base. However, a primitive-aspects in other case maybe be two or several face-aspect mergence. So it is necessary to study the mergence principle of face-aspect. When view point is located in general position, we result in twelve face-aspect mergence relations, which describe the projection of two neighbor face-aspects, and then classify these mergence codes into a base of mergence code[Fig. 4]. In base of mergence code, the mergence code means the mergence of two ordinal numbers describing face-aspect in base of face-aspect code. For example, mergence code 11 means two face-aspect codes are 4111111 because ordinal number 1 stands for face-aspect 4111111. Others are on the analogy of this.

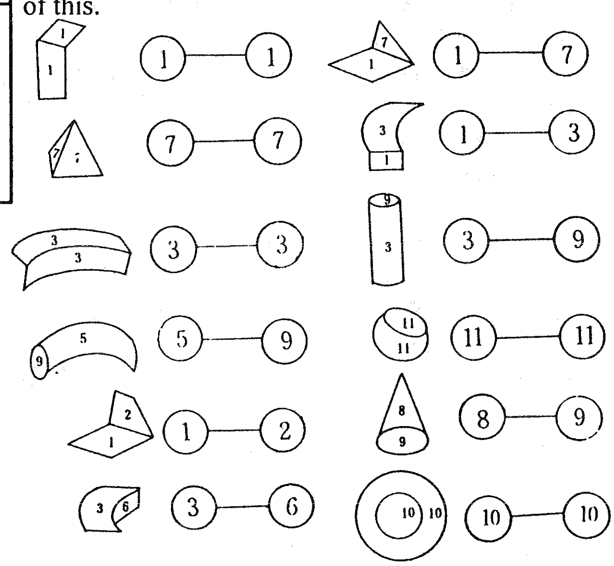


Fig. 4 Face-aspect mergence code

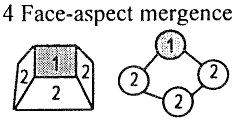


Fig. 5 Multiple faces mergence

For mergences more than two face-aspects, the principle is that if only a code differs from others in base of face-aspect, these visual faces are projection of a 3-D primitive. For example, Fig. 5a is a rhombus, its mergence code is 1222, and only a code is 1, others all are 2. Fig. 5b is its aspect-graph.

2.3 Coding Split Principle

Some primitive-aspects can correctly be interpreted by face-aspect code and face-aspect mergence code. However, projection in fact is very complex. In the process of two primitives merging by Boolean operator, the face could sometimes be smooth, and its 2-D projection seems like a face after image segment. But it is in fact mergence of several volume-primitives. In order to successfully interpret aspect like these easy confusion, the scheme of split face-aspect is proposed. We conclude some projection situations, and find that when neighbor edges within the same face form an edge-tuple, if the edge-tuple has the same code as well as differ from other neighbor edge code, the aspect is mergence projection of two 3-D primitives. In this case, we must split the original aspect into two aspects. Its split principle is to connect the head and end with neighbor edge code (1 or 0). 1 stands for straight line, and 0 stands for curve. Therefore, the original code must be rearranged into two split aspects. For example, Fig. 6a shows a projection of both cone and cylinder, if the aspect *A* be not split into two aspects, its code is 510011. In term to split principle above, we must split the aspect *A* into aspects *1,3*. The aspect 1 and aspect 3 are connected with dotted straight line[Fig. 6b] since neighbor code is 1(for straight).Their codes after split are 4111111, 3100 or 3001 or 3010 respectively.

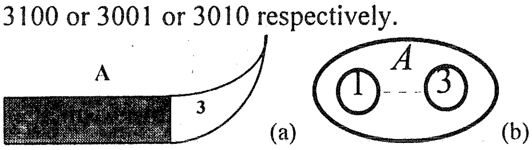


Fig. 6 Face-aspect split

So far, we can recognize some simple and regular primitives, which lost less information in projection, using aspect-interpretation including face-aspect, face-aspect mergence and face-aspect split. Nevertheless, most of primitives are still difficult to recognize correctly because original primitive-aspect is damaged by Boolean operator. So we still have to make full use the prior CAD information to achieve our aim.

3. IMAGE ATTRIBUTE RELATIONAL GRAPH-ASPECT GRAPH

There are a lot of different applications for CAD prior knowledge. We here use the CAD data structure to construct model attribute relational graph, and then implement model matching. Some definitions are in advance given as follows.

definition 1 An attributed pair is an ordinal pair(An, Vd), where An is the property name described object, and Vd is the property value. For example, describing a face-aspect is S=(area, 30), where area is property name, and 30 is property value.

definition 2 An attributed set is an m-tuple [p1, p2, ... pm]. Where each element in the tuple is an attributed pair. For example, describing a red triangle is S: [(color, red), (type, triangle), (area, 30)].

definition 3 An attributed relational graph of an object is a graph, represented G=(V, A), where V is a set of attribute node, A is a set of branch, sometimes called acre, used as describing connecting relation of two nodes.

definition 4 A primitive attribute Graph is graph represented volume-primitive projected of 3-D primitive.

definition 5 An attributed hypergraph consists of a set of hypernodes and hyperarce. Each hypernode stands for a primitive attribute graph. Each hyperarce stands for the relation between a pair of hypernode.

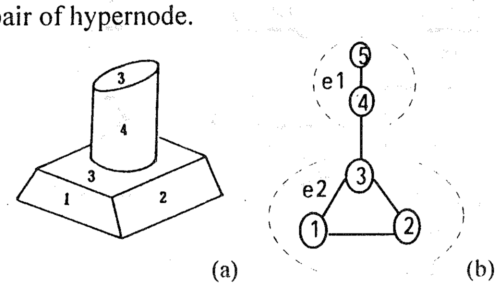


Fig. 7 Object and aspect-graph

In term to definition above, it is easy to construct the attribute graph, primitive graph and hypergraph[Fig. 7b] for some simple object such as Fig.7a. But it is sometimes relatively difficult to construct the attribute hypergraph such as Fig. 8b since we not know which face-aspect or which several face-aspects mergence can represent a whole primitive-aspect. Nevertheless, with the help of aspect-interpretation, it become very easy to construct the attribute graph, primitive graph and attribute hypergraph. So we now study how to construct the aspect-graph in term to aspect-interpretation.

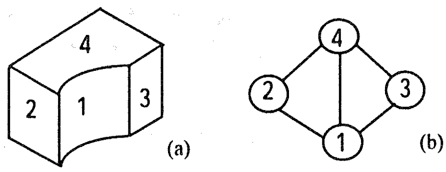


Fig. 8 Object and aspect

3.1 Image Aspect-graph Constructing Regulation

It may be completely different when constructing the aspect-graph duo to various tasks and aims. Since our aim is to recognize primitive, it is a key work to construct the primitive attribute graph and attribute hypergraph. But they are on the basis of describing face-aspect-graph, so we have to take face as an element(node). The regulation of constructing attribute graph is

1) A face-aspect is taken as a node[Fig.9a].
2) A share edge between neighbor face-aspect is taken as a branch[Fig. 9b].
3) The information in face-aspect, such as type, area, etc. is taken as subset of node [Fig.9c].
4) The information of ring, such as internal or external ring, is taken as subset of node [Fig. 9d].
5) The information of share edge is taken as subset of branch [Fig. 9e].
6) The information of two end-points in share edge is taken as subset of branch [Fig. 9f].

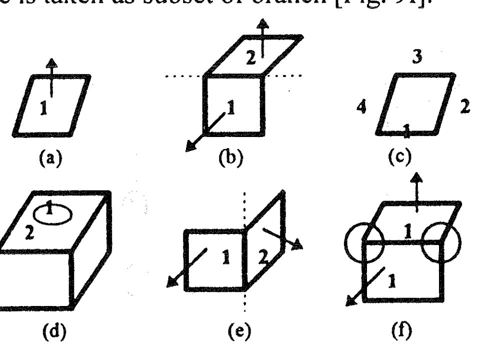


Fig. 9 The regulation of attribute graph

In addition, the attribute values are still suggested as follows:

1) The regulations to nodes are

• The area of face-aspect is taken as attribute value.

• The perimeter of face-aspect is taken as attribute value.

• The mass center coordinates(Xc, Yc) of face-aspect is taken as attribute value.

• The direction angle of face-aspectθ is taken as attribute value. where $X_c = M_{10}/M_{00}$ $Y_c = M_{10}/M_{00}$, $\theta = 1/2\, a\tan[2\, M_{11}/(M_{20} - M_{02})]$ $M_{i,j}$ is i, j-order moment, $M_{i,j} = (X_c - X)^i (Y_c - Y)^j$

2) The regulations to branch are

• The amount of share edge is taken as attribute value.

• The type of share edge is taken as attribute value. Straight line is 1, or 0.

• The slope of straight line is taken as attribute value.

• The curvature of curve is taken as attribute value.

• The information of two end-points of share edge is taken as attribute value.

It is convenient to construct 2-D attribute relation graph in term to proposal above. With the help of aspect-interpretation, we can further construct primitive attribute graph and attribute hypergraph. It is sometimes necessary to describe the connected relation of two primitive attribute graphs in attribute hypergraph, the attribute value is suggested as follows.

• The primitives recognized by aspect-interpretation is taken as attribute value.

• The code of face-aspect or face-aspect mergence is taken as attribute value.

With the regulation proposed above, we can construct an attribute relational graph, primitive attribute graph and attribute hypergraph, which correspond face-aspect, primitive-aspect and object-aspect respectively for a complex object.

The whole process is illustrated in Fig. 10.

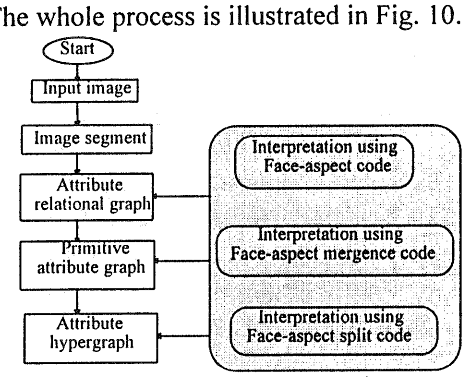


Fig. 10 The flow char of attribute graph constructing

4. FROM CAD to MODEL ASPECT-GRAPH

That we only have image aspect graph can not correctly recognize the primitive yet since a part possibly is occluded by another part or an object is occluded by another object. Therefore we must use prior CAD knowledge. Since the object is represented by data structure in CAD system, it is necessary to look for a relation between image aspect-graph and the data structure. Generally, it is very difficult to realize directly matching data structure with graph. So we have to study how to transfer data structure into attribute graph.

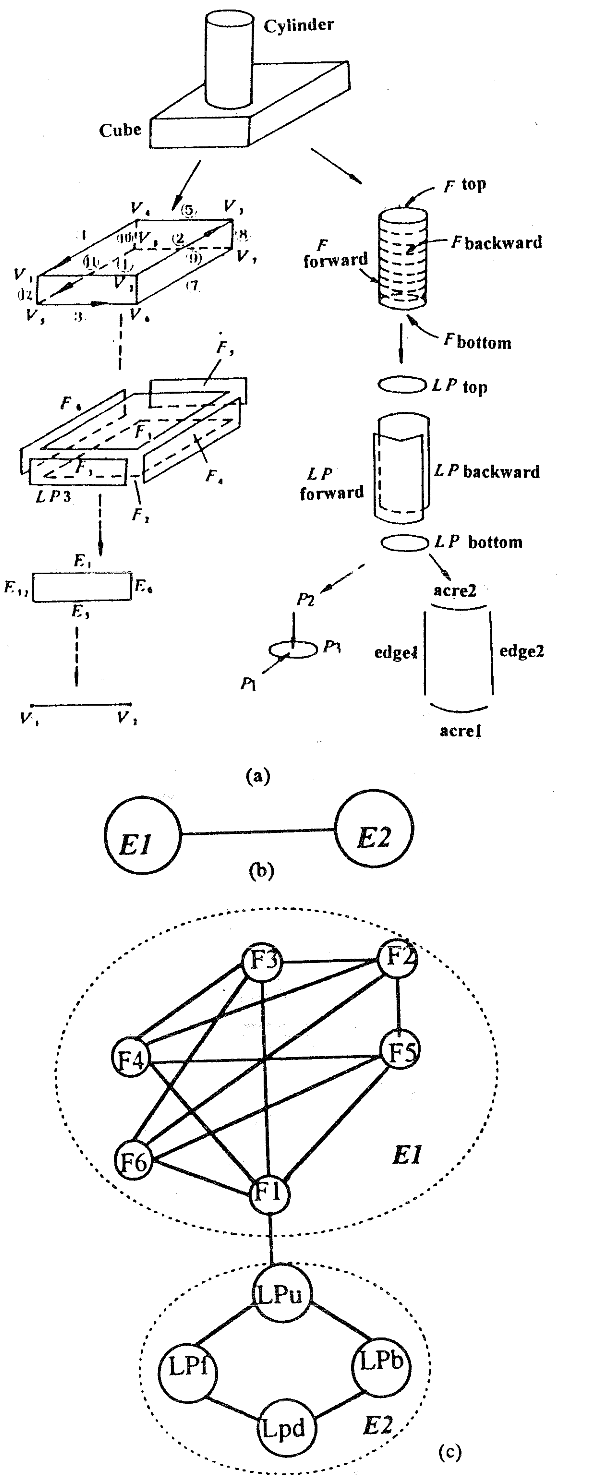


Fig. 11 CAD data construct and its aspect-graph

An object is represented by Construct Solid Geology(CSG) and Boundary represent(B-rep) in applied CAD system[10,11], i.e. an object is composed of many primitives by Boolean operator(unit, intersection and difference), and the primitive is described in B-rep. The Wing-edge structure is used as describing the topology information of boundary model[10,11]. In such an object represented with this kind of data structure, the object can be divided into volume, face, ring, edge and vextice[10]. Obviously, the mode attribute hypergraph describing this 3-D object can directly be constructed in term of its data structure. The primitive graph describing primitive can also directly be constructed according to its data structure. The attribute relational graph describing face can be constructed in the same way.

So the order of constructing mode aspect-graph is attribute hypergraph, primitive attribute graph and attribute graph, which correspond to object, volume and face respectively. The constructing regulation is the same as one of image aspect-graph above. For example, the Fig. 11a is an industrial object and its data structure represented with combining CSG and B-rep in GEMS[10,11], Fig. 11b is its attribute hypergraph, which describes connected relation between volumes. Fig. 11c is an attribute graph, which describes faces and their connected relation. The graph E1 and E2 are volume graph describing cone and cylinder respectively. How to transfer data structure into model aspect-graph was described in detail in references[6,10,11].

5. MODEL MATCHING and DATA STRUCTURE

In order to look for the relation between image graph and model graph, we quote several lemmas as follows [20].

lemma 1 The total number of nodes in a model graph is equal to the total number of faces in the 3-D object.

lemma 2 The total number of branches in a model graph is equal to the total number of edges in the 3-D object. An edge may be a straight or curved physical edge.

lemma 3 The connecting total number of every node is equal to the total number of all physical edges which the node lies in surface.

lemma 4 The projection from 3-D space does not change the topology of how the vertices are connected, The projection only hides some of the straight and junction in 2-D. These are called hidden straight and hidden surface in computer graphics literature.

lemma 5 The projection graphs constructed from the 2D projections of 3D object are subgraph isomorphism of the model graph constructed from the 3D object. Simply speaking, projection graph is subgraph isomorphism of model graph.

Obviously, since all possible projection graphs constructed from the 2D projections of a 3D object are subgraph isomorphism of the model graph. A projection graph may be match more than one model graph, and projection graph only conveys information about the topology of how vertices and edge are connected in 3D object. We therefore call subgraph matching is topology matching. i.e.

lemma 6 The subgraph isomorphism relationship between a projection graph and a model graph is a necessary but not sufficient condition for a 2D projection to match a 3D object.

As discussed above, model matching becomes two subprombles-topology matching and hypothesis verification. Topological matching become a problem of searching for subgraph isomorphism between a projection graph and a model graph in our approach.

Model matching based on aspect-interpretation in our system is a multiple levels matching. The first level is matching image hypergraph with model hypergraph. The purpose in this step is to recognize a few primitives whose information loss less in projection, and provides the start research point for next level matching. Since a part of object possibly occlude another part, as well as some faces are default due to Boolean operator(unit, intersection, difference). A number of primitives can not correctly be recognized in this step. It is necessary to implement the second level match. The second level is matching image attribute graph with model attribute graph. No matter which level match is it, the tree-research recursively is adopted[2,6].

In model matching, that there are a lot of techniques, such as selecting start point, cutting branch, selecting research approach, lessening the tree research burden, and so on were described in detail in [6]. In our scheme, an unique primitive is used as start research point, tree-research is used as mode matching.

Since subgraph isomorphism problem is NP-complete. It is therefore very unlikely that an efficient algorithm (i.e. an algorithm running in polynomial time) for solving this problem will even be found. But selecting good algorithm and data structure are beneficial to fast research. Our data structure in model matching is illustrated in Fig.12.

6. A MEASUREMENT SYSTEM BASED on CAD and LP

The developing sub-system which I suggested, is included the integration of CAD, Photogrammetry and Robot. The initial attempt developing this system is to try to build up a bridge between CAD and CV using photogrammetry approach. But now this sub-system has been installed into SIVE(Space Intelligent Vision Equipment) system[7]. I here briefly describe this sub-system.

6.1 Hardware

The system is implemented on a Sun 4/260 workstation with a Matrox MVP-ATi video processor. Two CCD cameras mounted on the ceiling above the testbed are black and while Javelin video cameras (model JE2362), a Video Junction Box (containing the camera power supply and connections enabling the two cameras to be genlocked). Two cameras can be realized self-calibrate in fixed control filed. The JE-2326 has a relatively high resolution array of 512(H) by 512(V) pixel. Digitized light intensity is 8 bit/pixel(256 gray level). The computer is current Sun 4/260 with a 10 MIPS.

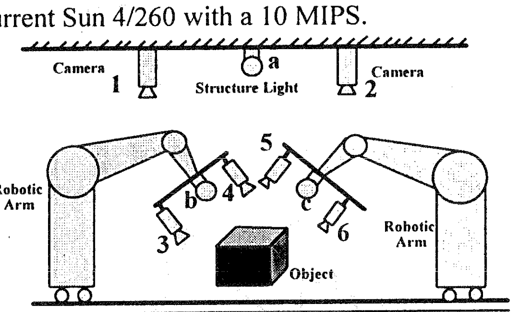


Fig. 13 SIVE System

6.2 Software Modules

Figure 14 illustrated the organization of system. All program is written in C-language under UNIX environment. I here want to introduce the CAD system.

The connected CAD system is a called GEMS(Geometric Modeling System) system, which is developed by Department of Computer Science and Technology at Tsinghua University. In this system, the object is represented with CSG and B-pre[10,11].

7. EXPERIMENTS

In order to verify the recognition capability, we experiment several industrial objects. An object is in scene in figures 15,16,17; three objects are in scene in figure 18; many an objects are in scene in figure 19. The results of recognition and measurement correspond to these figures b. Figure 17b,18b,19b are overlap of original image and contour image. Now we briefly summarize the recognition processes.

The original image is firstly put into computer after CCD grabbing, then filtering, enhancing and smoothing are implemented. Finally, the image is segmented into faces. Several main processing are dealt with as follows:

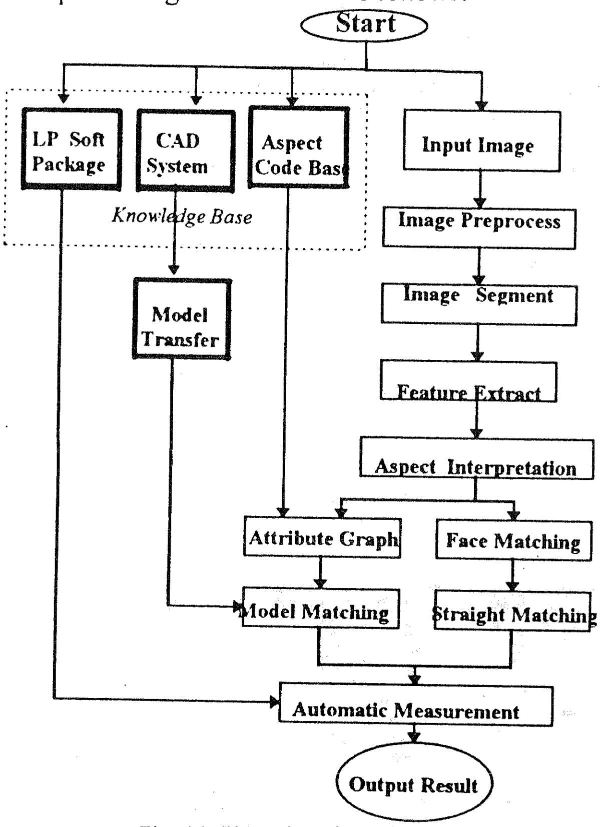


Fig. 14 Flow chart for soft component

1) Image segment: Man-machine interacting is designed for non-perfect segment since weak reflect edge, puzzle edge as well as discontinuous edge in image segment.

2) Face coding: In order to code for face, it is necessary to implement below steps:

- Face filling, which is adopted 4-neighbor seed point algorithm, it can obtain the area.
- Detecting boundary, after being filled with above, the face is binary gray image. Detecting, thinning and deleting burr are implemented for this boundary. Consequently, the boundary is determined with a pixel width.
- Boundary vector, which is adopted 8-neighbor tracking operator. It can obtain vector data of boundary and its perimeter.
- Detecting corner, which is adopted curvature difference maximum between forward K steps and backward K steps (K=3). It is usually necessary to suppress local non maximum.
- Put in order again for vector data, in order to detect whether the line is straight or curve or not between neighbor two corners, vector data have to be put in order again.
- Detecting line, which is to detect the line is straight or curve.
- Face coding, after being operated above, face coding is completed in the end.

3) Constructing attribute graph: We first put in order the faces with order number 1,2,3,···, then define the attribute name and attribute value according to regulations above. The connected relation is determined by neighbor faces which have the same coordinates.

4) Constructing volume primitive graph and hypergraph: The face interpretation by seeking for consistent face code is done, and then constructing the volume primitive graph and attribute hypergraph according to face coding labeling, face code mergence and face code splitting principle.

5) Model attribute graph: data structure from CAD system is translated into attribute relational graph including attribute hypergraph, primitive graph and attribute relational graph.

6) Model matching: an unique primitive is selected as start research point. Matching is adopted tree-back research.

7) Recognition of primitives: various primitives is recognized by operations above.

8) Reconstruction of 3D object: Recognized object is measured by LP, and then is represented with contour.

8. CONCLUSION

The presented system is an initial attempt, which integrates CAD, Photogrammetry and CV. However, it is a good start for photogrammetry in industrial application. It builds up a bridge between CAD and CV using photogrammetry. We should further perfect its functions.

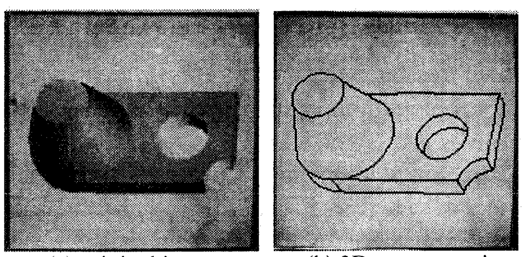
Acknowledgments

Most part of the work described in this paper was done when author is in pursuing Ph.D. in Dept. of Photogrammetry and Remote Sensing at Wuhan Technical University of Surveying and Mapping (WTUSM). So the author is certainly very greatly to my advisor, Prof. Li Deren, for the novelty idea and broad knowledge inspiring me, as well as enthusiasm and continued support. The author also wants to thank my colleagues, from particularly CAD and vision research group in Dept. of computer science and technology at Tsinghua University for many stimulating and helpful discussions in module of model transfer, which is how to transfer data structure in CAD system into vision model(attribute relational graph). They fully appreciate the work, and allow to mount our system into GEMS system and help me test whole system.

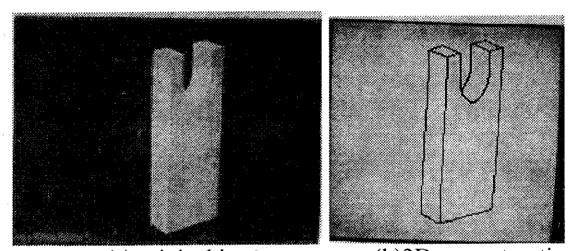
REFERENCES

1. I. Biederman, Human Image Understanding: recent research and theory, CVGIP, Vol. 37, No. 1, PP.29-73, 1985.

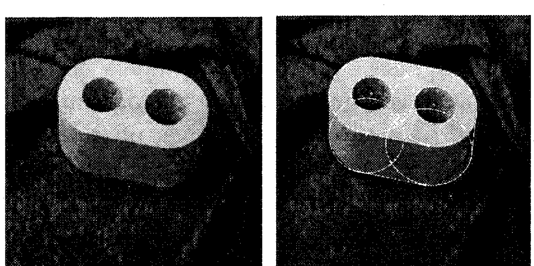
2. J.K. Cheng, T.S.Huang, A subgraph Isomorphism Algorithm Using Resolution, Pattern recognition, Vol. 13, No.5, PP.371-379, 1981.
3. Li Deren, Zhou Guoqing, CAD-based Line Photogrammetry for Automatic Measurement and Reconstruction of Industrial Object. in ISPRS Comm. V, Melbourne, March, PP. 231-240, 1994.
4. Li Deren, Zhou Guoqing, A Feasibility Study on Measurement Primitive Using Line Photogrammetry. Journal of Surveying and Mapping, Vol. No. 4, PP. 231-245, 1994.
5. S.J. Dickinson etc. 3-D Shape Recovery Using Distributed Aspect Matching, IEEE, PAMI-14, No.2, PP.174-198,1992
6. Zhou Guoqing, CAD-based Line Photogrammetry for Automatic Measurement and Reconstruction of Industrial Object. Ph.D. dissertation, Wuhan Technical Univ. of Surveying and Mapping, Oct. 1994.
7. Zhou Guoqing, A Vision System with Multiple Sensors for Spacr Robot, XVIII ISPRS Congress, Viena, Austria, 1996.
8. M.R. Henderson Extraction of feature information from three dimension CAD data, Ph.D. Dissertation, Purdue Univ. West-Lafayettee, IN 21094.
9. S.B. Joshi, CAD Interface for Automated Process Planning Ph.D. Dissertation, Purdue Univ. West-Lafayettee, IN 21094,
10. Sun, Jiaguang, etc., Three Dimension Geometric Modeling System-GEMS, Journal of Computer, vol.4, No.4, PP. 249-258, 1990
11. Sun, Jiaguang, etc., Data Structure for Geometric Modeling, Journal of Computer, No. 3, PP. 181-193, 1989
12. H. Kung, An Investigation Into Development of Process Plans from Solid Geometric Modeling Representation Ph.D. dissertation, Okla-homa State Univ., Stillwater, 1984
13. L.K.Kyprianou Shape Classification in Computer-Aided Design Ph.D. dissertation, Cambridge Univ., Cambridge, England July, 1980.
14. Y.C., Lee, and K.S. Fu, Machine Understanding of CSG: Extraction and Unification of Manufcturing Features, IEEE Computer Graphics Appli., PP. 20-32, 1987.
15. M. Marefat etc., Geometric Reasoning for Recognition of three-dimensional Object Features, IEEE PAMI-12, No. 10, PP. 949-965, 1990.
16. S.M. Staleey Using Syntactic Pattern Recognition to Extract Feature Information from A Solid Geometric Data Base, Compt. Mech. Eng. Vol.2, PP. 61-66, 1983.
17. W.H. Tsai etc. Subgraph Error-Correcting Isomorphism for Syntactic Pattern Recognition, IEEE, SMC-13, No.1, PP. 48-62, 1983.
18. T.C. Woo, Feature Extraction by Volume Decomposition, in proc. Conf. CAD/CAM Technol. Mech. Eng., M.I.T. pp. 76-94,1982.
19. Ren Zhonggui, CAD/CAM Principal Tsinghua University press 1991.
20. E.K.Wong Model Matching in Robot Vision By Subgraph Isomorphism, Pattern recognition, Vol.25, No. 3, PP. 287-303, 1992.



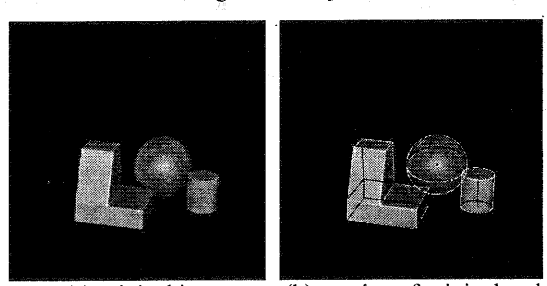
(a) original image (b) 3D reconstruction
Fig. 15 An object



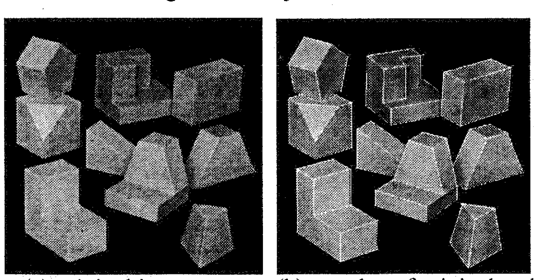
(a) original image (b)3D reconstruction
Fig. 16 An object



(a) original image (b) overlap of original and 3D
Fig. 17 An object



(a) original image (b) overlap of original and 3D
Fig. 18 Three objects



(a) original image (b) overlap of original and 3D
Fig. 19 Complex scene(many an objects)

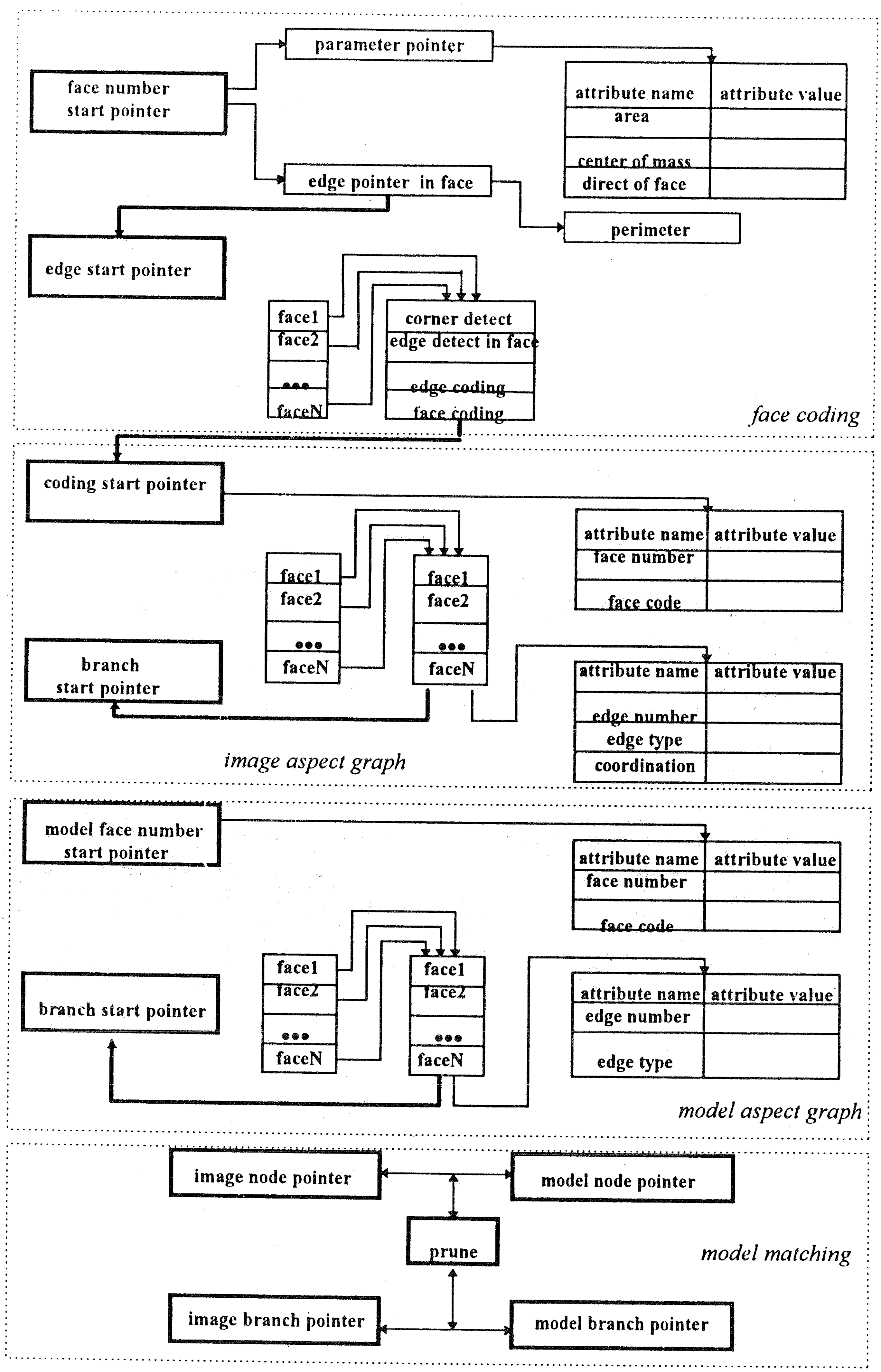


Fig. 12 data structure for model matching based on aspectt-interprettion